Radiation-Tolerant 10x SpaceWire Router
Radiation Tolerant 6x SpaceWire Router with PCI
RT-SPW-ROUTER
Data Sheet and User's Manual

Features

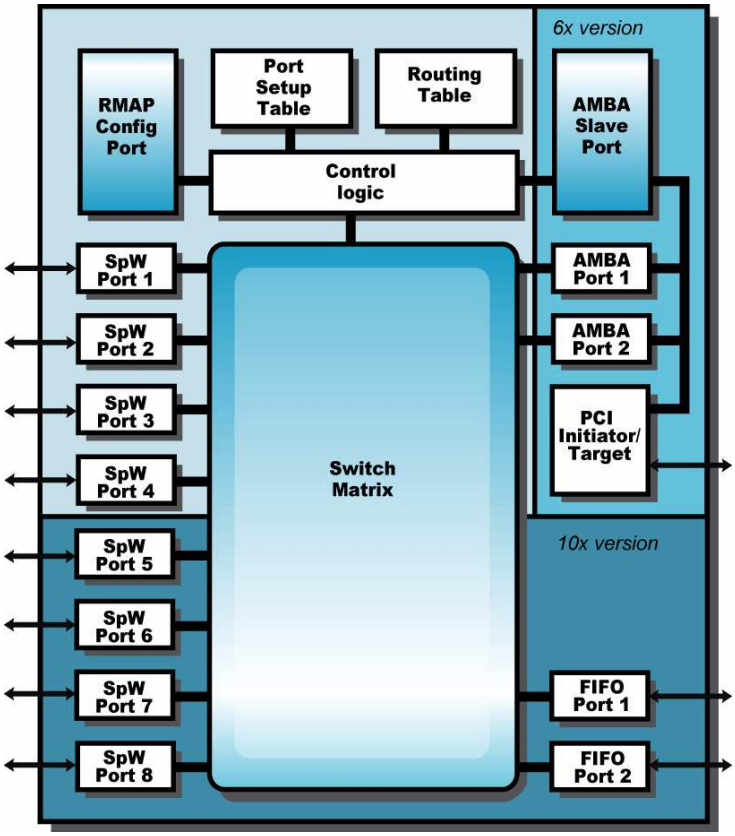
- SpaceWire Router compliant with ECSS-E-ST-50-12C
- Non-blocking switch-matrix connecting any input to any output
- Packet Distribution
- Group Adaptive Routing
- Path, Logical and Regional Logical addressing
- Two priority levels for output port arbitration
- Configuration port using RMAP, compliant with ECSS-E-ST-50-52C
- 8x SpaceWire ports, full duplex, on-chip or off-chip LVDS transceivers
- 2x external FIFO ports, 9-bit wide data paths
- Possible to cascade two routers without glue logic via the FIFO ports
- 2x internal AMBA ports with DMA and RMAP
- PCI Initiator/Target interface, 32-bit, 33 MHz
- System-time distribution via all ports
- Timers on all ports to prevent deadlock
- Fault-tolerant design

Description

The Radiation-Tolerant SpaceWire Router family is available as standard components using the Actel RTAX and RT ProASIC3 FPGAs. The fault tolerant design of the router in combination with the radiation tolerant FPGA makes it ideally suited for space and other high-rel applications.

Specification

- CCGA484, CQFP352, CCGA624
- Total Ionizing Dose (TID) to 300 krad (Si, functional)
- Single-Event Latch-Up Immunity (SEL) to LET_{TH} > 117 MeV-cm^2/mg
- Immune to Single-Event Upsets (SEU) to LET_{TH} > 37 MeV-cm^2/mg
- 1.2-1.5V, 2.5V & 3.3V supply, 500 mW consumption (TBD)
- Up to 200 MBPS on SpaceWire links



Applications

The router implements a routing switch as defined in the ECSS-E-ST-50-12C SpaceWire links, nodes, routers and networks standard, supporting all mandatory and optional features.

The router implements up to ten external routing ports and the mandatory configuration port.

The configuration port provides access to configuration and status registers, and the routing table using the Remote Memory Access Protocol (RMAP) as defined in the ECSS-E-ST-50-52C protocol standard.

The router also fully supports the ECSS-E-ST-50-51C SpaceWire protocol identification standard.

An optional PCI Initiator/Target interface is available.

Table of contents

1 Introduction

1.1 Overview

The SpaceWire Router family is based on a common architecture from which standard configurations are derived. The architecture is centered around the a non-blocking switch matrix which can connect any input port to any output port.

All the addressing modes such as path, logical and regional logical are supported. Group adaptive routing is fully supported, meaning that both path and logical addresses can be individually configured to use from one up to all ports. A unique feature is the support for packet distribution, which can be used to implement multicast and broadcast addresses. Output ports are arbitrated using two priority levels with a round-robin scheme within each level.

The 10x SpaceWire router implements 8 external SpaceWire ports and 2 external FIFO ports. The SpaceWire and FIFO interfaces are directly connected to the switch matrix.

The 6x SpaceWire router implements 4 external SpaceWire ports and a PCI initiator and target interface (32-bit, 33 MHz) accessed via two internal AMBA ports with DMA and RMAP. In this configuration, the AMBA Advanced High-speed Bus (AHB), to which the AMBA ports and the PCI interface are connected, is used for high-speed communication between the switch matrix and the external PCI interface. Supporting low-bandwidth items, such as configuration registers, are connected to the AMBA Advanced Peripheral Bus (APB) which is accessed through an AHB to APB bridge.

The full SpaceWire router architecture includes the following modules: SpaceWire Router, PCI Initiator/Target Interface, AMBA AHB Debug UART and AMBA AHB Status Register.

1.2 Standard configurations

Due to resource limitations, it is not possible to fit the full SpaceWire router architecture on a single device. Therefore, sub-set configurations have been defined to suit applications with different interfacing needs. Table 1 below shows which functions are available in each of the configurations.

Table 1. Standard configurations

Configuration name	10x SpaceWire Router	6x SpaceWire Router with PCI
Configuration ID (CID)	1	2
SpaceWire Ports	8	4
FIFO Ports	2	
AMBA Ports with RMAP		2
Configuration port with RMAP	Yes	Yes
Configuration port with AMBA		Yes
PCI Initiator/Target and Interrupt Controller		Yes
AMBA Status Register		Yes
UART Debug Link		Yes
JTAG Debug Link		
FPGA and Package	Actel RTAX2000S CCGA624	Actel RTAX2000S CCGA624
	Actel RTAX2000S CQFP352	Actel RTAX2000S CQFP352
	Actel RT3PE3000L CCGA484	
SpaceWire Physical Layer	LVTTL / LVDS (on-chip)	LVTTL / LVDS (on-chip)

1.3 Signal overview

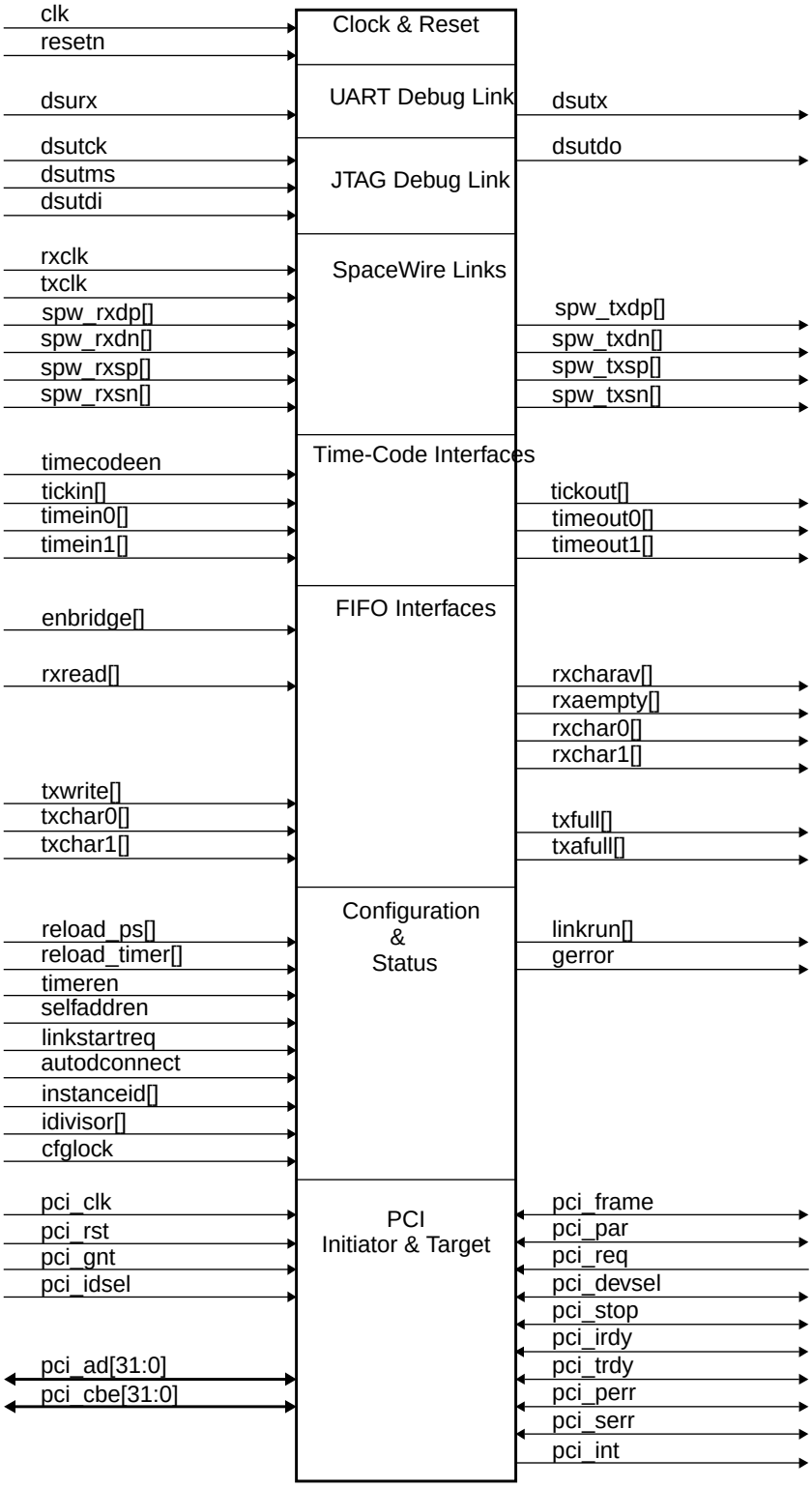


Figure 1. Signal overview

2 Architecture

2.1 Cores

The common architecture of the SpaceWire router family is based on cores from the GRLIB IP library. The vendor and device identifiers for each core can be extracted from the plug & play information. The used IP cores are listed in table 2.

Table 2. Used IP cores

Core	Function	Vendor	Device	CID
AHBCTRL	AHB Arbiter & Decoder	0x01	-	2
APBCTRL	AHB/APB Bridge	0x01	0x006	2
GRSPWROUTER	SpaceWire Router	0x01	-	All
	AMBA port	0x01	0x8A	2
	AMBA configuration port	0x01	0x8B	2
PCIF	PCI Initiator/Target	0x01	0x075	2
AHBUART	Serial/AHB debug interface	0x01	0x007	2
AHBJTAG	JTAG/AHB debug interface	0x01	0x01C	-
AHBSTAT	AHB failing address register	0x01	0x052	2

2.2 Interrupts

The SpaceWire router family uses the same interrupt assignment for all configurations. See the description of the individual cores for how and when the interrupts are raised. All interrupts are handled by the interrupt controller and forwarded to the PCI bus. Note that this only applies to CID 2.

Table 3. Interrupt assignment

Core	Interrupt	Comment	CID
GRSPWROUTER	1, 2	AMBA ports 0 and 1, respectively	2
PCIF	3	(spare for future use)	2
AHBSTAT	4	(spare for future use)	2

2.3 Memory map

The SpaceWire router family uses the same memory map for all standard configurations. The memory map shown in table 4 is based on the AMBA AHB address space. Access to addresses outside the ranges will return an AHB error response. The detailed register layout is defined in the description of each individual core. Note that this only applies to CID 2.

Table 4. AMBA AHB address range

Core	Address range	Area	CID
PCIF	0x00000000 - 0x3FFFFFFF	PCI memory area, 1 GByte	2
APBCTRL	0xFFE00000 - 0xFFEFFFFF	APB bridge	2
PCIF	0xFFF00000 - 0xFFF1FFFF	PCI I/O area	2
GRSPWROUTER	0xFFFC0000 - 0xFFFC0FFF	Configuration area	2
AHB plug&play	0xFFFFF000 - 0xFFFFFFFF		2

The control registers of most on-chip peripherals are accessible via the AHB/APB bridge, which is mapped at address 0x80000000. The memory map shown in table 5 is based on the AMBA AHB address space. Note that this only applies to CID 2.

Table 5. APB address range

Core	Address range	Comment	CID
GRSPWROUTER	0xFFE00100 - 0xFFE001FF	AMBA port 0	2
GRSPWROUTER	0xFFE00200 - 0xFFE002FF	AMBA port 1	2
PCIF	0xFFE00300 - 0xFFE003FF		2
AHBUART	0xFFE00400 - 0xFFE004FF		2
AHBJTAG	0xFFE00500 - 0xFFE005FF		-
AHBSTAT	0xFFE00600 - 0xFFE006FF		2
APB plug&play	0xFFEFF000 - 0xFFEFFFFF		2

2.4 Plug & play information

The LEON3FT RTAX family uses the same plug & play map for all standard configurations. The plug & play memory map and bus indexes for AMBA AHB masters are shown in table 6 and is based on the AMBA AHB address space. Note that this only applies to CID 2.

Table 6. Plug & play information for AHB masters

Core	Index	Function	Address range	CID
GRSPWROUTER	0	AMBA port 0	0xFFFFF000 - 0xFFFFF01F	2
GRSPWROUTER	1	AMBA port 1	0xFFFFF020 - 0xFFFFF03F	2
PCIF	2	PCI Initiator/Target	0xFFFFF040 - 0xFFFFF05F	2
AHBUART	3	Serial/AHB debug interface	0xFFFFF060 - 0xFFFFF07F	2
AHBJTAG	4	JTAG/AHB debug interface	0xFFFFF080 - 0xFFFFF09F	-

The plug & play memory map and bus indexes for AMBA AHB slaves are shown in table 7 and is based on the AMBA AHB address space.

Table 7. Plug & play information for AHB slaves

Core	Index	Function	Address range	CID
APBCTRL	0	AHB/APB Bridge	0xFFFFF800 - 0xFFFFF81F	2
GRSPWROUTER	1	Configuration area	0xFFFFF820 - 0xFFFFF83F	2
PCIF	2	PCI Initiator/Target	0xFFFFF840 - 0xFFFFF85F	2

The plug & play memory map and bus indexes for AMBA APB slaves are shown in table 8 and is based on the AMBA AHB address space.

Table 8. Plug & play information for APB slaves

Core	Index	Function	Address range	CID
GRSPWROUTER	0	AMBA port 0	0xFFEFF000 - 0xFFEFF017	2
GRSPWROUTER	1	AMBA port 1	0xFFEFF018 - 0xFFEFF01F	2
PCIF	2	PCI Initiator/Target	0xFFEFF020 - 0xFFEFF027	2
AHBUART	3	Serial/AHB debug interface	0xFFEFF028 - 0xFFEFF02F	2
AHBJTAG	4	JTAG/AHB debug interface	0xFFEFF030 - 0xFFEFF037	-
AHBSTAT	5	AHB failing address register	0xFFEFF038 - 0xFFEFF03F	2

2.5 Specifications

Table 9. Specifications

Technology	Actel RT Axcelerator	Actel RT ProASIC3
Device	RTAX2000S/SL	RT3PE3000L
Link speed	200 MBPS	100 MBPS
Receiver clock	200 MHz	100 MHz
Transmitter clock	100 MHz (DDR)	50 MHz (DDR)
System clock	25 MHz (CID 1) / 33 MHz (CID 2)	25 MHz (CID 1)
PCI clock	33 MHz	N/A
Power consumption	500 mW (TBD)	500 mW (TBD)
Package	CQFP352, CCGA624	CCGA484
Total Ionizing Dose [1)]	300 krad (Si)	15 krad (Si)
Single-Event Latch-Up Immunity (SEL)[1)]	LET_{TH} > 117 MeV-cm^2/mg	LET_{TH} > 68 MeV-cm^2/mg
Single-Event Upsets (SEU) [1)]	LET_{TH} > 37 MeV-cm^2/mg	LET_{TH} > 6 MeV-cm^2/mg (before TMR)
Supply voltage	1.5 V, 2.5 V & 3.3 V	1.2 V -1.5 V, 2.5 V & 3.3 V
Screening level [2)]	B, E, EV and PROTO	B

Note 1: Refer to Actel RTAX and RT ProASIC3 data sheets [RTAX] and [RT3PE] for details.

Note 2: The pre-programmed Actel RTAX and RT ProASIC3 devices are shipped in all quality levels supported by Actel, as defined in [RTAX] and [RT3PE], respectively.

2.6 Signals

The common architecture has the external signals shown in table 10.

Table 10. External signals

Name	Usage	Direction	Polarity	CID
clk	Main system clock	In	-	All
resetn	System reset	In	Low	All
dsutx	Debug UART transmit data	Out	Low	2
dsurx	Debug UART receive data	In	Low	2
dsutck	Debug JTAG Clock	In	-	-
dsutms	Debug JTAG Mode	In	High	-
dsutdi	Debug JTAG Input	In	High	-
dsutdo	Debug JTAG Output	Out	High	-
rxclk	SpaceWire link receive clock	In	-	All
txclk	SpaceWire link transmit clock	In	-	All
spw_rxdp[7:0]	Data input, positive	In, LVDS	High	All 1) 3)
spw_rxdn[7:0]	Data input, negative	In, LVDS	Low	All 1) 3)
spw_rxsp[7:0]	Strobe input, positive	In, LVDS	High	All 1) 3)
spw_rxsn[7:0]	Strobe input, negative	In, LVDS	Low	All 1) 3)
spw_txdp[7:0]	Data output, positive	Out, LVDS	High	All 1) 3)
spw_txdn[7:0]	Data output, negative	Out, LVDS	Low	All 1) 3)
spw_txsp[7:0]	Strobe output, positive	Out, LVDS	High	All 1) 3)
spw_txsn[7:0]	Strobe output, negative	Out, LVDS	Low	All 1) 3)
spw_rxd[7:0]	Data input	In,	High	All 2) 3)
spw_rxs[7:0]	Strobe input	In	High	All 2) 3)
spw_txd[7:0]	Data output	Out	High	All 2) 3)
spw_txs[7:0]	Strobe output	Out	High	All 2) 3)
timecodeen	Enable time-code functionality	In	High	1
tickin[1:0]	Tick input signals for FIFO interfaces	In	High	1
timein0[7:0]	Time input signals for FIFO 0 interface	In	-	1
timein1[7:0]	Time input signals for FIFO 1 interface	In	-	1
tickout[1:0]	Tick output signals for FIFO interfaces	Out	High	1
timeout0[7:0]	Time output signals for FIFO 0 interface	Out	-	1
timeout1[7:0]	Time output signals for FIFO 1 interface	Out	-	1
rxread[1:0]	Receiver FIFO read signals for FIFO interfaces	In	High	1
rxchar0[8:0]	Receiver character signals for FIFO 0 interface	Out	-	1
rxchar1[8:0]	Receiver character signals for FIFO 1 interface	Out	-	1
txwrite[1:0]	Transmitter FIFO write signals for FIFO interfaces	In	High	1
txchar0[8:0]	Transmitter character signals for FIFO 0 interface	In	-	1
txchar1[8:0]	Transmitter character signals for FIFO 1 interface	In	-	1
txfull[1:0]	Transmitter full signal for FIFO interfaces	Out	High	1

Table 10. External signals

Name	Usage	Direction	Polarity	CID
txafull[1:0]	Transmitter almost full signal for FIFO interfaces	Out	High	1
rxcharav[1:0]	Receiver data available signal for FIFO interfaces	Out	High	1
rxaempty[1:0]	Receiver empty signal for FIFO interfaces	Out	High	1
enbridge[1:0]	Enables bridge mode for the FIFO interfaces	In	High	1
linkrun[7:0]	SpaceWire link in run state when asserted	Out	High	All [3)]
gerror	Global error	Out	High	All
reload_ps[5:0]	Reset value for the timer prescaler. Mapped to bits 13, 11, 9, 7, 5 and 3 of the prescaler register, other bits to 0 at reset.	In	-	All
reload_timer[3:0]	Reset value for the port timer registers. Mapped to bits 8, 6, 4 and 2 of the timer registers, other bits cleared to 0 at reset.	In	-	All
timeren	Reset value for the timer enable bit in the port control registers	In	High	All
selfaddren	Reset value for *selfaddren* register bit	In	High	All
linkstartreq	Reset value for the *linkstartreq* register bit	In	High	All
autodconnect	Reset value for the *autodconnect* register bit	In	High	All
instanceid[3:0]	Reset value for the instance *id* field of the version/instance id register. Sets bits 3:0, other bits cleared to 0 at reset.	In	-	All
idivisor[3:0]	Reset value for the initialization divisor register and the SpaceWire port run-state divisor registers. Sets bits 3:0, other bits cleared to 0 at reset.	In	-	All
cfglock	Lock configuration port accesses from all ports except port 1 when asserted	In	High	All
pci_clk	PCI clock	In	-	2
pci_rst	PCI reset	In	Low	2
pci_frame	Cycle frame	BiDir	Low	2
pci_irdy	Initiator ready	BiDir	Low	2
pci_trdy	Target ready	BiDir	Low	2
pci_stop	Stop	BiDir	Low	2
pci_idsel	Device select during configuration	In	High	2
pci_devsel	Device select	BiDir	Low	2
pci_par	Parity signal	BiDir	High	2
pci_perr	Parity error	BiDir	Low	2
pci_serr	System error	BiDir	Low	2
pci_req	Request signal	BiDir	Low	2
pci_gnt	Grant signal	In	Low	2
pci_cbe[3:0]	Bus command and byte enable	BiDir	Low	2
pci_ad[31:0]	Address and Data bus	BiDir	High	2
pci_int	Interrupt signal	Out	Low	2

Note 1: These signals are only used in configuration with on-chip LVDS drivers.

Note 2: These signals are only used in configuration with off-chip LVDS drivers.

Note 3: All bus signal elements are not used in all listed configurations. See pin assignment at end of document for details.

3 SpaceWire router

3.1 Overview

The SpaceWire router core implements a SpaceWire routing switch as defined in the ECSS-E-ST-50-12C standard. It provides an RMAP target for configuration at port 0 used for accessing internal configuration and status registers. In addition to this there are three different external port types: SpaceWire links, FIFO interfaces and AMBA interfaces. An AHB slave interface is also provided for accessing the port 0 registers from the AHB bus. Group adaptive routing and packet distribution are fully supported (two ports up to all ports can be assigned to an address). System time-distribution is also supported. Timers are available for each port to prevent deadlock situations.

3.2 Operation

The router ports are interconnected using a non-blocking switch matrix which can connect any input port to any output port. Access to each output port is arbitrated using a round-robin arbitration scheme. A single routing-table is used for the whole router. Access to the table is also arbitrated using a round-robin scheme.

The ports consist of configuration port 0 and three different types of external ports: SpaceWire links, FIFO interfaces and AMBA interfaces.

All the ports regardless of their type have the same interface to the switch matrix and behave in the same manner. The difference in behavior is on the external side of the port. The SpaceWire ports provide standard SpaceWire link interfaces using either on- or off-chip LVDS. The FIFO interfaces store characters in two FIFOs which are accessed using 9-bit wide data paths with read/write signals. Lastly the AMBA ports transfer characters from and to an AHB bus using DMA. The four different port types are described in further detail in sections 3.3, 3.4, 3.5 and 3.6.

3.2.1 Port numbering

The ports are numbered in the following order: configuration port, SpW ports, AMBA ports, FIFO ports. The configuration port is always present and has number 0. If SpW ports are present in the router they are numbered starting from number 1. If AMBA ports are present they are numbered starting from the last SpW ports. If no SpW ports are present the AMBA ports start at number 1. Lastly, the FIFO ports are numbered starting after the last AMBA port, SpW port or at number 1 depending on if AMBA ports and SpW ports are present respectively.

For example if 7 SpW ports and 4 FIFO ports are included in the router they will have port numbers 1-7 and 8-11. If 16 SpW ports, 2 AMBA ports and 7 FIFO ports are included they have port numbers 1-16, 17-18 and 19-25 respectively.

3.2.2 Routing table

A single routing table is provided. The access to this routing table is arbitrated using a round-robin arbiter with each port being of equal priority. The operation is pipelined and one lookup can be done each cycle. This way the maximum latency is equal to the number of ports in the router minus one. The impact on throughput should be negligible provided that packets are not incoming at the same time. The probability for this is higher when the traffic only consist of very small packets sent continuously (the average size being about the same as the number of ports). This should be a very uncommon case. Latency is still bounded and probably negligible in comparison to other latencies in most systems. The benefit is a reduced area enabling the router to be implemented with a higher number of ports on many FPGA technologies.

Since the latency for the lookup is very small and deterministic there is not much to gain by having configurable priorities for this. Priorities are instead used for arbitrating packets contending for an output port as described in the next section.

The routing table and all the configuration registers are configured through an RMAP target or an AHB slave interface which use the same routing table as the logic handling packet traffic. They do not introduce any extra latency because they have lower priority than the packet traffic and thus are only allowed access on cycles when no lookup is needed for packets. This can slow down configuration accesses but they are probably mostly done before packet traffic starts and very seldom afterwards.

Logical addresses have a routing table entry containing a priority bit, header deletion enable bit and a entry enable bit. The routing table entry is enabled by writing a 1 to the enable bit. It can be disabled again by writing a 0. The contents of the routing table is undetermined after reset and should not be read. When a routing table entry is disabled, packets with a destination address corresponding to that entry will be discarded and the invalid address error bit asserted.

Before the routing table entry is enabled the corresponding port setup register must be initialized. The port setup register should be written with ones to one or more bits to enable packets to be transmitted on the ports corresponding to the bit numbers. See sections 3.2.4 and 3.2.5 for more details on how to use the port setup register. If the port setup register is not initialized but the routing table entry is enabled packets with that logical address will be discarded and the invalid address error bit asserted.

The mechanism is the same for path addresses except that they do not have a routing table entry and header deletion is always enabled. Packets will be routed to the output corresponding to the path address in the packet even if the port setup register has not been initialized. For group adaptive routing and packet distribution to be used the port setup register must be initialized also for path addresses.

The routing table entries are also marked as invalid before they have been written the first time. When the entries are invalid, packets with the corresponding logical address will be discarded and an invalid address error bit asserted.

3.2.3 Output port arbitration

Each output port is arbitrated individually using two priority levels with round-robin at each level. Each path or logical address can be configured to be high or low priority. In this case the delays can be very long (compared to when arbiting for access to the routing table) before the next arbitration because packets can be very large and the speed of the data consumer and the link itself cannot be known. In this case priority assignments can have a large impact on the amount of bandwidth a source port can use on a destination port.

The priority for path addresses is set in the port's control register with the port number corresponding to the path address. For logical addresses the priority is set in the routing table entry.

3.2.4 Group adaptive routing

Group adaptive routing is used to enable a packet to be transmitted on several different paths. For example a packet with address 45 can be enabled to be transmitted on port 1 and 2. If port 1 is busy when a packet with address 45 arrives it is transmitted on port 2 instead if not busy.

Group adaptive routing is used if bit 0 in the port setup register for the corresponding path or logical address is 0. Each bit in the register corresponds to the port with the same number as the bit index. So if bit 5 is set to 1 at address offset 0x80 it means that incoming packets with logical address 32 can be transmitted on port 5. If only one bit is set for an address all packets with that address will be transmitted on that port. If one or more bits are set the group adaptive function is used and the packet is transmitted on the first available port with a bit set to 1 starting from the lowest number. A port being

available means that no other packet transmission is active at the moment and also for SpaceWire links that the link is in run-state. For path addresses the bit corresponding to the path address will always be set. This is done as specified in the standard which requires a packet with a path address to be transmitted on the port with the same number as the address. The standard does not mention what should happen when group adaptive routing is used for path addresses but in this router the bit corresponding to the port number of the path address is always set so that the packet *can* be transmitted on that port also when group adaptive routing is used.

For logical addresses the corresponding routing table entry and port setup register must be valid for the packet to be routed (otherwise it is discarded). There is no default port as with path addresses so at least one bit in the port setup register must be 1 for the packet to be routed otherwise it is discarded.

3.2.5 Packet distribution

Packet distribution can be used to implement multicast and broadcast addresses. Packets with logical address 50 can for example be configured to be transmitted on ports 1, 2 and 3 while address 51 can be configured to be transmitted on all ports (broadcast).

When packet distribution is enabled the group adaptive routing register is used to determine the ports that a packet should be transmitted on. Packet distribution is enabled for a path or logical address by setting bit 0 in the corresponding port setup register to 1. The packet will be transmitted on all the ports with a bit set to 1 in the register. This means that if one of the ports enabled for packet distribution is busy the router will wait for it to become free before transmitting on any of the ports. Due to the wormhole routing implementation the slowest link will determine the speed at which a packet is transmitted on all the ports.

When packet distribution is used with path addresses the port with the same number as the address will always be enabled (as for group adaptive routing).

3.2.6 Port disable

The disable port bit in the port's control register can be used to disable a port for data traffic. It will behave just like if the physical port was not existing. Packets transmitted to it will be spilt and the invalid address bit on the source port is set. Packets received to the port will be silently discarded, no status bits are set.

All ports except the configuration port (0) and port 1 can be disabled to prevent the situation of all ports being locked out from happening.

3.2.7 Timers

Timers are individually enabled for each port by writing the timer enable bit in the port control register. When timers are enabled during packet transmission on a port the timer is reset each time a character is transmitted. If the timer expires the packet will be discarded and an EEP is inserted on all the ports to which the packet was transmitted (can be more than one if packet distribution was used). It does not matter if it is the output port or source port which is stalling. The blocking situation is always detected at the source port which handles the spilling. It also does not matter if the stall is caused by the link being stopped or lack of credits, the discard mechanism is always the same. When the timers are not implemented or disabled the source and destination ports will always block until the blocking situation is resolved.

The timers use a global prescaler and an individual timer per port. Both the prescaler and the individual timer tick rate can be configured through the configuration port.

In group adaptive routing mode the packet will be spilt if no characters have been transmitted for the timeout period after being assigned to a port. For packet distribution a packet will be spilt if no character has been transmitted for the timeout period after being assigned to all the ports. This means that it is enough for one port to stall for the packet to timeout and be spilt.

The behavior described above also means that the timeout is handled in the same way regardless of the port type (SpW, FIFO or AMBA).

If a destination port is disabled it behaves as if it is not existing and will not be used thus being spilt immediately. Timers are not applicable in that case. For group adaptive routing and packet distribution disabled ports are also masked before transmission starts and will not affect timers.

Details for the different scenarios will be listed in the remaining sub-sections.

3.2.7.1 Timers disabled

If timers are disabled packets will always wait indefinitely regardless of stall reason. In the case that timers are present but are enabled on some ports and disabled on others it is always the source port that determines whether the timer will be active or not. This means that if a packet arrives at port 2 which has its timer enabled and it is routed to port 4 which has timers disabled a timer will be active for that packet routing and transmission. The same applies for group adaptive routing and packet distribution.

3.2.7.2 Timer enabled and output port not in run state

The timer is started when the packet arrives and if the link has not entered run-state until the timer expires the packet will be spilt. No EEP will be written to the destination port in this case. If the link start on request feature has been enabled the router will try to start the link but still only waits for the timeout period for the link to start.

3.2.7.3 Timer enabled and output port in run state but busy with other transmission

The packet will wait indefinitely until the destination port becomes free. In the case that the destination port is stalled the port currently sourcing the packet for it has to have its timer enabled and spill the packet before the new port can be allocated for it. If the port stalls again the new port will also spill its packet after the timeout period. In this case and EEP will be written to the destination port since the transmission of the packet had started.

3.2.7.4 Timer enabled and group adaptive routing is enabled, ports not running

The timer is started when the packet arrives and if no port has been allocated until the timer expires the packet will be spilt. If link start on request is enabled the router will try to start all the links.

3.2.7.5 Timer enabled and group adaptive routing enabled, ports running but busy

The packet will wait until one port becomes free and then start transmitting. The timer is not started while waiting for busy ports.

3.2.7.6 Timer enabled and packet distribution enabled, ports not running

If at least one of the destination ports is not running the timer is started and the packet will be discarded if all the ports are not running when the timer expires.

3.2.7.7 Timer enabled and packet distribution enabled, ports running but busy

If at least one port is busy but all are running when packet distribution is enabled the packet will wait indefinitely. When the transmission has started the timer is restarted each time a character is transmitted and if the timer expires the remaining part of the packet is spilt and an EEP written to all the destination ports.

3.2.7.8 Timer functionality when accessing the configuration port

Timers work in the same way when accessing the configuration port as for the other ports. When the command is being received by the RMAP target the timer on the source port will trigger if the source of the command is too slow, spill the remaining part of the packet and insert an EEP to the configuration port. The RMAP target will always be able to receive the characters quick enough. If the source is too slow when the reply is sent the configuration port's timer will trigger and the remaining part of the packet is spilled and an EEP is inserted. This is to prevent the configuration port from being locked up by a malfunctioning source port.

3.2.8 On-chip memories

There are two memory blocks in the routing table, one for the port setup registers and one for the routing table. The port setup memory bit width is equal to the number of ports including the configuration port with depth 256. The routing table is 256 locations deep and 2 bits wide.

Each port excluding the configuration port also have FIFO memories. The SpaceWire ports have one FIFO per direction (rx, tx) which are 9-wide. The FIFO ports have the exact same FIFO configuration as the SpaceWire ports.

The AMBA ports have one 9-bit wide receiver FIFO and two 32-bit wide AHB FIFOs.

Parity is used to protect the memories and up to four bits per word can be corrected and there is a signal indicating an uncorrectable error.

If a memory error occurs in the port setup table or the routing table the memory error (ME) bit in the router configuration/status register is set and remains set until cleared by the user. If a memory error is detected in any of the ports FIFO memories the memory error (ME) bit in the respective port status register is set and remains set until cleared by the user. The ME bits are only set for uncorrectable errors.

When an uncorrectable error is detected in the port setup or routing table when a packet is being routed it will be discarded. Unocrrectable errors in the FIFO memories are not handled since they only affect the contents of the routed packet not the operation of the router itself. These type of errors should be caught by CRC checks if used in the packet.

The ME bit for the ports is only usable for detecting errors and statistics since there is no need to correct the error manually since the packet has already been routed when it is detected. The ME indication for the routing table and port setup registers can be used for starting a scrubbing operation if detected. There is also an option of having automatic scrubbing (see section 3.2.8.1)

3.2.8.1 Autoscrub

With autoscrubbing the routing table and port setup registers will be periodically read and rewritten. This is done to prevent buildup of SEUs to cause an uncorrectable error in the memories. It will run in the background and has no impact on routing table lookup for traffic but can delay configuration accesses with two cycles.

The scrubber starts at address 0 and simultaneously writes one location in the port setup memory and the routing table memory. It then waits for a timeout period until it writes the next word. Eventually the last location is reached and the process starts over from address 0.

The period between each word refresh is approximately 2^{26} core clock cycles. The scrubber uses a free slot when data traffic does not need to perform a table lookup to read and write the memories which causes a small indeterminism in the period.

3.2.9 System time-distribution

The router contains a global time-counter register which handles system time-distribution. All the different port types support time-code transmission. Incoming time-codes on the ports are checked against the time-counter which is then updated. If time-code was determined to have a count value one more modulo 64 than the previous value then a tick is generated and the time-code is forwarded to all the other ports. The time-codes are also forwarded to the FIFO and AMBA ports where they appear on their respective external interfaces. Time-codes can also be transmitted from the FIFO and AMBA ports. In that case they are also compared to the time-counter and propagated to the other ports if valid.

The current router master time-counter and control flag values can be read through the configuration port (see the time-code register in section 3.7).

In default mode the router does not check the control flags so time-codes will be accepted regardless of their value. If the TF bit in the router configuration/status register is set to 1 time-code control flag filtering is enabled and the time-codes are required to have the control flags set to “00” to be accepted, otherwise they are dropped when received.

After reset all the ports are enabled to receive and transmit time-codes. The TE bit in a port’s control register can be set to 0 to disable time-code transmission and reception on that port.

Time-code transfers can also be disabled globally using a signal.

3.2.10 Invalid address error

An invalid address error occurs when a port receives a packet with an destination address that belongs to one or more of the three following groups:

1. Destination address is a path address corresponding to a non-existing port number. For example if the router only has 8 ports and a packet has destination address 15 this error will occur. If a router has 31 ports (32 including the configuration port) this error cannot occur.

2. Destination address is a logical address corresponding to a routing table entry which has not been configured. The routing table entries start at address 0x480.

3. Destination address is a logical address corresponding to a port setup register which has not been configured. The port setup registers start at address 0x80 for logical addresses.

4. The destination port determined either through physical or logical address has the disable (DI) bit set in the port control register.

3.2.11 Global configuration features

3.2.11.1 Self addressing

Normally the ports are allowed to address themselves i.e. a packet is received on a port with a destination address configured to be transmitted on the same port (which the packet was received on). This

can be disabled by setting the self addressing enable (SA) bit in the router configuration/status register to 0. The reset value of this bit is set using a signal.

This also applies to group adaptive routing and packet distribution. When group adaptive routing is enabled for an address a packet with that destination address will be spilt due to self-addressing only if the packet is actually routed to the source port. That is if ports 1 and 2 are enabled for address 1 and a packet with address 1 arrives and it is routed to port 2 the transfer will be performed normally. If it is routed to port 1 and self-addressing is disabled it will be discarded.

For packet distribution the packet will always be discarded if the source port is included in the list of destination ports since the packet will be sent to all destinations.

3.2.11.2 Link start on request

Ports can be configured to start automatically when a packet is waiting to be transmitted on it. This is done by setting the LS bit in the router configuration/status register to 1. If the port link is disabled it will override the start feature and the link will not start. The reset value of this bit is set using a signal. This feature is only applicable for SpaceWire ports.

If the linkstart bit for the port is set the setting for the link start on request bit will have no effect. The link will continue to be started until a '0' is written to the linkstart bit of the port or if the auto disconnect feature is enabled (see next section).

3.2.11.3 Auto disconnect

If the link was started by the link start feature the auto disconnect feature can be enabled to automatically stop the link if inactive during a timeout period. The auto disconnect feature is enabled by setting the AD bit in the router configuration/status register. The reset value is set using a signal. This feature is only applicable to SpaceWire ports.

The link will be disconnected under the following conditions. The link start on request feature is enabled and the link was not in run-state when the packet arrived at the output port. Then the link will be disconnected when the packet transmission has finished (output port free), the transmit FIFO is empty, no receive operation is active and the timeout period has expired since the last of the requirements for disconnect (the ones listed here) became true.

3.3 SpaceWire ports

When a port is configured as a SpaceWire link it consists of a SpaceWire codec with FIFO interfaces. All the configurable parameters for the link are accessible through the router configuration port (see the register section for the configurable parameters).

3.4 FIFO ports

A port configured as a FIFO port contains one FIFO in each direction to/from the switch matrix.

3.4.1 Transmitter

The transmitter FIFO interface consists of the following signals: txfull, txafull, txwrite, txchar, txcharcnt. Figure 2 illustrates the write operation. Note that txfull would only be asserted as illustrated in the figure when txcharcnt is 4 if the FIFO size is 4 (which is not the case typically).

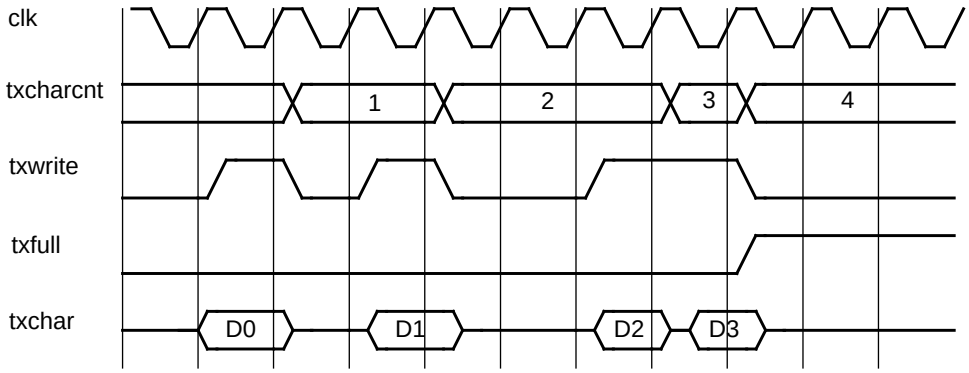


Figure 2. Transmitter FIFO interface write cycle.

Txwrite is the write signal and each time when asserted on the rising edge of the clock the value on the txchar signal will be written into the transmitter FIFO if it is not full. If it is full the character will be dropped. Txcharcnt indicates the number of characters currently in the FIFO. Txfull is asserted when the FIFO is full and txafull is asserted when the FIFO is almost full.

The transmitter FIFO can be reset through the port's control register using the TF bit.

3.4.2 Receiver

The receiver FIFO interface consists of the following signals: rxread, rxchar, rxcharav and rxaempty. Figure 3 illustrates the read operation. Note that rxcharav would only be deasserted as illustrated in the figure if the FIFO contained 4 characters.

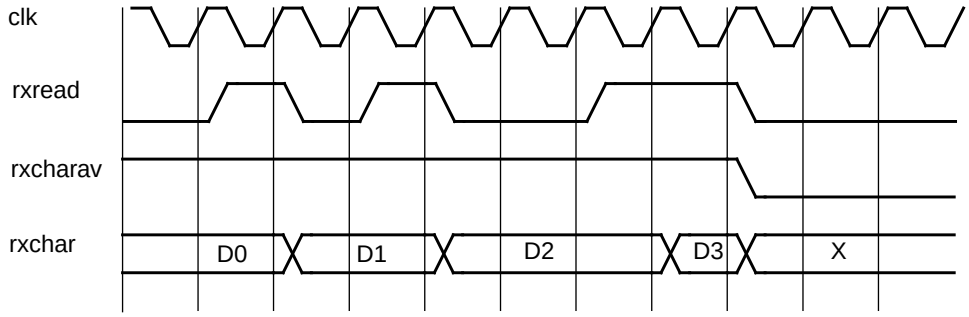


Figure 3. Receiver FIFO interface read cycle.

Each time rxread is asserted on the rising edge of the clock a new character will be available on the rxchar output the next cycle if available. If the FIFO is empty the value is undefined. Rxcharav is asserted when the FIFO contains at least one character. Rxaempty is asserted when the FIFO is almost empty.

The receiver FIFO can be reset using the RF bit in the port's control register.

3.4.3 Time-code transmit

The time-code transmit interface consists of the following signals: tickin, timein. Figure 4 illustrates the tickin operation.

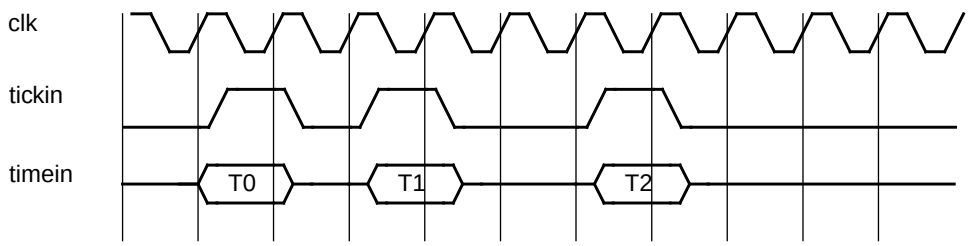


Figure 4. Time interface tickin operation.

3.4.4 Time-code receive

The time-code receive interface consists of the following signals: tickout, timeout. Figure 5 illustrates the tickout operation.

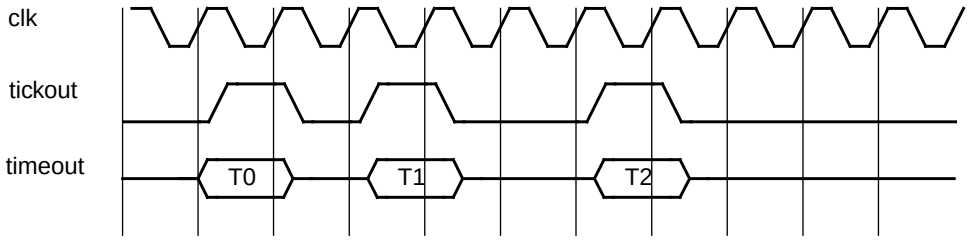


Figure 5. Time interface tickout operation.

The clock that all the interface signals are synchronized to is the same as the core clock (the clock that everything except the SpaceWire links' transmitters and receivers are running on). It can run on any frequency but to support the maximum throughput it has to be at least one eighth of the maximum link bitrate.

3.4.5 Bridge mode

The FIFO ports normally operate in standard mode which has been described so far in this section. But they can also be set in bridge mode through the bridge enable (BE) bit in the port's control register. The reset value of this signal is set through an input signal so this mode can be enabled per default after reset.

In bridge mode two FIFO ports can be connected together with automatic packet and time-code transfer without any glue logic. Table 11 shows how the signals should be mapped. Rxaempty and txafull are unused in this mode.

Table 11. Signal mappings of FIFO port in bridge mode.

Port 0	**Port 1**
rxchar	txchar
rxread	txfull
txwrite	rxcharav
txchar	rxchar
txfull	rxread
rxcharav	txwrite
tickin	tickout
timein	timeout
tickout	tickin
timeout	timein

3.5 AMBA ports

The AMBA ports consists of what is basically a GRSPW2 core with the SpaceWire codec removed. The same drivers that are provided for the GRSPW2 core can be used for each AMBA port on the router. Only an additional driver is needed which handles the setup of all the registers on the configuration port.

3.5.1 Overview

The Router AMBA port is configured through a set of registers accessed through an APB interface. Data is transferred through DMA channels using an AHB master interface.

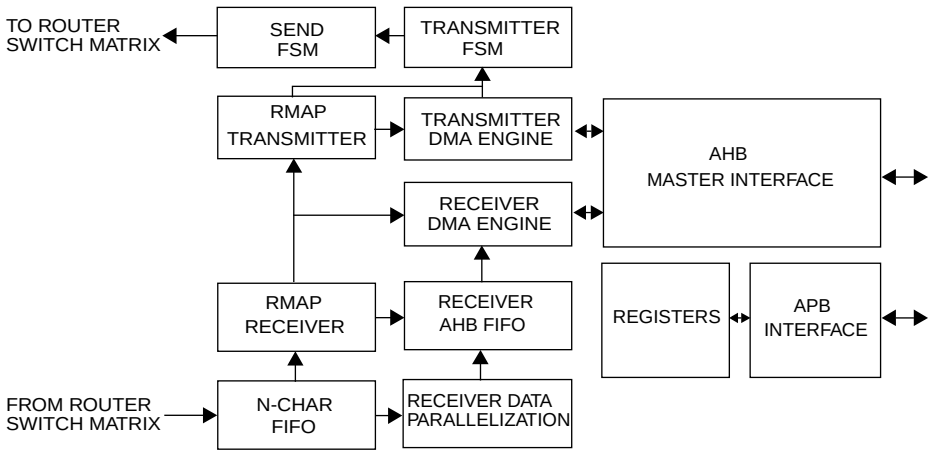


Figure 6. Block diagram of the Router DMA port

3.5.2 Operation

The main sub-blocks of the router AHB interfaces are the DMA engines, the RMAP target and the AMBA interface. A block diagram of the internal structure can be found in figure 6.

The AMBA interface is divided into the AHB master interface and the APB interface. The DMA engines have FIFO interfaces to the router switch matrix. These FIFOs are used to transfer N-Chars between the AMBA bus and the other ports in the router.

The RMAP target handles incoming packets which are determined to be RMAP commands instead of the receiver DMA engine. The RMAP command is decoded and if it is valid, the operation is performed on the AHB bus. If a reply was requested it is automatically transmitted back to the source by the RMAP transmitter.

3.5.2.1 Protocol support

The AMBA port only accepts packets with a valid destination address in the first received byte. Packets with address mismatch will be silently discarded (except in promiscuous mode which is covered in section 3.5.3.10).

The second byte is sometimes interpreted as a protocol ID a described hereafter. The RMAP protocol (ID=0x1) is the only protocol handled separately in hardware while other packets are stored to a DMA channel. If the RMAP target is present and enabled all RMAP commands will be processed, executed and replied automatically in hardware. Otherwise RMAP commands are stored to a DMA channel in the same way as other packets. RMAP replies are always stored to a DMA channel. More information on the RMAP protocol support is found in section 3.5.5 (note that this RMAP target is different from the one in the configuration port). When the RMAP target is not present or disabled, there is no need to include a protocol ID in the packets and the data can start immediately after the address.

All packets arriving with the extended protocol ID (0x00) are stored to a DMA channel. This means that the hardware RMAP target will not work if the incoming RMAP packets use the extended protocol ID. Note also that packets with the reserved extended protocol identifier (ID = 0x000000) are not ignored by the AMBA port. It is up to the client receiving the packets to ignore them.

When transmitting packets, the address and protocol-ID fields must be included in the buffers from where data is fetched. They are *not* automatically added by the AMBA port DMA engine.

Figure 7 shows the packet types accepted by the port. The port also allows reception and transmission with extended protocol identifiers but without support for RMAP CRC calculations and the RMAP target.

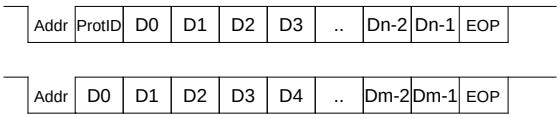


Figure 7. The SpaceWire packet types supported by the port.

3.5.2.2 Time interface

The time interface is used for sending Time-codes over the SpaceWire network and consists of a time-counter register, time-ctrl register, tick-in signal, tick-out signal, tick-in register field and a tick-out register field. There are also two control register bits which enable the time receiver and transmitter respectively.

Each Time-code sent from the sent from the port is a concatenation of the time-ctrl and the time-counter register. There is a timetxen bit which is used to enable Time-code transmissions. It is not possible to send time-codes if this bit is zero.

Received Time-codes are stored to the same time-ctrl and time-counter registers which are used for transmission. The timerxen bit in the control register is used for enabling time-code reception. No time-codes will be received if this bit is zero.

The two enable bits are used for ensuring that a node will not (accidentally) both transmit and receive time-codes which violates the SpaceWire standard. It also ensures that a master sending time-codes on a network will not have its time-counter overwritten if another (faulty) node starts sending time-codes.

The time-counter register is set to 0 after reset and is incremented each time the tick-in signal is asserted for one clock-period and the timetxen bit is set. This also causes the new value to be sent to the router (which will propagate the time-code to the other ports if valid just as if it was transmitted on a normal SpW link). Tick-in can be generated either by writing a one to the register field or by asserting the tick-in signal. A Tick-in should not be generated too often since if the time-code after the previous Tick-in has not been sent the register will not be incremented and no new value will be sent. The tick-in field is automatically cleared when the value has been sent and thus no new ticks should be generated until this field is zero. If the tick-in signal is used there should be at least 4 system-clock plus 25 transmit-clock cycles between each assertion.

A tick-out is generated each time a valid time-code is received and the timerxen bit is set. When the tick-out is generated the tick-out signal will be asserted one clock-cycle and the tick-out register field is asserted until it is cleared by writing a one to it.

The current time counter value can be read from the time register. It is updated each time a Time-code is received and the timerxen bit is set. The same register is used for transmissions and can also be written directly from the APB interface.

The control bits of the Time-code are stored to the time-ctrl register when a Time-code is received whose time-count is one more than the nodes current time-counter register. The time-ctrl register can be read through the APB interface. The same register is used during time-code transmissions.

It is possible to have both the time-transmission and reception functions enabled at the same time.

3.5.3 Receiver DMA channels

The receiver DMA engine handles reception of data from the SpaceWire network to different DMA channels.

3.5.3.1 Address comparison and channel selection

Packets are received to different channels based on the address and whether a channel is enabled or not. When the receiver N-Char FIFO contains one or more characters, N-Chars are read by the receiver DMA engine. The first character is interpreted as the logical address and is compared with the addresses of each channel starting from 0. The packet will be stored to the first channel with an matching address. The complete packet including address and protocol ID but excluding EOP/EEP is stored to the memory address pointed to by the descriptors (explained later in this section) of the channel.

Each SpaceWire address register has a corresponding mask register. Only bits at an index containing a zero in the corresponding mask register are compared. This way a DMA channel can accept a range of addresses. There is a default address register which is used for address checking in all implemented DMA channels that do not have separate addressing enabled and for RMAP commands in the RMAP target. With separate addressing enabled the DMA channels' own address/mask register pair is used instead.

If an RMAP command is received it is only handled by the target if the default address register (including mask) matches the received address. Otherwise the packet will be stored to a DMA channel if one or more of them has a matching address. If the address does not match neither the default address nor one of the DMA channels' separate register, the packet is still handled by the RMAP target if enabled since it has to return the invalid address error code. The packet is only discarded (up to and including the next EOP/EEP) if an address match cannot be found and the RMAP target is disabled.

Packets, other than RMAP commands, that do not match neither the default address register nor the DMA channels' address register will be discarded. Figure 8 shows a flowchart of packet reception.

At least 2 non EOP/EEP N-Chars needs to be received for a packet to be stored to the DMA channel unless the promiscuous mode is enabled in which case 1 N-Char is enough. If it is an RMAP packet with hardware RMAP enabled 3 N-Chars are needed since the command byte determines where the packet is processed. Packets smaller than these sizes are discarded.

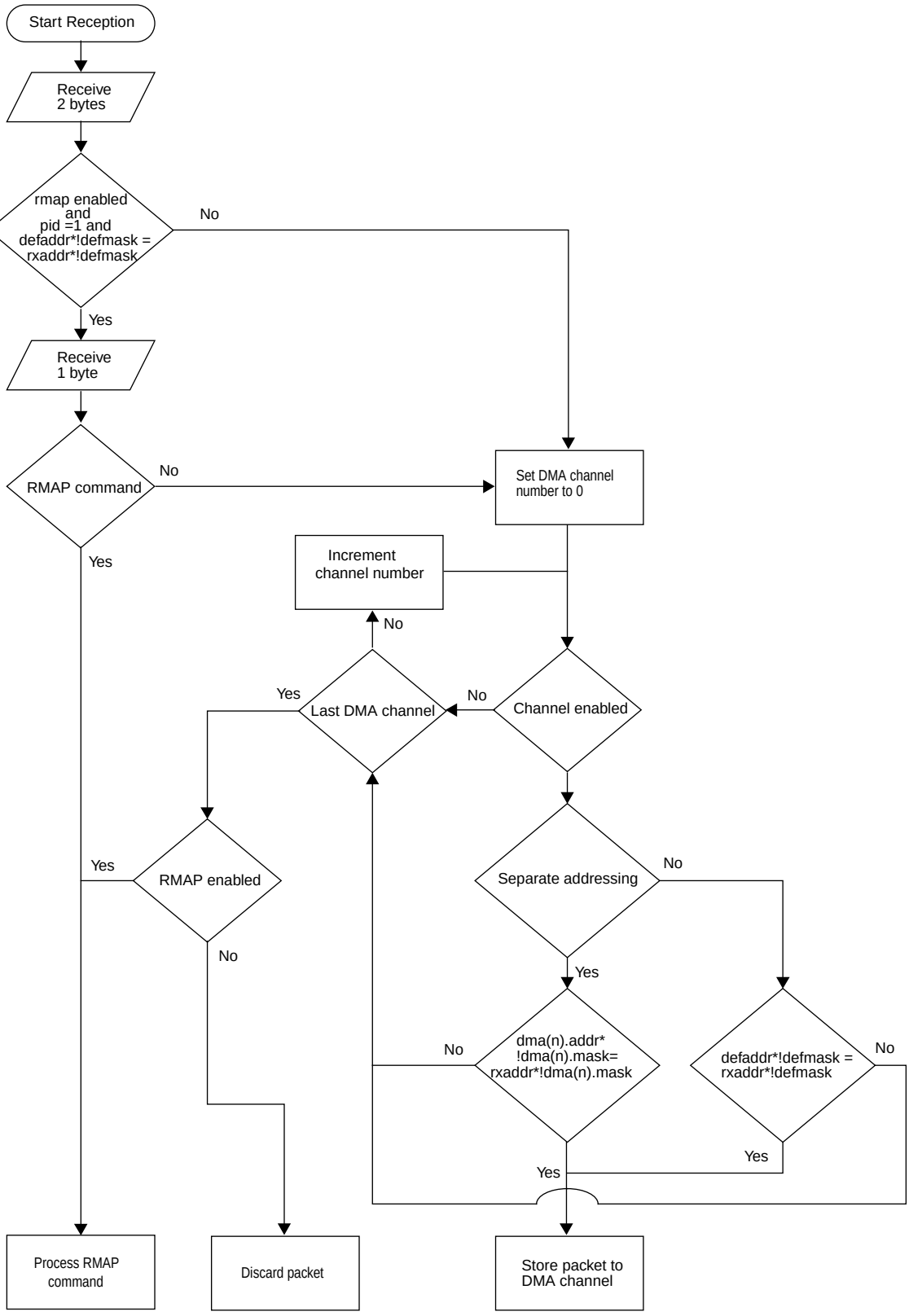


Figure 8. Flow chart of packet reception (promiscuous mode disabled).

3.5.3.2 Basic functionality of a channel

Reception is based on descriptors located in a consecutive area in memory that hold pointers to buffers where packets should be stored. When a packet arrives at the port the channel which should receive it is first determined as described in the previous section. A descriptor is then read from the channels' descriptor area and the packet is stored to the memory area pointed to by the descriptor. Lastly, status is stored to the same descriptor and increments the descriptor pointer to the next one. The following sections will describe DMA channel reception in more detail.

3.5.3.3 Setting up the port for reception

A few registers need to be initialized before reception to a channel can take place. The DMA channel has a maximum length register which sets the maximum packet size in bytes that can be received to this channel. Larger packets are truncated and the excessive part is spilled. If this happens an indication will be given in the status field of the descriptor. The minimum value for the receiver maximum length field is 4 and the value can only be incremented in steps of four bytes up to the maximum value 33554428. If the maximum length is set to zero the receiver will *not* function correctly.

Either the default address register or the channel specific address register (the accompanying mask register must also be set) needs to be set to hold the address used by the channel. A control bit in the DMA channel control register determines whether the channel should use default address and mask registers for address comparison or the channel's own registers. Using the default register the same address range is accepted as for other channels with default addressing and the RMAP target while the separate address provides the channel its own range. If all channels use the default registers they will accept the same address range and the enabled channel with the lowest number will receive the packet.

Finally, the descriptor table and control register must be initialized. This will be described in the two following sections.

3.5.3.4 Setting up the descriptor table address

The port reads descriptors from an area in memory pointed to by the receiver descriptor table address register. The register consists of a base address and a descriptor selector. The base address points to the beginning of the area and must start on a 1024 bytes aligned address. It is also limited to be 1024 bytes in size which means the maximum number of descriptors is 128 since the descriptor size is 8 bytes.

The descriptor selector points to individual descriptors and is increased by 1 when a descriptor has been used. When the selector reaches the upper limit of the area it wraps to the beginning automatically. It can also be set to wrap at a specific descriptor before the upper limit by setting the wrap bit in the descriptor. The idea is that the selector should be initialized to 0 (start of the descriptor area) but it can also be written with another 8 bytes aligned value to start somewhere in the middle of the area. It will still wrap to the beginning of the area.

If one wants to use a new descriptor table the receiver enable bit has to be cleared first. When the rxactive bit for the channel is cleared it is safe to update the descriptor table register. When this is finished and descriptors are enabled the receiver enable bit can be set again.

3.5.3.5 Enabling descriptors

As mentioned earlier one or more descriptors must be enabled before reception can take place. Each descriptor is 8 byte in size and the layout can be found in the tables below. The descriptors should be written to the memory area pointed to by the receiver descriptor table address register. When new

descriptors are added they must always be placed after the previous one written to the area. Otherwise they will not be noticed.

A descriptor is enabled by setting the address pointer to point at a location where data can be stored and then setting the enable bit. The WR bit can be set to cause the selector to be set to zero when reception has finished to this descriptor. IE should be set if an interrupt is wanted when the reception has finished. The DMA control register interrupt enable bit must also be set for an interrupt to be generated.

Table 12. RXDMA receive descriptor word 0 (address offset 0x0)

31	30	29	28	27	26	25	24 ... 0
TR	DC	HC	EP	IE	WR	EN	PACKETLENGTH

31	Truncated (TR) - Packet was truncated due to maximum length violation.
30	Data CRC (DC) - 1 if a CRC error was detected for the data and 0 otherwise.
29	Header CRC (HC) - 1 if a CRC error was detected for the header and 0 otherwise.
28	EEP termination (EP) - This packet ended with an Error End of Packet character.
27	Interrupt enable (IE) - If set, an interrupt will be generated when a packet has been received if the receive interrupt enable bit in the DMA channel control register is set.
26	Wrap (WR) - If set, the next descriptor used by the GRSPW will be the first one in the descriptor table (at the base address). Otherwise the descriptor pointer will be increased with 0x8 to use the descriptor at the next higher memory location. The descriptor table is limited to 1 kbytes in size and the pointer will be automatically wrap back to the base address when it reaches the 1 kbytes boundary.
25	Enable (EN) - Set to one to activate this descriptor. This means that the descriptor contains valid control values and the memory area pointed to by the packet address field can be used to store a packet.
24: 0	Packet length (PACKETLENGTH) - The number of bytes received to this buffer. Only valid after EN has been set to 0 by the GRSPW.

Table 13. RXDMA receive descriptor word 1 (address offset 0x4)

31 ... 0
PACKETADDRESS

31: 0	Packet address (PACKETADDRESS) - The address pointing at the buffer which will be used to store the received packet.

3.5.3.6 Setting up the DMA control register

The final step to receive packets is to set the control register in the following steps: The receiver must be enabled by setting the rxen bit in the DMA control register. This can be done anytime and before this bit is set nothing will happen. The rxdescav bit in the DMA control register is then set to indicate that there are new active descriptors. This must always be done after the descriptors have been enabled or the port might not notice the new descriptors. More descriptors can be activated when reception has already started by enabling the descriptors and writing the rxdescav bit. When these bits are set reception will start immediately when data is arriving.

3.5.3.7 The effect to the control bits during reception

When the receiver is disabled all packets going to the DMA-channel are discarded if the packet's address does not fall into the range of another DMA channel. If the receiver is enabled and the address falls into the accepted address range, the next state is entered where the rxdescav bit is checked. This bit indicates whether there are active descriptors or not and should be set by the external application

using the DMA channel each time descriptors are enabled as mentioned above. If the rxdescav bit is ‘0’ and the nospill bit is ‘0’ the packets will be discarded. If nospill is one the grspw waits until rxdescav is set and the characters are kept in the N-Char fifo during this time. If the fifo becomes full further N-char transmissions are inhibited by stopping the transmission of FCTs.

When rxdescav is set the next descriptor is read and if enabled the packet is received to the buffer. If the read descriptor is not enabled, rxdescav is set to ‘0’ and the packet is spilled depending on the value of nospill.

The receiver can be disabled at any time and will stop packets from being received to this channel. If a packet is currently received when the receiver is disabled the reception will still be finished. The rxdescav bit can also be cleared at any time. It will not affect any ongoing receptions but no more descriptors will be read until it is set again. Rxdescav is also cleared by the port when it reads a disabled descriptor.

3.5.3.8 Status bits

When the reception of a packet is finished the enable bit in the current descriptor is set to zero. When enable is zero, the status bits are also valid and the number of received bytes is indicated in the length field. The DMA control register contains a status bit which is set each time a packet has been received. The port can also be made to generate an interrupt for this event.

The RMAP CRC calculation is always active for all received packets and all bytes except the EOP/EEP are included. The packet is always assumed to be a RMAP packet and the length of the header is determined by checking byte 3 which should be the command field. The calculated CRC value is then checked when the header has been received (according to the calculated number of bytes) and if it is non-zero the HC bit is set indicating a header CRC error.

The CRC value is not set to zero after the header has been received, instead the calculation continues in the same way until the complete packet has been received. Then if the CRC value is non-zero the DC bit is set indicating a data CRC error. This means that the port can indicate a data CRC error even if the data field was correct when the header CRC was incorrect. However, the data should not be used when the header is corrupt and therefore the DC bit is unimportant in this case. When the header is not corrupted the CRC value will always be zero when the calculation continues with the data field and the behaviour will be as if the CRC calculation was restarted

If the received packet is not of RMAP type the header CRC error indication bit cannot be used. It is still possible to use the DC bit if the complete packet is covered by a CRC calculated using the RMAP CRC definition. This is because the port does not restart the calculation after the header has been received but instead calculates a complete CRC over the packet. Thus any packet format with one CRC at the end of the packet calculated according to RMAP standard can be checked using the DC bit.

If the packet is neither of RMAP type nor of the type above with RMAP CRC at the end, then both the HC and DC bits should be ignored.

3.5.3.9 Error handling

If an AHB error occurs during reception the current packet is spilled up to and including the next EEP/EOP and then the currently active channel is disabled and the receiver enters the idle state. A bit in the channels control/status register is set to indicate this condition.

3.5.3.10 Promiscuous mode

The port supports a promiscuous mode where all the data received is stored to the first DMA channel enabled regardless of the node address and possible early EOPs/EEPs. This means that all non-eop/eep N-Chars received will be stored to the DMA channel. The rxmaxlength register is still checked and packets exceeding this size will be truncated.

RMAP commands will still be handled by it when promiscuous mode is enabled if the rmapen bit is set. If it is cleared, RMAP commands will also be stored to a DMA channel.

3.5.4 Transmitter DMA channels

The transmitter DMA engine handles transmission of data from the DMA channels to the SpaceWire network. Each receive channel has a corresponding transmit channel which means there can be up to 4 transmit channels. It is however only necessary to use a separate transmit channel for each receive channel if there are also separate entities controlling the transmissions. The use of a single channel with multiple controlling entities would cause them to corrupt each other's transmissions. A single channel is more efficient and should be used when possible.

Multiple transmit channels with pending transmissions are arbitrated in a round-robin fashion.

3.5.4.1 Basic functionality of a channel

A transmit DMA channel reads data from the AHB bus and stores them in the transmitter FIFO for transmission on the SpaceWire network. Transmission is based on the same type of descriptors as for the receiver and the descriptor table has the same alignment and size restrictions. When there are new descriptors enabled the port reads them and transfer the amount data indicated.

3.5.4.2 Setting up the core for transmission

Four steps need to be performed before transmissions can be done with the port. First the link interface must be enabled and started by writing the appropriate value to the ctrl register. Then the address to the descriptor table needs to be written to the transmitter descriptor table address register and one or more descriptors must also be enabled in the table. Finally, the txen bit in the DMA control register is written with a one which triggers the transmission. These steps will be covered in more detail in the next sections.

3.5.4.3 Enabling descriptors

The descriptor table address register works in the same way as the receiver's corresponding register which was covered in section 3.5.3. The maximum size is 1024 bytes as for the receiver but since the descriptor size is 16 bytes the number of descriptors is 64.

To transmit packets one or more descriptors have to be initialized in memory which is done in the following way: The number of bytes to be transmitted and a pointer to the data has to be set. There are two different length and address fields in the transmit descriptors because there are separate pointers for header and data. If a length field is zero the corresponding part of a packet is skipped and if both are zero no packet is sent. The maximum header length is 255 bytes and the maximum data length is 16 Mbyte - 1. When the pointer and length fields have been set the enable bit should be set to enable the descriptor. This must always be done last. The other control bits must also be set before enabling the descriptor.

The transmit descriptors are 16 bytes in size so the maximum number in a single table is 64. The different fields of the descriptor together with the memory offsets are shown in the tables below.

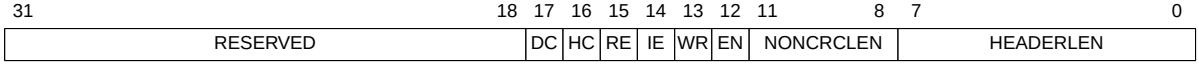
The HC bit should be set if RMAP CRC should be calculated and inserted for the header field and correspondingly the DC bit should be set for the data field. The header CRC will be calculated from the data fetched from the header pointer and the data CRC is generated from data fetched from the data pointer. The CRCs are appended after the corresponding fields. The NON-CRC bytes field is set to the number of bytes in the beginning of the header field that should not be included in the CRC calculation.

The CRCs are sent even if the corresponding length is zero, but when both lengths are zero no packet is sent not even an EOP.

3.5.4.4 Starting transmissions

When the descriptors have been initialized, the transmit enable bit in the DMA control register has to be set to tell the port to start transmitting. New descriptors can be activated in the table on the fly (while transmission is active). Each time a set of descriptors is added the transmit enable register bit should be set. This has to be done because each time the core encounters a disabled descriptor this register bit is set to 0.

Table 14. TXDMA transmit descriptor word 0 (address offset 0x0)



31: 18	RESERVED
17	Append data CRC (DC) - Append CRC calculated according to the RMAP specification after the data sent from the data pointer. The CRC covers all the bytes from this pointer. A null CRC will be sent if the length of the data field is zero.
16	Append header CRC (HC) - Append CRC calculated according to the RMAP specification after the data sent from the header pointer. The CRC covers all bytes from this pointer except a number of bytes in the beginning specified by the non-crc bytes field. The CRC will not be sent if the header length field is zero.
15	RESERVED
14	Interrupt enable (IE) - If set, an interrupt will be generated when the packet has been transmitted and the transmitter interrupt enable bit in the DMA control register is set.
13	Wrap (WR) - If set, the descriptor pointer will wrap and the next descriptor read will be the first one in the table (at the base address). Otherwise the pointer is increased with 0x10 to use the descriptor at the next higher memory location.
12	Enable (EN) - Enable transmitter descriptor. When all control fields (address, length, wrap and crc) are set, this bit should be set. While the bit is set the descriptor should not be touched since this might corrupt the transmission. The GRSPW clears this bit when the transmission has finished.
11: 8	Non-CRC bytes (NONCRCLEN)- Sets the number of bytes in the beginning of the header which should not be included in the CRC calculation. This is necessary when using path addressing since one or more bytes in the beginning of the packet might be discarded before the packet reaches its destination.
7: 0	Header length (HEADERLEN) - Header Length in bytes. If set to zero, the header is skipped.

Table 15. TXDMA transmit descriptor word 1 (address offset 0x4)

31: 0	Header address (HEADERADDRESS) - Address from where the packet header is fetched. Does not need to be word aligned.

Table 16. TXDMA transmit descriptor word 2 (address offset 0x8)

31	24	23	0
RESERVED		DATALEN	

31: 24	RESERVED
23: 0	Data length (DATALEN) - Length of data part of packet. If set to zero, no data will be sent. If both data- and header-lengths are set to zero no packet will be sent.

Table 17. TXDMA transmit descriptor word 3(address offset 0xC)

31	0
DATAADDRESS	

31: 0	Data address (DATAADDRESS) - Address from where data is read. Does not need to be word aligned.

3.5.4.5 The transmission process

When the txen bit is set the port starts reading descriptors immediately. The number of bytes indicated are read and transmitted. When a transmission has finished, status will be written to the first field of the descriptor and a packet sent bit is set in the DMA control register. If an interrupt was requested it will also be generated. Then a new descriptor is read and if enabled a new transmission starts, otherwise the transmit enable bit is cleared and nothing will happen until it is enabled again.

3.5.4.6 The descriptor table address register

The internal pointer which is used to keep the current position in the descriptor table can be read and written through the APB interface. This pointer is set to zero during reset and is incremented each time a descriptor is used. It wraps automatically when the 1024 bytes limit for the descriptor table is reached or it can be set to wrap earlier by setting a bit in the current descriptor.

The descriptor table register can be updated with a new table anytime when no transmission is active. No transmission is active if the transmit enable bit is zero and the complete table has been sent or if the table is aborted (explained below). If the table is aborted one has to wait until the transmit enable bit is zero before updating the table pointer.

3.5.4.7 Error handling

3.5.4.7.1 Abort Tx

The DMA control register contains a bit called Abort TX which if set causes the current transmission to be aborted, the packet is truncated and an EEP is inserted. This is only useful if the packet needs to be aborted because of congestion on the SpaceWire network. If the congestion is on the AHB bus this will not help (This should not be a problem since AHB slaves should have a maximum of 16 wait-states). The aborted packet will have its LE bit set in the descriptor. The transmit enable register bit is also cleared and no new transmissions will be done until the transmitter is enabled again.

3.5.4.7.2 AHB error

When an AHB error is encountered during transmission the currently active DMA channel is disabled and the transmitter goes to the idle mode. A bit in the DMA channel's control/status register is set to indicate this error condition and, if enabled, an interrupt will also be generated. Further error handling depends on what state the transmitter DMA engine was in when the AHB error occurred. If the descriptor was being read the packet transmission had not been started yet and no more actions need to be taken.

If the AHB error occurs during packet transmission the packet is truncated and an EEP is inserted. Lastly, if it occurs when status is written to the descriptor the packet has been successfully transmitted but the descriptor is not written and will continue to be enabled (this also means that no error bits are set in the descriptor for AHB errors).

The client using the channel has to correct the AHB error condition and enable the channel again. No more AHB transfers are done again from the same unit (receiver or transmitter) which was active during the AHB error until the error state is cleared and the unit is enabled again.

3.5.5 RMAP target

The Remote Memory Access Protocol (RMAP) is used to implement access to resources on the AHB bus via the SpaceWire Link. Some common operations are reading and writing to memory, registers and FIFOs. The port has an optional hardware RMAP target. This section describes the target implementation.

3.5.5.1 Fundamentals of the protocol

RMAP is a protocol which is designed to provide remote access via a SpaceWire network to memory mapped resources on a SpaceWire node. It has been assigned protocol ID 0x01. It provides three operations write, read and read-modify-write. These operations are posted operations which means that a source does not wait for an acknowledge or reply. It also implies that any number of operations can be outstanding at any time and that no timeout mechanism is implemented in the protocol. Timeouts must be implemented in the user application which sends the commands. Data payloads of up to 16 Mb - 1 is supported in the protocol. A destination can be requested to send replies and to verify data before executing an operation. A complete description of the protocol is found in the RMAP standard.

3.5.5.2 Implementation

The port includes a target for RMAP commands which processes all incoming packets with protocol ID = 0x01, type field (bit 7 and 6 of the 3rd byte in the packet) equal to 01b and an address falling in the range set by the default address and mask register. When such a packet is detected it is not stored to the DMA channel, instead it is passed to the RMAP receiver.

The target implements all three commands defined in the standard with some restrictions. Support is only provided for 32-bit big-endian systems. This means that the first byte received is the msb in a word. The target will not receive RMAP packets using the extended protocol ID which are always dumped to the DMA channel.

The RMAP receiver processes commands. If they are correct and accepted the operation is performed on the AHB bus and a reply is formatted. If an acknowledge is requested the RMAP transmitter automatically send the reply. RMAP transmissions have priority over DMA channel transmissions.

There is a user accessible destination key register which is compared to destination key field in incoming packets. If there is a mismatch and a reply has been requested the error code in the reply is set to 3. Replies are sent if and only if the ack field is set to '1'.

When a failure occurs during a bus access the error code is set to 1 (General Error). There is predetermined order in which error-codes are set in the case of multiple errors in the core. It is shown in table 31.

Table 18. The order of error detection in case of multiple errors in the GRSPW. The error detected first has number 1.

Detection Order	Error Code	Error
1	12	Invalid destination logical address
2	2	Unused RMAP packet type or command code
3	3	Invalid destination key
4	9	Verify buffer overrun
5	11	RMW data length error
6	10	Authorization failure
7*	1	General Error (AHB errors during non-verified writes)
8	5/7	Early EOP / EEP (if early)
9	4	Invalid Data CRC
10	1	General Error (AHB errors during verified writes or RMW)
11	7	EEP
12	6	Too Much Data
*The AHB error is not guaranteed to be detected before Early EOP/EEP or Invalid Data CRC. For very long accesses the AHB error detection might be delayed causing the other two errors to appear first.		

Read accesses are performed on the fly, that is they are not stored in a temporary buffer before transmitting. This means that the error code 1 will never be seen in a read reply since the header has already been sent when the data is read. If the AHB error occurs the packet will be truncated and ended with an EEP.

Errors up to and including Invalid Data CRC (number 8) are checked before verified commands. The other errors do not prevent verified operations from being performed.

The details of the support for the different commands are now presented. All defined commands which are received but have an option set which is not supported in this specific implementation will not be executed and a possible reply is sent with error code 10.

3.5.5.3 Write commands

The write commands are divided into two subcategories when examining their capabilities: verified writes and non-verified writes. Verified writes have a length restriction of 4 bytes and the address must be aligned to the size. That is 1 byte writes can be done to any address, 2 bytes must be halfword aligned, 3 bytes are not allowed and 4 bytes writes must be word aligned. Since there will always be only on AHB operation performed for each RMAP verified write command the incrementing address bit can be set to any value.

Non-verified writes have no restrictions when the incrementing bit is set to 1. If it is set to 0 the number of bytes must be a multiple of 4 and the address word aligned. There is no guarantee how many words will be written when early EOP/EEP is detected for non-verified writes.

3.5.5.4 Read commands

Read commands are performed on the fly when the reply is sent. Thus if an AHB error occurs the packet will be truncated and ended with an EEP. There are no restrictions for incrementing reads but non-incrementing reads have the same alignment restrictions as non-verified writes. Note that the “Authorization failure” error code will be sent in the reply if a violation was detected even if the length field was zero. Also note that no data is sent in the reply if an error was detected i.e. if the status field is non-zero.

3.5.5.5 RMW commands

All read-modify-write sizes are supported except 6 which would have caused 3 B being read and written on the bus. The RMW bus accesses have the same restrictions as the verified writes. As in the verified write case, the incrementing bit can be set to any value since only one AHB bus operation will be performed for each RMW command. Cargo too large is detected after the bus accesses so this error will not prevent the operation from being performed. No data is sent in a reply if an error is detected i.e. the status field is non-zero.

3.5.5.6 Control

The RMAP target mostly runs in the background without any external intervention, but there are a few control possibilities.

There is an enable bit in the control register of the core which can be used to completely disable the RMAP target. When it is set to ‘0’ no RMAP packets will be handled in hardware, instead they are all stored to the DMA channel.

There is a possibility that RMAP commands will not be performed in the order they arrive. This can happen if a read arrives before one or more writes. Since the target stores replies in a buffer with more than one entry several commands can be processed even if no replies are sent. Data for read replies is read when the reply is sent and thus writes coming after the read might have been performed already if there was congestion in the transmitter. To avoid this the RMAP buffer disable bit can be set to force the target to only use one buffer which prevents this situation.

The last control option for the target is the possibility to set the destination key which is found in a separate register.

3.5.6 AMBA interface

The AMBA interface consists of an APB interface, an AHB master interface and DMA FIFOs. The APB interface provides access to the user registers. The DMA engines have 32-bit wide FIFOs to the AHB master interface which are used when reading and writing to the bus.

The transmitter DMA engine reads data from the bus in bursts which are half the FIFO size in length. A burst is always started when the FIFO is half-empty or if it can hold the last data for the packet. The burst containing the last data might have shorter length if the packet is not an even number of bursts in size.

The receiver DMA works in the same way except that it checks if the FIFO is half-full and then performs a burst write to the bus which is half the fifo size in length. The last burst might be shorter. Byte accesses are used for non word-aligned buffers and/or packet lengths that are not a multiple of four bytes. There might be 1 to 3 single byte writes when writing the beginning and end of the received packets.

Table 19. AMBA port hardware RMAP handling of different packet type and command fields.

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Command**	**Action**
Reserved	**Command / Response**	**Write / Read**	**Verify data before write**	**Acknow-ledge**	**Increment Address**		
0	0	-	-	-	-	Response	Stored to DMA-channel.
0	1	0	0	0	0	Not used	Does nothing. No reply is sent.
0	1	0	0	0	1	Not used	Does nothing. No reply is sent.
0	1	0	0	1	0	Read single address	Executed normally. Address has to be word aligned and data size a multiple of four. Reply is sent. If alignment restrictions are violated error code is set to 10.
0	1	0	0	1	1	Read incre-menting address.	Executed normally. No restric-tions. Reply is sent.
0	1	0	1	0	0	Not used	Does nothing. No reply is sent.
0	1	0	1	0	1	Not used	Does nothing. No reply is sent.
0	1	0	1	1	0	Not used	Does nothing. Reply is sent with error code 2.
0	1	0	1	1	1	Read-Mod-ify-Write increment-ing address	Executed normally. If length is not one of the allowed rmw val-ues nothing is done and error code is set to 11. If the length was correct, alignment restric-tions are checked next. 1 byte can be rmw to any address. 2 bytes must be halfword aligned. 3 bytes are not allowed. 4 bytes must be word aligned. If these restrictions are violated nothing is done and error code is set to 10. If an AHB error occurs error code is set to 1. Reply is sent.
0	1	1	0	0	0	Write, sin-gle-address, do not verify before writ-ing, no acknowledge	Executed normally. Address has to be word aligned and data size a multiple of four. If alignment is violated nothing is done. No reply is sent.
0	1	1	0	0	1	Write, incre-menting address, do not verify before writ-ing, no acknowledge	Executed normally. No restric-tions. No reply is sent.

Table 19. AMBA port hardware RMAP handling of different packet type and command fields.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Command	Action
Reserved	Command / Response	Write / Read	Verify data before write	Acknow-ledge	Increment Address		
0	1	1	0	1	0	Write, single-address, do not verify before writing, send acknowledge	Executed normally. Address has to be word aligned and data size a multiple of four. If alignment is violated nothing is done and error code is set to 10. If an AHB error occurs error code is set to 1. Reply is sent.
0	1	1	0	1	1	Write, incrementing address, do not verify before writing, send acknowledge	Executed normally. No restrictions. If AHB error occurs error code is set to 1. Reply is sent.
0	1	1	1	0	0	Write, single address, verify before writing, no acknowledge	Executed normally. Length must be 4 or less. Otherwise nothing is done. Same alignment restrictions apply as for rmw. No reply is sent.
0	1	1	1	0	1	Write, incrementing address, verify before writing, no acknowledge	Executed normally. Length must be 4 or less. Otherwise nothing is done. Same alignment restrictions apply as for rmw. If they are violated nothing is done. No reply is sent.
0	1	1	1	1	0	Write, single address, verify before writing, send acknowledge	Executed normally. Length must be 4 or less. Otherwise nothing is done and error code is set to 9. Same alignment restrictions apply as for rmw. If they are violated nothing is done and error code is set to 10. If an AHB error occurs error code is set to 1. Reply is sent.
0	1	1	1	1	1	Write, incrementing address, verify before writing, send acknowledge	Executed normally. Length must be 4 or less. Otherwise nothing is done and error code is set to 9. Same alignment restrictions apply as for rmw. If they are violated nothing is done and error code is set to 10. If an AHB error occurs error code is set to 1. Reply is sent.
1	0	-	-	-	-	Unused	Stored to DMA-channel.
1	1	-	-	-	-	Unused	Stored to DMA-channel.

3.5.6.1 APB slave interface

As mentioned above, the APB interface provides access to the user registers which are 32-bits in width. The accesses to this interface are required to be aligned word accesses. The result is undefined if this restriction is violated.

3.5.6.2 AHB master interface

The port contains a single master interface which is used by both the transmitter and receiver DMA engines. The arbitration algorithm between the channels is done so that if the current owner requests the interface again it will always acquire it. This will not lead to starvation problems since the DMA engines always deassert their requests between accesses.

The AHB accesses can be of size byte, halfword and word (HSIZE = 0x000, 0x001, 0x010). Byte and halfword accesses are always NONSEQ.

The burst length will be half the AHB FIFO size except for the last transfer for a packet which might be smaller. Shorter accesses are also done during descriptor reads and status writes.

The AHB master also supports non-incrementing accesses where the address will be constant for several consecutive accesses. HTRANS will always be NONSEQ in this case while for incrementing accesses it is set to SEQ after the first access. This feature is included to support non-incrementing reads and writes for RMAP.

If the core does not need the bus after a burst has finished there will be one wasted cycle (HTRANS = IDLE).

BUSY transfer types are never requested and the port provides full support for ERROR, RETRY and SPLIT responses.

3.5.6.3 Clocking

The AMBA ports run on the same clock as the router switch matrix.

3.5.7 Registers

The port is programmed through registers mapped into APB address space. The addresses in the table below are offsets from each port's base address. The actual AMBA AHB address used to access the port is determined as follows: The AMBA ports' registers are accessed through an APB interface which resides on the APB bus. The APB bus is connected to the AHB bus using an APB controller whose AHB address determines the first base address for the AMBA ports.

Table 20. AMBA port registers

APB address offset	**Register**
0x0	Control
0x4	Status/Interrupt-source
0x8	Default address
0xC	Reserved
0x10	Destination key
0x14	Time
0x20	DMA channel 1 control/status
0x24	DMA channel 1 rx maximum length
0x28	DMA channel 1 transmit descriptor table address.
0x2C	DMA channel 1 receive descriptor table address.
0x30	DMA channel 1 address register
0x34	Unused
0x38	Unused
0x3C	Unused
0x40 - 0x5C	DMA channel 2 registers
0x60 - 0x7C	DMA channel 3 registers
0x80 - 0x9C	DMA channel 4 registers

Table 21. AMBA port control register

31	30	29	28 27	26 25 24 23 22 21 20 19 18	17	16	15 14 13 12	11	10	9	8	7	6	5	4	3	2 1 0
RA	RX	RC	NCH	RESERVED	RD	RE	RESERVED	TR	TT		TQ		RS	PM	TI	IE	RESERVED
NA	NA	NA	NA	NA	0	*	NA	0	0	NA	0	NA	0	0	0	0	NA

31	RMAP available (RA) - Set to one if the RMAP target is available.	r
30	RX unaligned access (RX) - Set to one if unaligned writes are available for the receiver.	r
29	RMAP CRC available (RC) - Set to one if RMAP CRC is enabled in the core.	r
28: 27	Number of DMA channels (NCH) - The number of available DMA channels minus one (Number of channels = NCH+1).	r
26: 18	RESERVED	r
17	RMAP buffer disable (RD) - If set only one RMAP buffer is used. This ensures that all RMAP commands will be executed consecutively.	rw
16	RMAP Enable (RE) - Enable RMAP target.	rw
15: 12	RESERVED	r
11	Time Rx Enable (TR) - Enable time-code receptions.	rw
10	Time Tx Enable (TT) - Enable time-code transmissions.	rw
9	RESERVED	r
8	Tick-out IRQ (TQ) - Generate interrupt when a valid time-code is received.	rw
7	RESERVED	t
6	Reset (RS) - Make complete reset of the SpaceWire node. Self clearing.	rw
5	Promiscuous Mode (PM) - Enable Promiscuous mode.	rw
4	Tick In (TI) - The host can generate a tick by writing a one to this field. This will increment the timer counter and the new value is transmitted after the current character is transferred. A tick can also be generated by asserting the tick_in signal.	rw
3	Interrupt Enable (IE) - If set, an interrupt is generated when bit 8 is set and its corresponding event occurs.	rw
2: 0	RESERVED	r

Table 22. AMBA port status register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9	8	7	6 5 4 3 2 1	0
RESERVED	EE	IA	RESERVED	TO
NA	0	0	NA	0

31: 9	RESERVED	r
8	Early EOP/EEP (EE) - Set to one when a packet is received with an EOP after the first byte for a non-rmap packet and after the second byte for a RMAP packet.	wc
7	Invalid Address (IA) - Set to one when a packet is received with an invalid destination address field, i.e it does not match the nodeaddr register.	wc
6: 1	RESERVED	r
0	Tick Out (TO) - A new time count value was received and is stored in the time counter field.	wc

Table 23. AMBA port default address register

31 16	15 8	7 0
RESERVED	DEFMASK	DEFADDR
NA	0x00	0xFE

31: 8	RESERVED	r
15: 8	Default mask (DEFMASK) - Default mask used for node identification on the SpaceWire network. This field is used for masking the address before comparison. Both the received address and the DEFADDR field are anded with the inverse of DEFMASK before the address check.	rw
7: 0	Default address (DEFADDR) - Default address used for node identification on the SpaceWire network. Reset value: 254.	rw

Table 24. AMBA port destination key

31 8	7 0
RESERVED	DESTKEY
NA	0x00

31: 8	RESERVED	r
7: 0	Destination key (DESTKEY) - RMAP destination key.	rw

Table 25. AMBA port time register

31 8	7 6	5 0
RESERVED	TCTRL	TIMECNT
NA	00	0x00

31: 8	RESERVED	r
7: 6	Time control flags (TCTRL) - The current value of the time control flags. Sent with time-code resulting from a tick-in. Received control flags are also stored in this register.	rw
5: 0	Time counter (TIMECNT) - The current value of the system time counter. It is incremented for each tick-in and the incremented value is transmitted. The register can also be written directly but the written value will not be transmitted. Received time-counter values are also stored in this register	rw

Table 26. AMBA port dma control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SP	SA	EN	NS	RD	RX	AT	RA	TA	PR	PS	AI	RI	TI	RE	TE
NA																0	0	0	0	0	NA	0	0	0	0	0	NR	NR	NR	0	0

31: 16	RESERVED	r
15	Strip pid (SP) - Remove the pid byte (second byte) of each packet. The address byte (first byte) will also be removed when this bit is set independent of the SA bit.	rw
14	Strip addr (SA) - Remove the addr byte (first byte) of each packet.	rw
13	Enable addr (EN) - Enable separate node address for this channel.	rw
12	No spill (NS) - If cleared, packets will be discarded when a packet is arriving and there are no active descriptors. If set, the GRSPW will wait for a descriptor to be activated.	rw
11	Rx descriptors available (RD) - Set to one, to indicate to the GRSPW that there are enabled descriptors in the descriptor table. Cleared by the GRSPW when it encounters a disabled descriptor:	rw
10	RX active (RX) - Is set to ‘1’ if a reception to the DMA channel is currently active otherwise it is ‘0’.	r
9	Abort TX (AT) - Set to one to abort the currently transmitting packet and disable transmissions. If no transmission is active the only effect is to disable transmissions. Self clearing.	rw
8	RX AHB error (RA) - An error response was detected on the AHB bus while this receive DMA channel was accessing the bus.	wc
7	TX AHB error (TA) - An error response was detected on the AHB bus while this transmit DMA channel was accessing the bus.	wc
6	Packet received (PR) - This bit is set each time a packet has been received. never cleared by the SW-node.	wc
5	Packet sent (PS) - This bit is set each time a packet has been sent. Never cleared by the SW-node.	wc
4	AHB error interrupt (AI) - If set, an interrupt will be generated each time an AHB error occurs when this DMA channel is accessing the bus.	rw
3	Receive interrupt (RI) - If set, an interrupt will be generated each time a packet has been received. This happens both if the packet is terminated by an EEP or EOP.	rw
2	Transmit interrupt (TI) - If set, an interrupt will be generated each time a packet is transmitted. The interrupt is generated regardless of whether the transmission was successful or not.	rw
1	Receiver enable (RE) - Set to one when packets are allowed to be received to this channel.	rw
0	Transmitter enable (TE) - Write a one to this bit each time new descriptors are activated in the table. Writing a one will cause the SW-node to read a new descriptor and try to transmit the packet it points to. This bit is automatically cleared when the SW-node encounters a descriptor which is disabled.	rw

Table 27. AMBA port RX maximum length register.

31	25	24	0
RESERVED		RXMAXLEN	
NA		NR	

31: 25	RESERVED	r
24: 0	RX maximum length (RXMAXLEN) - Receiver packet maximum length in bytes. Only bits 24 - 2 are writable. Bits 1 - 0 are always 0.	rw

Table 28. AMBA port transmitter descriptor table address register.

31 10	9 4	3 0
DESCBASEADDR	DESCSEL	RESERVED
NR	0	NA

31: 10	Descriptor table base address (DESCBASEADDR) - Sets the base address of the descriptor table.	rw
9: 4	Descriptor selector (DESCSEL) - Offset into the descriptor table. Shows which descriptor is currently used by the GRSPW. For each new descriptor read, the selector will increase with 16 and eventually wrap to zero again.	rw
3: 0	RESERVED	r

Table 29. AMBA port receiver descriptor table address register.

31 10	9 3	2 0
DESCBASEADDR	DESCSEL	RESERVED
NR	0	NA

31: 10	Descriptor table base address (DESCBASEADDR) - Sets the base address of the descriptor table. Not reset.	rw
9: 3	Descriptor selector (DESCSEL) - Offset into the descriptor table. Shows which descriptor is currently used by the GRSPW. For each new descriptor read, the selector will increase with 8 and eventually wrap to zero again. Reset value: 0.	rw
2: 0	RESERVED	r

Table 30. AMBA port dma channel address register

31 16	15 8	7 0
RESERVED	MASK	ADDR
NA	NR	NR

31: 8	RESERVED	r
15: 8	Mask (MASK) - Mask used for node identification on the SpaceWire network. This field is used for masking the address before comparison. Both the received address and the ADDR field are anded with the inverse of MASK before the address check.	rw
7: 0	Address (ADDR) - Address used for node identification on the SpaceWire network for the corresponding dma channel when the EN bit in the DMA control register is set.	rw

3.6 Configuration port

The configuration port uses the RMAP protocol (ECSS-E-ST-50-52C). Verified writes, reads and read-modify-writes all of length 4 bytes are supported (8B for RMW if the mask field is included in the count). Replies sent from the configuration port are always replied to the port they arrived from regardless of the source address. The address space of the configuration port is specified in section 3.7. Addresses outside of the range will result in an authorization error. Table 32 gives a detailed listing of the configuration port's handling of RMAP packets.

Per default the configuration area can be accessed from all the ports. Configuration accesses can be individually disabled per port using the CE bit in the port control register. Writes to the configuration area can be globally disabled by writing a 0 to the WE bit in the configuration write enable register. This disables write accesses from all ports to all registers except the configuration write enable register itself.

There is also a signal which can be used to disable configuration accesses from all ports except port 1. This signal overrides the settings of the CE bits in the port control registers. The global write enable register still affects port 1 in this case.

When an otherwise correct RMAP command destined to the configuration port is received but not allowed due to one or more of the configuration access disable options being enabled a reply with status set to authorization failure will be sent if requested. If a reply is not requested the packet will be silently discarded. In both cases the command will not be performed and has no effect on the configuration port registers.

3.6.1 AMBA AHB slave interface

An AMBA AHB slave interface can be optionally included which makes the whole configuration port memory area accessible from the AHB bus. The address offsets are the same as when accessing through RMAP but the base address is different.

Only word accesses (32-bit) are allowed. The routing table is shared between the ports, RMAP target and AHB slave so accesses from the AHB slave might be stalled because the of accesses from the other sources. The priority order starting from the highest is router ports, RMAP target and AHB slave. The router ports access order is controlled using a round-robin arbitration mechanism.

None of the registers and signals for limiting configuration accesses have any effect on the AHB slave interface.

There is predetermined order in which error-codes are set in the case of multiple errors in the core. It is shown in table 31.

Errors up to and including Invalid Data CRC (number 8) are checked before executing a command. The other errors do not prevent verified operations from being performed.

The details of the support for the different commands are now presented. All defined commands which are received but have an option set which is not supported in this specific implementation will not be executed and a reply is sent (if the acknowledge bit was set) with error code 10.

Table 31. The order of error detection in case of multiple errors in the configuration port RMAP target. The error detected first has number 1.

Detection Order	Error Code	Error
1	12	Invalid destination logical address
2	2	Unused RMAP packet type or command code
3	3	Invalid destination key
4	9	Verify buffer overrun
5	11	RMW data length error
6	10	Authorization failure
8	5/7	Early EOP / EEP (if early)
9	4	Invalid Data CRC
11	7	EEP
12	6	Too much data

3.6.2 Write commands

The write commands are divided into two subcategories when examining their capabilities: verified writes and non-verified writes. The configuration port only supports verified writes with the length restricted to 4 and 0 bytes and the address must be 4 B aligned (address(1:0)=00). Since only one location can be accessed for each command the incrementing address bit can be set to either 0 or 1 and the behavior will be the same. If any of the restrictions mentioned above are violated an reply (if requested) will be sent with the status field set to 10.

3.6.3 Read commands

Read commands are also restricted to 4 or 0 bytes in length and the address has to be 4 B aligned. As for writes each read command only accesses one location so the increment bit can be either 0 or 1.

3.6.4 RMW commands

RMW supports the size 4 or 0 bytes (8 B or 0 B if the mask field is included in the count). The RMW accesses have the same restrictions as the verified writes. As in the verified write case, the incrementing bit can be set to any value since only one operation will be performed for each command. Too much data is detected after the complete command was received and will not prevent the write from being executed. No data is sent in a reply if an error is detected i.e. the status field is non-zero.

The RMW operation is performed by first reading the address location and then writing the same location with the value obtained from the formula data = (writedata and mask) or (readdata and not mask). This means the data bits corresponding to mask bits set to 0 will retain their old value and other bits will be updated with a new value from the data provided in the rmw command.

Table 32. RMAP command support by the configuration port.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Packet type	Action
Reserved	**Command / Response**	**Write / Read**	**Verify data before write**	**Acknow-ledge**	**Increment Address**		
0	0	-	-	-	-	Response	Packet is discarded, no operation is performed and no reply is sent.
0	1	0	0	0	0	Not used	Packet is discarded, no operation is performed and no reply is sent.
0	1	0	0	0	1	Not used	Packet is discarded, no operation is performed and no reply is sent.
0	1	0	0	1	0	Read single address	Supported. Address has to be word aligned and belonging to the defined address range, data length has to be 4. Reply is sent. If alignment restrictions or address ranges are violated error code is set to 10.
0	1	0	0	1	1	Read incrementing address.	Supported. Address has to be word aligned and belonging to the defined address range, data length has to be 4. Reply is sent. If alignment restrictions or address ranges are violated error code is set to 10.
0	1	0	1	0	0	Not used	Packet is discarded, no operation is performed and no reply is sent.
0	1	0	1	0	1	Not used	Packet is discarded, no operation is performed and no reply is sent.
0	1	0	1	1	0	Not used	No operation is performed. Reply is sent with error code 2.
0	1	0	1	1	1	Read-Modify-Write incrementing address	Supported. The data length has to be 8 (4 B mask and 4 B data) and the address word aligned and within the supported range. If these restrictions are violated the command is not executed and the error code is set to 10. Reply is sent.
0	1	1	0	0	0	Write, single-address, do not verify before writing, no acknowledge	Not implemented. Packet is discarded and no reply is sent.

Table 32. RMAP command support by the configuration port.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Packet type	Action
Reserved	Command / Response	Write / Read	Verify data before write	Acknow-ledge	Increment Address		
0	1	1	0	0	1	Write, incre-menting address, do not verify before writ-ing, no acknowledge	Not implemented. Packet is dis-carded and no reply is sent.
0	1	1	0	1	0	Write, sin-gle-address, do not verify before writ-ing, send acknowledge	Not implemented. Command is not executed. Error code is set to 10 and a Reply is sent.
0	1	1	0	1	1	Write, incre-menting address, do not verify before writ-ing, send acknowledge	Not implemented. Command is not executed. Error code is set to 10 and a Reply is sent.
0	1	1	1	0	0	Write, single address, ver-ify before writing, no acknowledge	Supported. Length must be 4, address must be word aligned and within the allowed range. Otherwise the command is not executed. No reply is sent.

Table 32. RMAP command support by the configuration port.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Packet type	Action
Reserved	Command / Response	Write / Read	Verify data before write	Acknow-ledge	Increment Address		
0	1	1	1	0	1	Write, incre-menting address, ver-ify before writing, no acknowledge	Supported. Length must be 4, address must be word aligned and within the allowed range. Otherwise the command is not executed. No reply is sent.
0	1	1	1	1	0	Write, single address, ver-ify before writing, send acknowledge	Supported. Length must be 4, address must be word aligned and within the allowed range. Otherwise the command is not executed. If one of these errors are detected the error code is set to 10. Reply is sent.
0	1	1	1	1	1	Write, incre-menting address, ver-ify before writing, send acknowledge	Supported. Length must be 4, address must be word aligned and within the allowed range. Otherwise the command is not executed. If one of these errors are detected the error code is set to 10. Reply is sent.
1	0	-	-	-	-	Unused	Packet is discarded, no opera-tion is performed and no reply is sent.
1	1	-	-	-	-	Unused	Packet is discarded, no opera-tion is performed and no reply is sent.

3.7 Registers

The registers listed here are accessed through the RMAP target and the addresses specified shall be set in the address field of the RMAP command. They can also be accessed through AHB if the AHB slave is available.

3.7.1 Reset value definitions

Table 33. Reset value definitions

Value	Description
0	Reset to value 0
1	Reset to value 1
0x0	Hexadecimal value which can be used for multibit fields
NA	Not applicable. For example reserved fields or read only fields which are constant
NR	Not reset
*	Special reset condition. Described in textual description of the bit. Used for example when reset value is taken from a signal

3.7.2 Register type definitions

Table 34. Register type definitions

Value	Description
r	Read only
w	Write only
rw	Readable and writable
wc	Readable and cleared when written with a 1
rc	Readable and cleared when read

Table 35. GRSPWROUTER registers

RMAP address	Register
0x0	RESERVED
0x4-0x7C	Port setup for ports 1-31
0x80-0x3FC	Port setup for logical addresses 32-255
0x400-0x47C	RESERVED
0x480-0x7FC	Routing table entry for logical addresses 32-255
0x800-0x87C	Port 0-31 control
0x880-0x8FC	Port 0-31 status
0x900-0x97C	Timer reload ports 0-31
0xA00	Router configuration/status
0xA04	Time-code
0xA08	Version/instance ID
0xA0C	Initialization divisor
0xA10	Configuration write enable
0xA14	Timer prescaler reload
0xC04-0xC7C	Port 1-31 outgoing character count
0xC84-CFC	Port 1-31 incoming character count
0xD04-0xD7C	Port 1-31 outgoing packet count
0xD84-0xDFC	Port 1-31 incoming packet count

Table 36. Port setup register

31	1	0
PORT ENABLE BITS		PD
NR		NR

31: 1 Port enable bits (PORT ENABLE BITS) - Each individual bit enables, when set to 1, packets with the path or logical address corresponding to this port setup register to be sent on the port with the same number as the bit index. Only bits up to and including the highest port number are valid. rw

0 Packet distribution (PD) - When set to 1 packet distribution is used for the path or logical address corresponding to this port setup register. When set to 0 group adaptive routing is used. rw

NOTE: When this register has been written with a non-zero value it will be considered valid and used for determining the destination port. A memory error at location of the port setup information for the destination address will cause the packet to be discarded which will make the associated physical address unusable until the error is fixed (done automatically if auto-scrubbing is enabled). If this behavior is unacceptable GAR or PD should not be enabled for physical addresses. If a port setup entry for a physical address is valid it can be invalidated by writing zeros (to the valid bits in the register) to the memory location.

Table 37. Routing table entry

31	3	2	1	0
RESERVED		EN	PR	HD
NA		NR	NR	NR

31: 3	RESERVED	
2	Enable (EN) - Enables routing table entry. If enabled the corresponding logical address can be used to route packets, otherwise an invalid address error will be generated and the packet is discarded. Note that the corresponding port setup register must not be set to 0 when a routing table entry is enabled.	rw
1	Priority (PR) - Sets the arbitration priority of this port. 0=low priority, 1=high priority.	rw
0	Header deletion (HD) - Enable header deletion for this logical address.	rw

Table 38. Port control for configuration port (port 0)

31	10	9	8	7	6	5	4	3	2	1	0
RESERVED		TR	RESERVED								
N/A		*	N/A								

31: 10	RESERVED	r
9	Timer enable (TR) - Enable timer for packet transfer timeouts for this port. Reset value set from signal.	rw
8: 0	RESERVED	r

Table 39. Port control

31 24	23 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD	RESERVED	ET	NP	PS	BE	DI	TR	PR	TF	RS	TE	RE	CE	AS	LS	LD
*	N/A	*	0	0	*	0	*	0	0	0	1	NA	1	1	0	0

31: 24	Run-state clock divisor (RD) - Clock divisor value used for this link when in the run-state. Only available for SpW ports, reads as 0 otherwise. Reset value taken from signal.	rw
23: 15	RESERVED	r
14	Enable external time (ET) - When 0 an increment (+1)of the internal time-counter is used when sending a time-code as a result from a tickin. When 1 the time-code value is taken from external pins at the clock edge the tickin is asserted. Reset value taken from the ri.en_ext_time signal with the index corresponding to the current FIFO port starting with 0 from the lowest index FIFO port. Only available for FIFO ports.	rw
13	No port force (NP)- When set to 1 the active port between the primary and redundant port is selected automatically by detecting activity on the incoming signals. When 0 the active port is selected using the PS bit. Only applicable to SpaceWire ports. Only available when dual port is enabled.	rw
12	Port select (PS) - Selects between the primary and redundant port when NP is 0. It has no effect when NP is 1. Only applicable to SpaceWire ports.	rw
11	FIFO bridge enable (BE) - Set to 1 to enable bridge mode for the FIFO port which will enable it to be connected to another GRSPWROUTER FIFO port with automatic transfers. Only available for FIFO ports. Reset value set from signal.	rw
10	Disable port (DI) - Disable data transfers on this port. When asserted packets sent to this port will be spilt and the invalid address bit is set in the source port. For group adaptive routing disabled ports will not be included in the group of possible destinations. For packet distribution they will not be transmitted to but the other destination ports will still be transmitted to. Port 1 cannot be disabled and this bit is read only 0 in that case.	rw
9	Timer enable (TR) - Enable timer for packet transfer timeouts for this port. Reset value set from signal.	rw
8	Priority (PR) - Sets the arbitration priority for the path address corresponding to this port. 0=low priority, 1=high priority.	rw
7	Transmitter FIFO reset (TF) - Resets the transmitter FIFO on this port. This means that the FIFO is emptied (counters and pointers set to 0) and lastly an EEP is written to the FIFO to ensure that incomplete packets are detected by the receiver. If a packet transmission was active (a source port was writing to the destination port) when the FIFO reset was asserted the remainder of that packet up to and including EOP/EEP will be spilt.	rw
6	Receiver spill (RS) - Spills the receiver FIFO meaning that the packet currently being received is spilt up to and including the EOP/EEP. If no packet reception is currently active on this port nothing happens. The port or ports in the case of packet distribution will have an EEP written to the transmit FIFO to indicate that the packet was ended prematurely. Not available for AMBA ports.	rw
5	Time-code enable (TE) - Enables time-codes to be received and transmitted on this port. When enabled received time-codes are processed and if valid a tick-out will be generated and the time-code is forwarded to the other ports. When disabled received time-codes are ignored. Time-codes are only transmitted on this port if this bit is set to 1 and tick-in is ignored otherwise.	rw
4	RESERVED	r
3	Configuration port enable (CE) - Enable accesses to the configuration port. If disabled packets to the configuration port will be spilt.	rw
2	Autostart (AS) - Enable the Link interface FSM Autostart feature. Only available for SpW ports, reads as 0 otherwise.	rw
1	Link start (LS) - Start the link interface FSM. Only available for SpW ports, reads as 0 otherwise.	rw
0	Link disabled (LD) - Disable the link interface FSM. Only available for SpW ports, reads as 0 otherwise.	rw

Table 40. Port status configuration port (port 0)

31 25	24	23 20	19	18	17 12	11 7	6 0
RESERVED	CE	ERRCODE	RE	TS	RESERVED	TP	RESERVED
NA	0	NR	NA	0	NA	00000	NA

31: 25	RESERVED	r
24	Clear error code (CE) - Write with a 1 to clear the ERRCODE field.	rw
23: 20	Error code (ERRCODE) - Shows the latest nonzero RMAP status code. If zero no error has occurred.	r
19	RESERVED	r
18	Timeout spill (TS) - Packet spilled due to timeout.	wc
17: 12	RESERVED	r
11: 7	Transmitting port (TP) - The number of the port currently accessing the configuration port.	r
6: 0	RESERVED	r

Table 41. Port status

31 30	29 20	19	18	17	16	15	14 12	11 7	6	5	4	3	2	1	0
PT	RESERVED	AP	TS	ME	TF	RE	LS	TP	PB	PR	IA	CE	ER	DE	PE
NA	NA	NR	0	0	0	1	000	00000	NA	NA	0	0	0	0	0

31: 30	Port type (PT) - The type of this port. “00” = SpaceWire port, “01” = AMBA port, “10”= FIFO port	r
29: 20	RESERVED	r
19	Active port (AP) - Show which of the primary and redundant port is active. 0 = primary port, 1 = redundant port. Only applicable to SpaceWire ports..	r
18	Timeout spill (TS) - Packet spilled due to timeout.	wc
17	Memory error (ME) - Uncorrectable parity error detected in FIFO memories on this link.	wc
16	Transmit FIFO full (TF) - Set to 1 when the transmit FIFO on this port is full.	r
15	Receive FIFO empty (RE) - Set to 1 when the receive FIFO on this port is full. Not available for AMBA ports.	r
14: 12	Link state (LS) - Current link state. 000 = Error reset. 001 = Error wait, 010 = Ready, 011 = Started, 100 = Connecting, 101 = Run state. Only available for SpW ports, reads as 0 otherwise.	r
11: 7	Transmitting port (TP) - The number of the port currently transmitting on this port. Only valid if the PB bit is set to 1.	r
6	Port transmit busy (PB) - Set to 1 when a packet is being transmitted on this port.	r
5	Port receive busy (PR) - Set to 1 when a packet is being received on this port.	r
4	Invalid address (IA) - A packet with an invalid address was received on this link.	wc
3	Credit error (CE) - Set when a credit error has occurred on the link. Only available for SpW ports, reads as 0 otherwise.	wc
2	Escape error (ER) - Set when an escape error has occurred on the link. Only available for SpW ports, reads as 0 otherwise.	wc
1	Disconnect error (DE) - Set when a disconnect error has occurred on the link. Only available for SpW ports, reads as 0 otherwise.	wc
0	Parity error (PE) - Set when a parity error has occurred on the link. Only available for SpW ports, reads as 0 otherwise.	wc

Table 42. Timer reload

31 10	9 0
RESERVED	RELOAD
	*

31: 10	RESERVED	r
9: 0	Timer reload (RELOAD) - Port specific timer reload value. This timer runs on the prescaler generated tick and determines the timeout period for the port it is associated with. The timeout period will be between reload and reload+1 prescaler ticks. The minimum value of this register is 1. Writing a 0 will result in 1 being written. Reset value set from signal.	rw

Table 43. Router configuration/status

31 27	26 22	21 17	16 8	7	6	5	4	3	2	1	0
SPWPORTS	AMBAPORTS	FIFOPORTS	RESERVED	RE	AD	LS	SA	TF	ME	TA	PP
NA	NA	NA	NA	0	*	*	*	0	0	NA	NA

31: 27	SpaceWire ports (SPWPORTS) - Set to the number of SpaceWire ports in the router.	r
26: 22	AMBA ports (AMBAPORTS) - Set to the number of AMBA ports in the router.	r
21: 17	FIFO ports (FIFOPORTS) - Set to the number of FIFO ports in the router.	r
16: 8	RESERVED	r
7	Reset (RE) - Resets the complete router when written with a 1. The router will not respond for 6-7 core clock cycles. Thus when writing this register through RMAP the reply bit should NOT bet set since the reply will not be sent.	rw
6	Auto disconnect (AD) - When set to 1 ports will be automatically stopped after a timeout period of inactivity. Reset value taken from signal.	rw
5	Link start on request (LS) - When set to 1 ports will be started automatically when there is a request to transmit a packet on the port. The link will only start if it is not disabled. Reset value taken from signal.	rw
4	Self addressing enable (SA) - If set to 1 ports are allowed to send packets to themselves. If 0 packets with the same source and destination port are spilt and an invalid address error is asserted. Reset value taken from signal.	rw
3	Time-code control flag mode (TF) - When 0 the time-code control flags can have any value for normal operation. When set to 1 the control flags must have the value “00” for normal time-code operation, otherwise the time-codes are discarded.	rw
2	Memory error (ME) - Set to one each time an uncorrectable error has been detected in either the routing table or port setup memory.	wc
1	Timers available (TA) - Set to one if the router has watchdog timer support.	r
0	Plug and Play available (PP) - Set to one if the router has Plug and Play support.	r

Table 44. Time-code

31 … 9	8	7 … 6	5 … 0
RESERVED	RE	EN	CF	TIMECNT
NA	0	1	0	0

31: 10	RESERVED	r
9	Reset time-code (RE) - Resets the control flags and time counters to 0 when written with a 1. Always reads as 0.	rw
8	Enable time-codes (EN) - Enable time-codes to propagate and update the counter and control flags. When disabled time-codes are ignored.	rw
7: 6	Time-control flags (CF) - The current value of the router control flags.	r
5: 0	Time-counter (TIMECNT) - Current value of the router time counter.	r

Table 45. Version/Instance ID

31 … 24	23 … 16	15 … 8	7 … 0
MAJOR VERSION	MINOR VERSION	PATCH	INSTANCE ID
NA	NA	NA	*

31: 24	Major version (MAJOR VERSION) - Holds the major version number of the router.	r
23: 16	Minor version (MINOR VERSION) - Holds the minor version number of the router.	r
15: 8	Patch (PATCH) - Holds the patch number of the router.	r
7: 0	Instance ID (INSTANCE ID) - Holds the instance ID number of the router. Reset value taken from signal.	rw

Table 46. Initialization divisor

31 … 8	7 … 0
RESERVED	ID
N/A	*

31: 8	RESERVED	r
7: 0	Initialization clock divisor (ID) - Clock divisor value used by all the SpaceWire links to generate the 10 Mbit/s rate during initialization. All the links use a common clock for this so only one divisor is needed. Only available if there is at least one SpaceWire port in the router, reads as 0 otherwise. Reset value taken from signal.	rw

Table 47. Configuration write enable

31	1	0
RESERVED		WE
NA		1

31: 1	RESERVED	r
0	Configuration write enable (WE) - When set to 1 write accesses to the configuration area are allowed. When set to 0 writes are not allowed except to this register. Write or RMW commands will be replied with an authorization error if a reply was requested.	rw

Table 48. Timer prescaler

31	16	16-1	0
RESERVED		PRESCALER	
NA		*	

31: 16	RESERVED	r
16 0 -1:	Timer prescaler (PRESCALER) - Global prescaler value used for generating a common tick for the port timers. The prescaler runs on clk (table 539) and a tick is generated every prescaler+1 cycle. Reset value taken from signal.	rw

3.8 Signal definitions and reset values

The signals and their reset values are described in table 49.

Table 49. Signal definitions and reset values

Signal name	Type	Function	Active	Reset value
spw_clk	Input	Transmitter default run-state clock	Rising edge	-
spw_rxdp[ ]	Input, LVDS	Data input, positive 1) 3)	High	-
spw_rxdn[ ]	Input, LVDS	Data input, negative 1) 3)	Low	-
spw_rxsp[ ]	Input, LVDS	Strobe input, positive 1) 3)	High	-
spw_rxsn[ ]	Input, LVDS	Strobe input, negative 1) 3)	Low	-
spw_txdp[ ]	Output, LVDS	Data output, positive 1) 3)	High	Logical 0
spw_txdn[ ]	Output, LVDS	Data output, negative 1) 3)	Low	Logical 1
spw_txsp[ ]	Output, LVDS	Strobe output, positive 1) 3)	High	Logical 0
spw_txsn[ ]	Output, LVDS	Strobe output, negative 1) 3)	Low	Logical 1
spw_rxd[ ]	Input, LVTTL	Data input, positive 2) 3)	High	-
spw_rxs[ ]	Input, LVTTL	Strobe input, positive 2) 3)	High	-
spw_txd[ ]	Output, LVTTL	Data output, positive 2) 3)	High	Logical 0
spw_txs[ ]	Output LVTTL	Strobe output, positive 2) 3)	High	Logical 0
timecodeen	Input	Enable time-code functionality	High	
tickin[ ]	Input	Tick input signals for FIFO interfaces	High	
timein*n*[7:0]	Input	Time input signals for FIFO interface *n*	-	
tickout[ ]	Output	Tick output signals for FIFO interfaces	High	
timeout*n*[7:0]	Output	Time output signals for FIFO interface *n*	-	
rxread[ ]	Input	Receiver FIFO read signals for FIFO interfaces	High	
rxchar*n*[8:0]	Output	Receiver character signals for FIFO interface *n*	-	Logical 0
txwrite[ ]	Input	Transmitter FIFO write signals for FIFO interfaces	High	
txchar*n*[8:0]	Input	Transmitter character signals for FIFO interface *n*	-	
txfull[ ]	Output	Transmitter full signal for FIFO interfaces	High	Logical 0
txafull[ ]	Output	Transmitter almost full signal for FIFO interfaces	High	Logical 0
rxcharav[ ]	Output	Receiver data available signal for FIFO interfaces	High	Logical 0
rxaempty[ ]	Output	Receiver empty signal for FIFO interfaces	High	Logical 0
enbridge[ ]	Input	Enables bridge mode for the FIFO interfaces	High	

Table 49. Signal definitions and reset values

Signal name	Type	Function	Active	Reset value
linkrun[ ]	Output	SpaceWire link in run state when asserted 3)	High	Logical 0
gerror	Output	Global error	High	Logical 0
reload_ps[ ]	Input	Reset value for the timer prescaler. Mapped to bits of the prescaler register. 4)	-	
reload_timer[ ]	Input	Reset value for the port timer registers. Mapped to bits of the timer registers. 4)	-	
timeren	Input	Reset value for the timer enable bit in the port control registers	High	
selfaddren	Input	Reset value for *selfaddren* register bit	High	
linkstartreq	Input	Reset value for the *linkstartreq* register bit	High	
autodconnect	Input	Reset value for the *autodconnect* register bit	High	
instanceid[ ]	Input	Reset value for the instance *id* field of the version/ instance id register. 4)	-	
idivisor[ ]	Input	Reset value for the initialization divisor register and the SpaceWire port run-state divisor registers. 4)	-	
cfglock	Input	Lock configuration port accesses from all ports except port 1 when asserted	High	

Note 1: These signals are only used in configuration with on-chip LVDS drivers.

Note 2: These signals are only used in configuration with off-chip LVDS drivers.

Note 3: All bus signal elements are not used in all listed configurations. See pin assignment at the end of document for details.

Note 4: For the actual mapping between signals and register bits can vary. See signal definition at the beginning of document for details.

3.9 Timing

The timing waveforms are shown in figure 9 and 10. Timing parameters are defined in table 50 and 51.

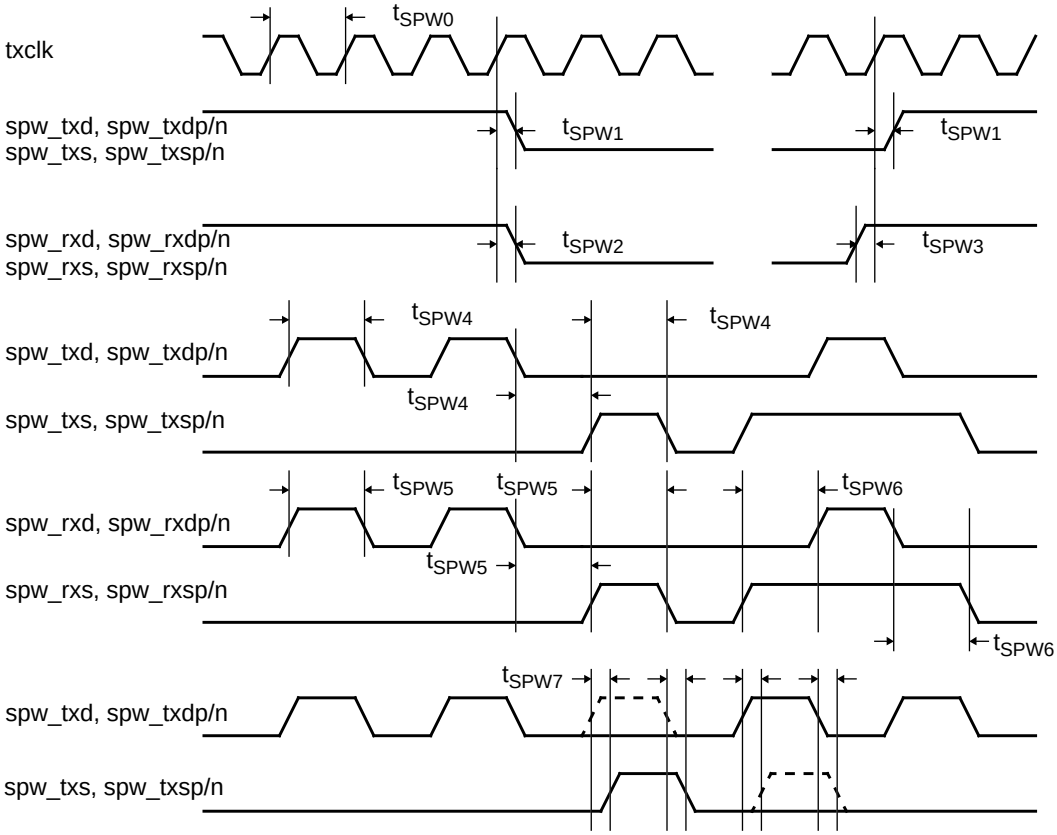


Figure 9. Timing waveforms

Table 50. Timing parameters

Name	Parameter	Reference edge	Min	Max	Unit
t_{SPW0}	transmit clock period	-	10	-	ns
t_{SPW1}	clock to output delay	rising *txclk* edge	0	23	ns
t_{SPW2}	input to clock hold	-	-	-	not applicable
t_{SPW3}	input to clock setup	-	-	-	not applicable
t_{SPW4}	output data bit period	-	-		*clk* periods
		-	5	500	ns
t_{SPW5}	input data bit period	-	5	500	ns
t_{SPW6}	data & strobe edge separation	-	2600	-	ps
t_{SPW7}	data & strobe output skew	-	-	220	ps

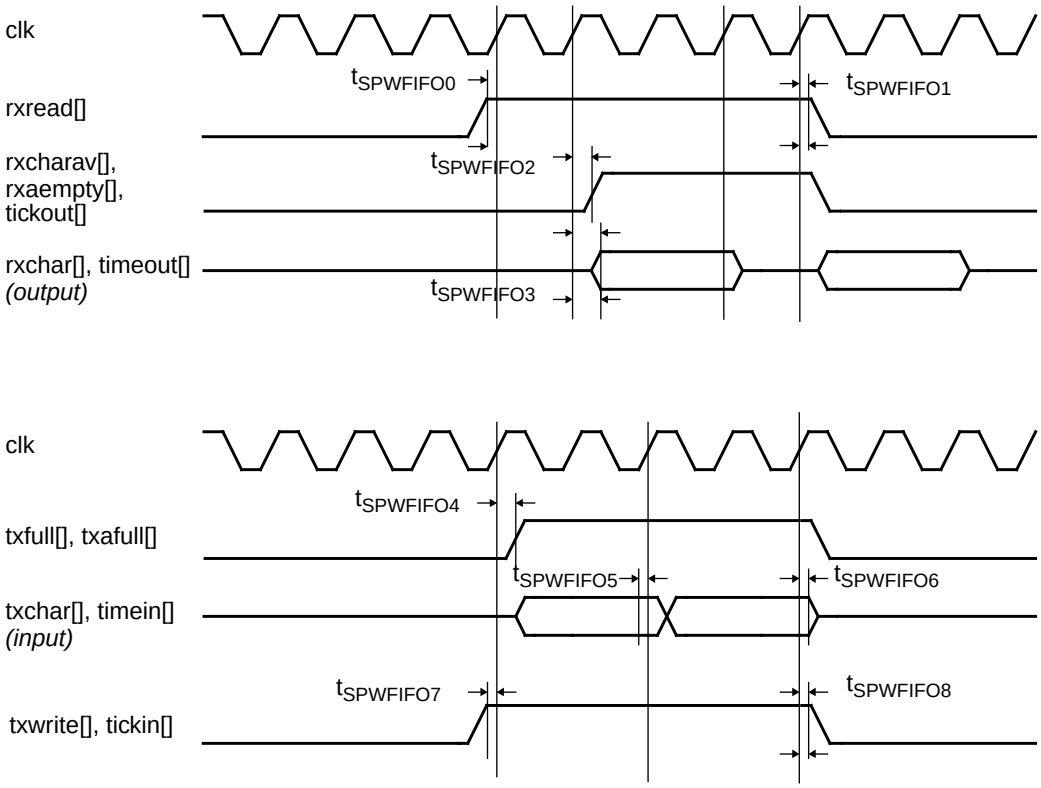


Figure 10. Timing waveforms - FIFO interface

Table 51. Timing parameters - FIFO interface

Name	Parameter	Reference edge	Min	Max	Unit
$t_{SPWFIFO0}$	input to clock setup	rising clk edge	21	-	ns
$t_{SPWFIFO1}$	input from clock hold	rising clk edge	0.5	-	ns
$t_{SPWFIFO2}$	clock to output delay	rising clk edge	4	17	ns
$t_{SPWFIFO3}$	clock to rxchar, timeout output delay	rising clk edge	4	14	ns
$t_{SPWFIFO4}$	clock to output delay	rising clk edge	4	14	ns
$t_{SPWFIFO5}$	txchar, timein input to clock setup	rising clk edge	19	-	ns
$t_{SPWFIFO6}$	txchar, timein input from clock hold	rising clk edge	0.5	-	ns
$t_{SPWFIFO7}$	input to clock setup	rising clk edge	20	-	ns
$t_{SPWFIFO8}$	input from clock hold	rising clk edge	0.5	-	ns

4 SpaceWire encoder-decoder

4.1 Overview

The SpaceWire encoder-decoder implements an encoder-decoder compliant to the SpaceWire standard (ECSS-E-ST-50-12C). It provides a generic host-interface consisting of control signals, status signals, time-code interface and 9-bit wide data buses connecting to a pair of FIFOs.

The core is practically identical to the encoder-decoder used in the GRSPW2 the only difference being the host-interface.

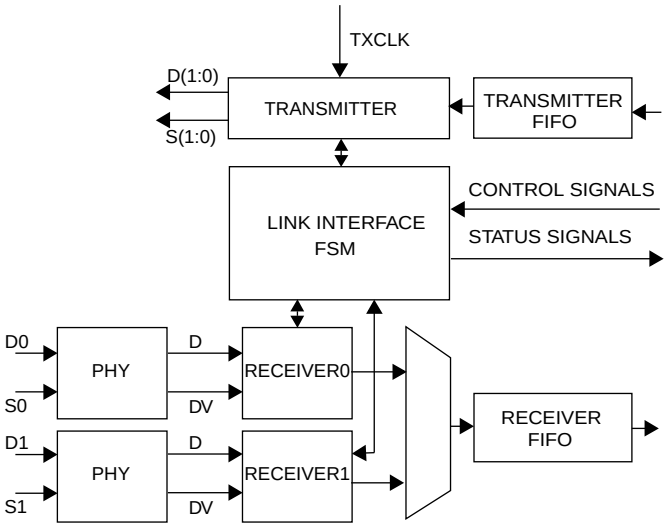


Figure 11. Block diagram

Note: The signals described in this section are internal only and cannot be accessed by the user of the device. This section is provided for information only.

4.2 Operation

4.2.1 Overview

A block diagram of the internal structure of the core can be found in figure 11. It consists of the receiver, transmitter and the link interface FSM. They handle communication on the SpaceWire network. The PHY blocks provides a common interface for the receiver to the four different data recovery schemes and is external to this core.

Time-codes are transmitted through a signal interface as specified in the SpaceWire standard.

4.2.2 Link-interface FSM

The link-interface FSM controls the link interface (a more detailed description is found in the SpaceWire standard). The low-level protocol handling (the signal and character level of the SpaceWire standard) is handled by the transmitter and receiver while the FSM handles the exchange level.

The link-interface FSM is controlled through the control signals. The link can be disabled through the link disabled signal, which depending on the current state, either prevents the link-interface from

reaching the started state or forces it to the error-reset state. When the link is not disabled, the link interface FSM is allowed to enter the started state when either the link start signal is asserted or when a NULL character has been received and the autostart signal is asserted.

The current state of the link-interface determines which type of characters are allowed to be transmitted which together with the requests made from the host interface determine what character will be sent.

Time-codes are sent when the FSM is in the run-state and a request is made through the time-interface (described in section 4.2.5).

When the link-interface is in the connecting- or run-state it is allowed to send FCTs. FCTs are sent automatically by the link-interface when possible. This is done based on the maximum value of 56 for the outstanding credit counter and the currently free space in the receiver FIFO. FCTs are sent as long as the outstanding counter is less than or equal to 48 and there are at least 8 more empty FIFO entries than the counter value.

N-Chars are sent in the run-state when they are available from the transmitter FIFO and there are credits available. NULLs are sent when no other character transmission is requested or the FSM is in a state where no other transmissions are allowed.

The credit counter (incoming credits) is automatically increased when FCTs are received and decreased when N-Chars are transmitted. Received N-Chars are stored to the receiver N-Char FIFO while received Time-codes are handled by the time-interface.

4.2.3 Transmitter

The state of the FSM, credit counters, requests from the time-interface and requests from the transmitter FIFO are used to decide the next character to be transmitted. The type of character and the character itself (for N-Chars and Time-codes) to be transmitted are presented to the low-level transmitter which is located in a separate clock-domain.

The transmitter logic in the host clock domain decides what character to send next and sets the proper control signal and presents any needed character to the low-level transmitter as shown in figure 12. The transmitter sends the requested characters and generates parity and control bits as needed. If no requests are made from the host domain, NULLs are sent as long as the transmitter is enabled. Most of the signal and character levels of the SpaceWire standard is handled in the transmitter. External LVDS drivers are needed for the data and strobe signals.

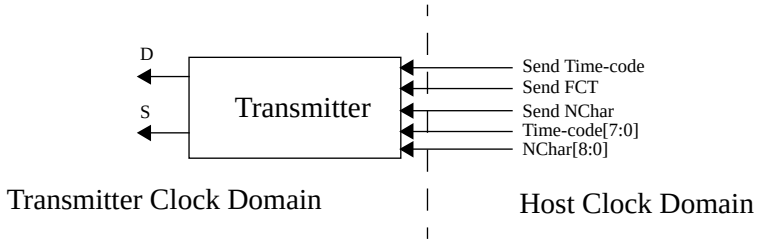


Figure 12. Schematic of the link interface transmitter.

4.2.4 Receiver

The receiver is activated as soon as the link-interface leaves the error reset state. Then after a NULL is received it can start receiving any characters. It detects parity, escape and credit errors which causes

the link interface to enter the error-reset state. Disconnections are handled in the link-interface part in the tx clock domain because no receiver clock is available when disconnected.

Received characters are flagged to the host domain and the data is presented in parallel form. L-Chars are the handled automatically by the host domain link-interface part while all N-Chars are stored in the receiver FIFO for further handling. If two or more consecutive EOPs/EEPs are received all but the first are discarded.

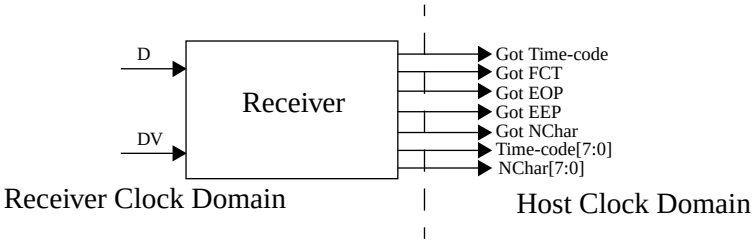


Figure 13. Schematic of the link interface receiver.

4.2.5 Time interface

The time interface is used for sending Time-codes over the SpaceWire network and consists of a timein signal, timectrlin signal, tickin signal, timeout signal, timectrlout signal and a tickout signal.

Each Time-code sent from the core is a concatenation of the timectrlin and the timein signal. It is transmitted each time tickin is kept asserted until the tickin_done output is asserted. Time-codes are only transmitted when the link-interface FSM is in run-state and when the previous character transmission is finished. This can cause a delay between when the assertion of tickin_done after tickin has been asserted. If tickin is not kept asserted until tickin_done is asserted the time-code will not be transmitted. Figure 14 shows an example of how a time-code is transmitted.

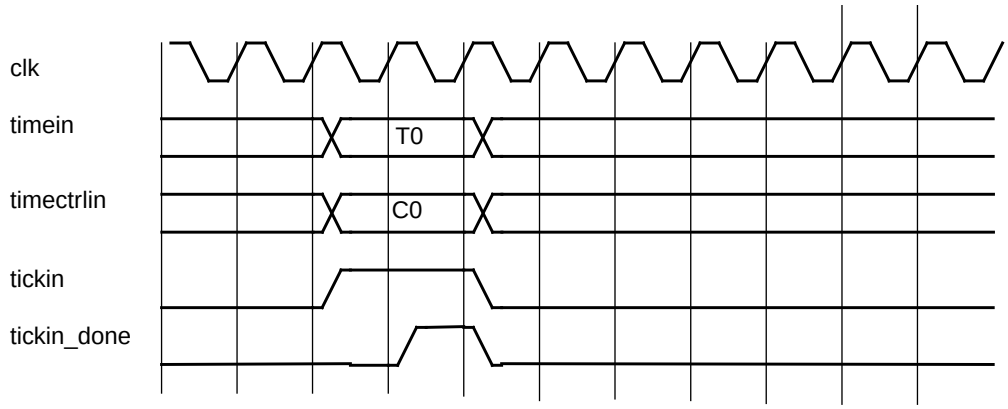


Figure 14. Transmitting a time-code using tickin, timein and timectrlin.

Received Time-codes are presented on the timectrlout and timectrl signals. They are valid when the tickout output is asserted. A tickout is generated each time a valid time-code is received. When the tickout is generated the tick-out signal will be asserted one clock-cycle. Figure 15 shows how time-codes are received.

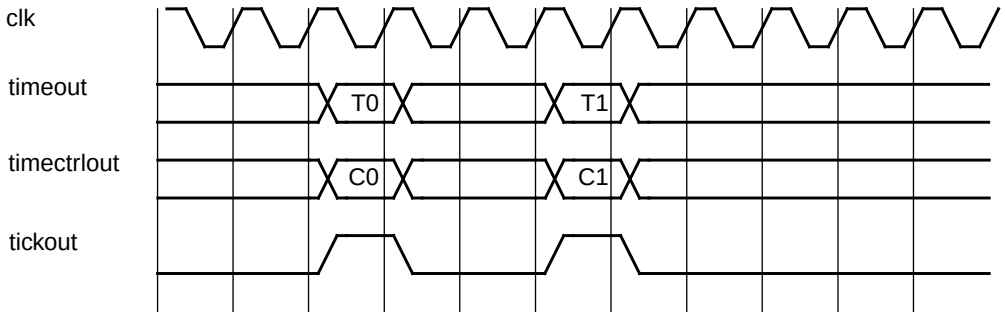


Figure 15. Receiving time-codes using tickout, timeout and timectrlout.

4.3 Receiver interface

The receiver interface consists of the following signals connected to the receiver FIFO: rxicharav, rxicharcnt, rxichar, rxiread. Rxicharav is asserted when there are one or more characters available in the receiver FIFO while rxicharcnt shows the actual number available. Rxiread should be asserted for one cycle when rxicharav is asserted to read out a character. The character will be available on rxichar the cycle following the assertion of rxiread. Figure 16 shows an example of reading characters from the receiver FIFO.

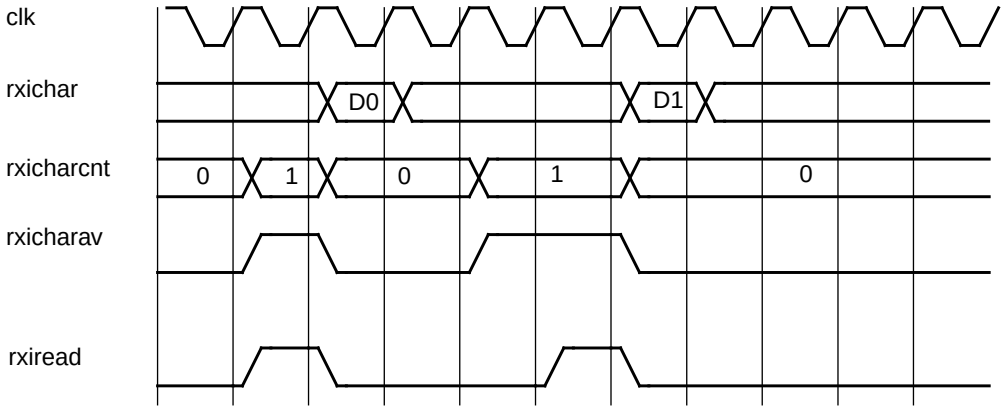


Figure 16. Receiving characters through the FIFO interface.

4.3.1 Link errors

When an link error occurs during reception an EEP is automatically inserted into the Receiver FIFO if the previous character written to the FIFO was not an EOP or EEP.

4.4 Transmitter interface

The transmitter interface consists of the following signals: txiwrite, txichar, txiflush, txicharcnt, txifull. Txifull is asserted when the transmitter FIFO is full while txicharcnt shows the actual number of characters currently in the FIFO. Txiwrite should be asserted for one cycle to write the value on txichar into the FIFO. Txiflush should be asserted for one cycle to discard all characters in the FIFO. No

new characters should be written to the FIFO the same or the following cycle that txiflush is asserted. Figure 17 shows an example of writing characters to the transmitter FIFO.

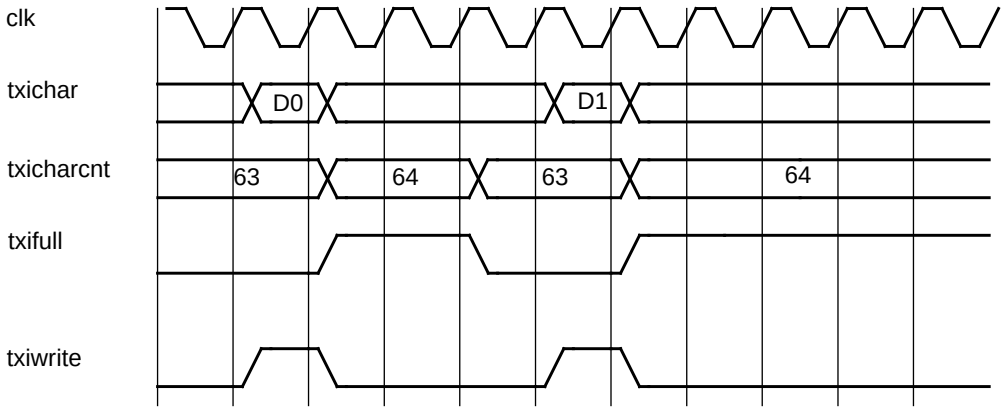


Figure 17. Transmitting characters through the FIFO interface.

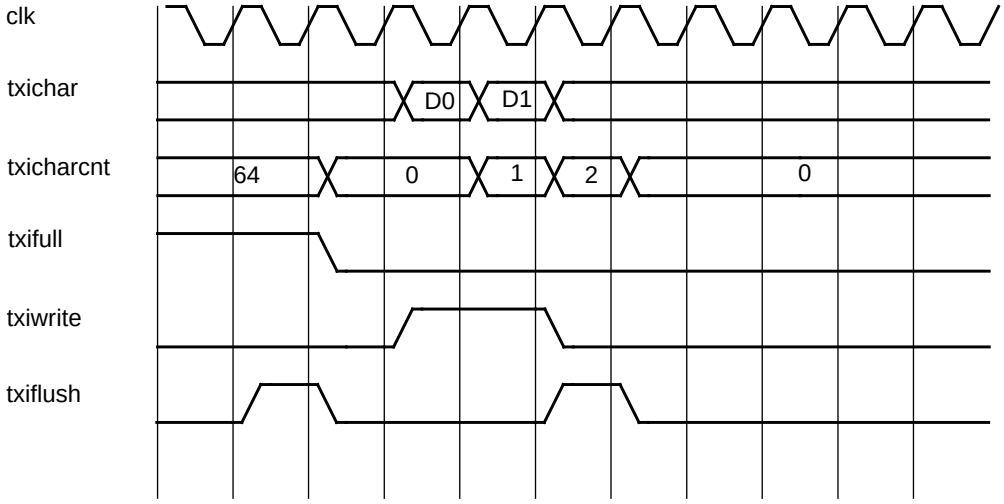


Figure 18. Use of txiflush.

4.4.1 Link errors

When a link error occurs characters read from the transmitter FIFO by the transmission logic will be discarded up to and including the next EOP or EEP character.

4.5 Registers

There are no user accessible registers in the core.

5 PCI Initiator/Target

5.1 Overview

This core provides a complete interface to an external PCI bus, with both initiator and target functions. The PCI initiator has PCI system host capability. The interface is based on the Actel CorePCIF IP and provides an AMBA bus backend.

The interface consists of five main blocks: the PCI master/target, the AMBA AHB master, the AMBA AHB slave, AMBA APB slave, and the interrupt controller. The interrupt controller forwards all interrupts generated on the AMBA bus to the PCI bus.

The core acts as a bridge between the PCI bus and the AMBA AHB bus. An access to the PCI target is transferred to the AMBA AHB bus using the AMBA master interface An access to the AMBA AHB slave interface is transferred to the PCI bus using the PCI master interface. The PCI bus and the AMBA interface belong to two different clock domains. Synchronization is performed inside the core through FIFOs. The AMBA interface is extended with the GRLIB plug&play information.

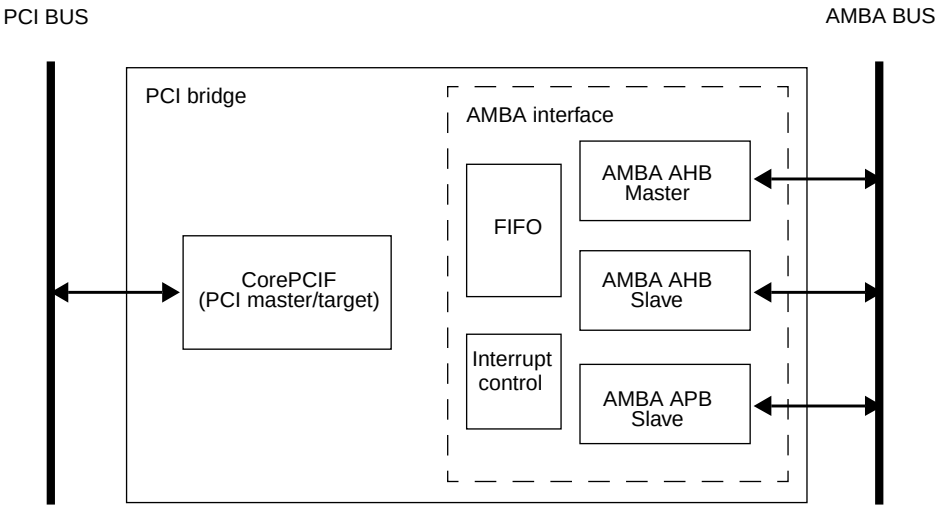


Figure 19. Block diagram

5.2 Operation

5.2.1 PCI Initiator

The PCI initiator can be enabled and disabled in the PCI configuration space. When the PCI master is disabled, all accesses to the AMBA backend except for accesses to its own PCI configuration space will generate a error response on the AMBA bus. The core occupies two areas in the AMBA address space. One AHB memory bank used for generation of PCI memory cycles and one AHB I/O bank for PCI I/O and PCI configuration cycles. The AHB I/O bank has a fixed size of 128 kBytes.

PCI memory cycles: The PCI master generates “Memory read multiple” accesses on the PCI bus for burst read accesses to the memory bank of the AMBA backend. For single read accesses to the AMBA backend, “Memory read” accesses are generated on the PCI bus. All write accesses to the AMBA backend generate “Memory write” accesses on the PCI bus. The most significant bits of the

PCI address are set by mapping registers, while the least significant bits are directly transferred from the AMBA backend. A separate mapping register is implemented for each AMBA master.

To generate PCI memory accesses, the following steps must be executed. The master function must be enabled, preferably by the PCI system host during the PCI configuration and the mapping register must be programmed with the most significant bits of the PCI address that should be accessed. After this, read and write accesses to the AMBA backend will be transferred the corresponding PCI address.

PCI I/O cycles: Accesses to the lower half of the AHB I/O bank is translated to PCI I/O cycles. The upper 16 bits of the PCI address are defined by a mapping register (AMBA to PCI mapping for PCI I/O cycles) while the lower 16 bits are transferred form the AMBA bus.

To generate PCI I/O accesses, The master function must be enabled, preferably by the PCI system host during the PCI configuration and the mapping register must be programmed with the most significant bits of the PCI address that should be accessed. After this, read and write accesses to the lower half of the AHB I/O bank will be translated to “I/O read” and “I/O write” on the PCI bus.

PCI configuration cycles: Accesses to the upper half of the AHB I/O bank translates to PCI configuration cycles. The upper 21 bits of the PCI address is calculated from bit 15:11 of the AMBA address (PCI_address[AMBA_address[15:11] + 10] = ‘1’). When AMBA_address[15:11] is equal to zero, the core makes accesses to its own PCI configuration space (this will not generate any accesses on the PCI bus). The lower 11 bits of the PCI address is transferred from AMBA address except for bit 1 and 0 which are always set to zero. Accesses to the core’s own PCI configuration space are allowed before the master function is enabled, to be able to enable the master. Note: When in peripheral slot, configuration cycles must not be initiated.

5.2.2 PCI Target

The PCI target function can be enabled and disabled in the PCI configuration space. When enabled, the PCI target accepts “Memory read”, “Memory read multiple”, and “Memory write” commands for data accesses. Access to the PCI configuration space is provided for “Configuration read” and “Configuration write” commands. When the PCI target is accessed (except for Configuration read/write), the core transfers this access to the AMBA backend. The most significant bits of the address used by the AMBA backend is controlled by a mapping register, while the least significant bits of the address are directly transferred from the PCI access. A separate mapping register is implemented for PCI BAR 1 - 4.

The following configuration steps are required for the PCI target to correctly respond to data accesses. The target must be setup to accept memory accesses, preferably by the PCI system host during PCI configuration. The mapping register must be programmed with the most significant bits of the AMBA address. After this configuration, accesses to the PCI target interface are transferred to the AMBA bus.

The PCI target interface provides three PCI memory bars: BAR 0 enables access to the configuration registers; BAR 1 is used for data transfers; BAR 5 is used by the master function and should never be accessed by any other master on the PCI bus.

5.2.3 Configuration

The core has configuration registers accessible via the AMBA APB interface and via the PCI BAR 0. The PCI BAR to AMBA address mapping registers and the interrupt controller registers are accessible via the PCI BAR 0. The interrupt registers must be setup to enable interrupt handling. The PCI to AMBA address mapping registers must be setup to translate the PCI access into the correct AMBA access. These mapping registers are also accessible via the AMBA APB interface. The AMBA to PCI

address mapping registers are accessible via the AMBA APB interface. These registers must be setup to translate the access to the AMBA AHB slave interface into the correct PCI address.

5.2.4 Byte access

Single byte accesses are supported by the interface. The byte is directly transferred between the two buses (i.e. a write of the byte located at bit 7:0 on the AMBA bus is transferred to a write of the byte located at bit 7:0 on the PCI bus). The core does not support the PCI byte-enables to be changed during burst accesses. The PCI byte-enable is only sampled for the first word in a burst.

5.2.5 Error response

The PCI target do not generate error responses. A PCI access that transfer into an AMBA access with an invalid address is not generating an error response on the PCI bus.

The AMBA backend generate a two cycle error response in the following cases. When the AMBA backend is accessed and the PCI master function is disabled, or when Master/Target abort is detected by the PCI master (except in case a PCI configuration access is performed). In the case of a Master/Target abort, the core automatically resets the error bits in the PCI configuration space and saves the error status in its status register.

5.2.6 Interrupt controller (generation of PCI interrupt)

The interrupt controller monitors interrupt 1 - 15 provided by the AMBA system. Each interrupt can be assigned to one of two levels (0 or 1) as programmed in the interrupt level register. Level 1 has higher priority than level 0. The interrupt are prioritized within each level, with interrupt 15 having the highest priority and interrupt 1 the lowest. The highest interrupt at level 1 will be forwarded to the PCI bus. If no unmasked pending interrupt exist at level 1, then the highest interrupt at level 0 will be forwarded. To determine which interrupt has occurred the interrupt status/ack register can be read. To acknowledge the interrupt the interrupt status/ack register is written.

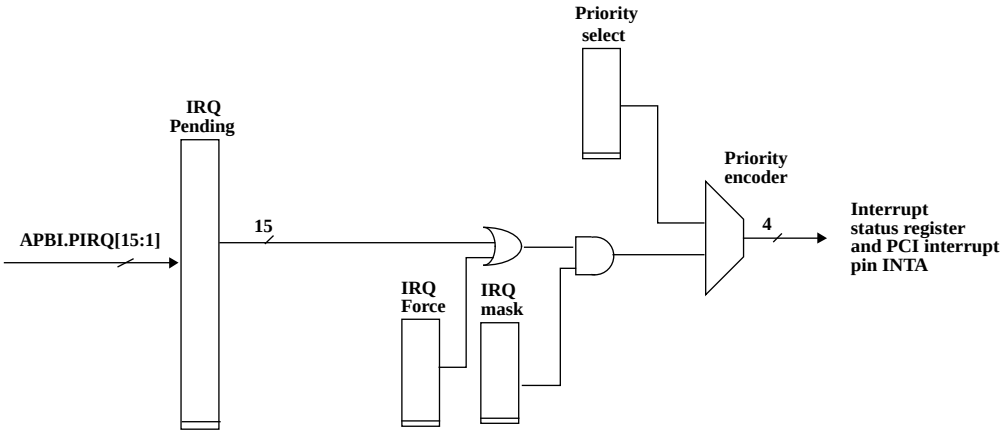


Figure 20. Block diagram

When an interrupt is acknowledged, the corresponding pending bit will automatically be cleared. Interrupts may also be forced by setting a bit in the interrupt force register. In this case, the acknowledgement will clear the force bit rather than the pending bit. After reset, the interrupt mask register is set to all zeros while the remaining control registers are undefined.

5.3 Registers

The core is programmed via registers mapped into the APB address space and into PCI BAR 0.

Table 52. PCIF: APB registers

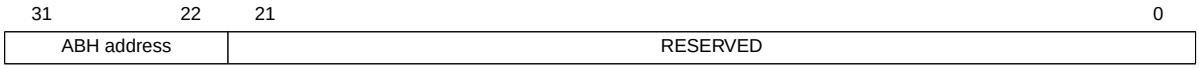
APB address offset	Register
0x00	PCI to AMBA mapping for PCI BAR 1
0x04	PCI to AMBA mapping for PCI BAR 2
0x08	PCI to AMBA mapping for PCI BAR 3
0x0C	PCI to AMBA mapping for PCI BAR 4
0x10	Reserved
0x14	AMBA to PCI address mapping for PCI I/O cycles
0x18	Status register
0x1C	Input interrupt mask (not used)
0x40 - 0x7C	AMBA master to PCI address mapping registers

The AMBA master to PCI mapping registers are all word aligned. Only the registers corresponding to a master included in the system are implemented (i.e. if a system includes 8 masters, with master ID 0 to 7, the 8 first mapping registers are implemented).

Table 53. PCIF: PCI BAR 0 registers

PCI address offset	Register
0x00	PCI to AMBA mapping for PCI BAR 1
0x04	PCI to AMBA mapping for PCI BAR 2
0x08	PCI to AMBA mapping for PCI BAR 3
0x0C	PCI to AMBA mapping for PCI BAR 4

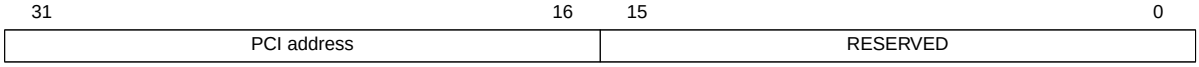
Table 54. PCIF: PCI to AMBA mapping for PCI BAR *n* registers



31 : 22 MBS of the AMBA address

21 : 0 RESERVED

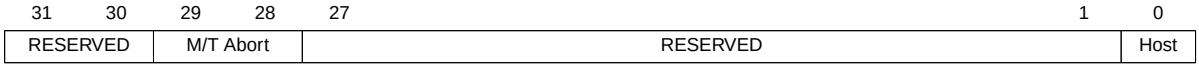
Table 55. PCIF: AMBA to PCI address mapping for PCI I/O cycles



31 : 16 MBS of the PCI address

15 : 0 RESERVED

Table 56. PCIF: Status register

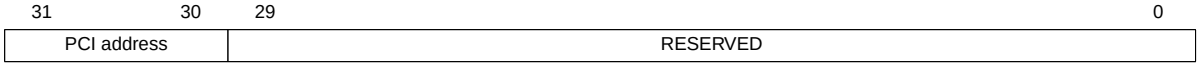


31 : 30	RESERVED
29 : 28	Master and Target abort status from PCI configuration space
27 : 1	RESERVED
0	System host (’1’ = in peripheral slot)

Table 57. PCIF: Input interrupt mask

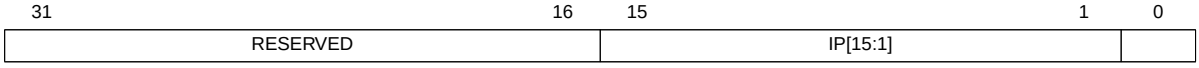
31 : 0	RESERVED

Table 58. PCIF: AMBA master to PCI address mapping register



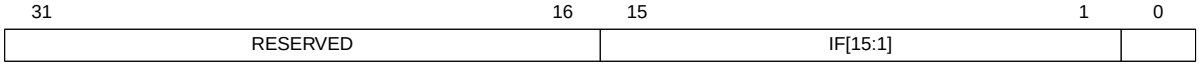
31 : 30	MBS of the PCI address
29 : 0	RESERVED

Table 59. PCIF: Interrupt pending register



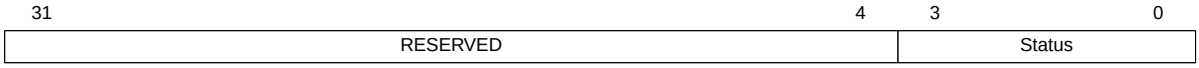
31 : 16	RESERVED
15 : 1	Interrupt pending n (IP[n]): Interrupt pending for interrupt n
0	RESERVED

Table 60. PCIF: Interrupt force register



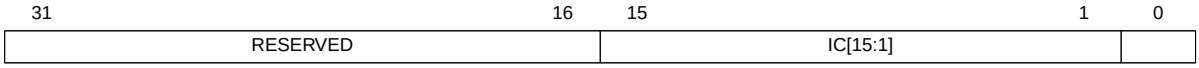
31 : 16	RESERVED
15 : 1	Interrupt force n (IF[n]): Force interrupt nr n
0	RESERVED

Table 61. PCIF: Interrupt status/ack register



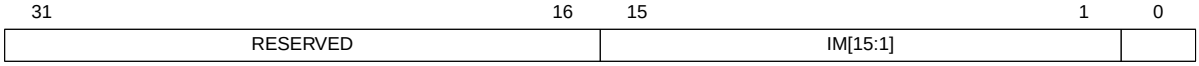
31 : 4	RESERVED
3 : 0	Interrupt status

Table 62. PCIF: Interrupt clear register



31 : 16	RESERVED
15 : 1	Interrupt clear n (IC[n]): Writing ‘1’ to IC[n] will clear interrupt n
0	RESERVED

Table 63. PCIF: Interrupt mask register



31 : 16	RESERVED
15 : 1	Interrupt mask n(IM[n]): If IM[n] = 0 the interrupt n is masked, otherwise it is enabled
0	RESERVED

5.4 Signal definitions and reset values

The signals and their reset values are described in table 64.

Table 64. Signal definitions and reset values

Signal name	Type	Function	Active	Reset value
pci_clk	Input	PCI clock	-	-
pci_rst	Input	Reset	Low	-
pci_gnt	Input	Grant signal	Low	-
pci_idsel	Input	Device select during configuration	High	-
pci_ad[31:0]	Input/Output	Address and Data bus	High	Tri.state
pci_cbe[3:0]	Input/Output	Bus command and byte enable	Low	Tri-state
pci_par	Input/Output	Parity signal	High	Tri-state
pci_frame	Input/Output	Cycle frame	Low	Tri-state
pci_devsel	Input/Output	Device select	Low	Tri-state
pci_irdy	Input/Output	Initiator ready	Low	Tri-state
pci_trdy	Input/Output	Target ready	Low	Tri-state
pci_stop	Input/Output	Stop	Low	Tri-state
pci_perr	Input/Output	Parity error	Low	Tri-state
pci_serr	Input/Output	System error	Low	Tri-state
pci_req	Input/Output	Request signal	Low	Logical 1
pci_int	Output	Interrupt signal	Low	Tri-state

5.5 Timing

The timing waveforms and timing parameters are shown in figure 21 and are defined in table 65.

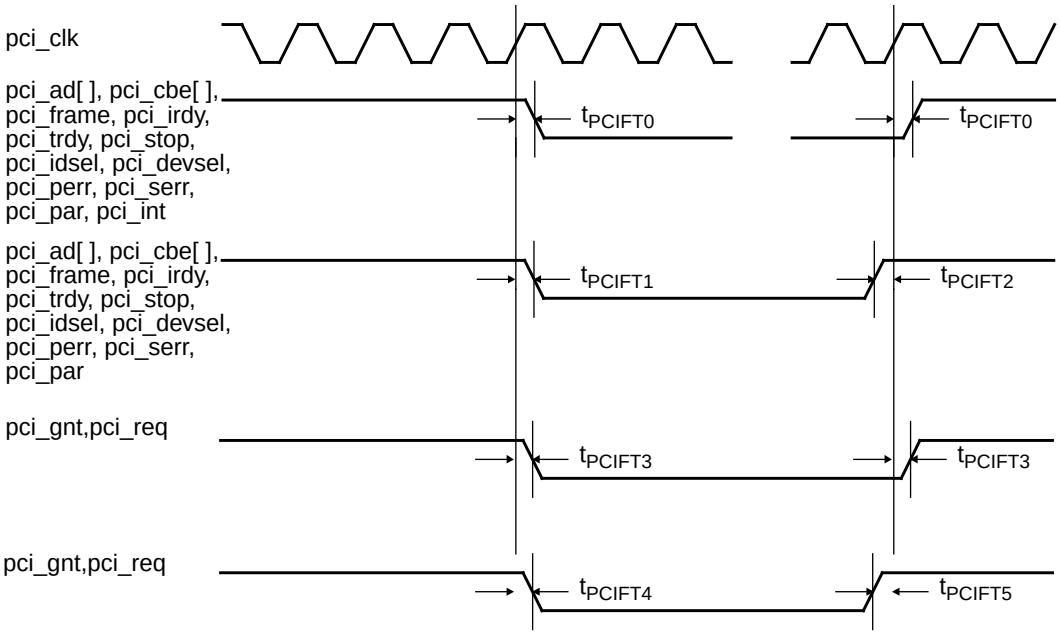


Figure 21. Timing waveforms

Table 65. Timing parameters

Name	Parameter	Reference edge	Min	Max	Unit
$t_{PCICLK0}$	PCI clock period	-	30 *1)*	-	ns
t_{PCIFT0}	clock to output delay	rising clk edge	2	11	ns
t_{PCIFT1}	input to clock hold	rising clk edge	0	-	ns
t_{PCIFT2}	input to clock setup	rising clk edge	7	-	ns
t_{PCIFT3}	clock to output delay	rising clk edge	2	12	ns
t_{PCIFT4}	input to clock hold	rising clk edge	0	-	ns
t_{PCIFT5}	input to clock setup	rising clk edge	10, 12	-	ns

Note 1: The minimum clock period, and the resulting maximum clock frequency, is dependent on the manufacturing lot for the Actel RTAX2000S/SL parts and expected radiation levels.

The degradation criterion for Propagation Delays for the RTAX2000S/SL parts is according to Actel Total Ionizing Dose Test Report 10% at 300 krad (Si). The above specified timing values are guaranteed up to 50 krad (Si).

If a higher total ionizing dose is expected than 50 krad (Si), the corresponding post-annealing propagation delays that can be found in the Actel Total Ionizing Dose Test Report can be used to degrade the timing more accurately (typical degradation is within 1% after 300 krad (Si)).

The functional behavior of the part is guaranteed up to 300 krad (Si).

6 Status Registers

6.1 Overview

The status registers store information about AMBA AHB accesses triggering an error response. There is a status register and a failing address register capturing the control and address signal values of a failing AMBA bus transaction, or the occurrence of a correctable error being signaled from a fault tolerant core.

6.2 Operation

6.2.1 Errors

The registers monitor AMBA AHB bus transactions and store the current HADDR, HWRITE, HMASTER and HSIZE internally. The monitoring are always active after startup and reset until an error response (HRESP = “01”) is detected. When the error is detected, the status and address register contents are frozen and the New Error (NE) bit is set to one. At the same time an interrupt is generated, as described hereunder.

Note that many of the fault tolerant units containing EDAC signal an un-correctable error as an AMBA error response, so that it can be detected by the processor as described above.

6.2.2 Correctable errors

Not only error responses on the AHB bus can be detected. Many of the fault tolerant units containing EDAC have a correctable error signal which is asserted each time a single error is detected. When such an error is detected, the effect will be the same as for an AHB error response. The only difference is that the Correctable Error (CE) bit in the status register is set to one when a single error is detected.

When the CE bit is set the interrupt routine can acquire the address containing the single error from the failing address register and correct it. When it is finished it resets the CE bit and the monitoring becomes active again. Interrupt handling is described in detail hereunder.

Note: The SpaceWire Router does not included any units that can generate a correctable error.

6.2.3 Interrupts

The interrupt is connected to the interrupt controller to inform the processor of the error condition. The normal procedure is that an interrupt routine handles the error with the aid of the information in the status registers. When it is finished it resets the NE bit and the monitoring becomes active again. Interrupts are generated for both AMBA error responses and correctable errors as described above.

6.3 Registers

The core is programmed through registers mapped into APB address space.

Table 66. AHB Status registers

APB address offset	Registers
0x0	AHB Status register
0x4	AHB Failing address register

Table 67. AHB Status register

31 10	9	8	7 6	3	2 0
RESERVED	CE	NE	HWRITE	HMASTER	HSIZE

31: 10	RESERVED
9	CE: Correctable Error. Set if the detected error was caused by a single error and zero otherwise.
8	NE: New Error. Deasserted at start-up and after reset. Asserted when an error is detected. Reset by writing a zero to it.
7	The HWRITE signal of the AHB transaction that caused the error.
6: 3	The HMASTER signal of the AHB transaction that caused the error.
2: 0	The HSIZE signal of the AHB transaction that caused the error

Table 68. AHB Failing address register

31 0
AHB FAILING ADDRESS

31: 0	The HADDR signal of the AHB transaction that caused the error.

7 Serial Debug Interface

7.1 Overview

The interface consists of a UART connected to the AMBA AHB bus as a master. A simple communication protocol is supported to transmit access parameters and data. Through the communication link, a read or write transfer can be generated to any address on the AMBA AHB bus.

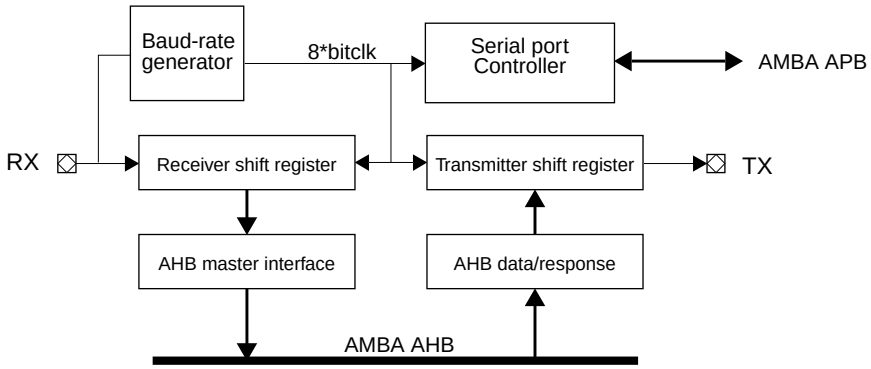


Figure 22. Block diagram

7.2 Operation

7.2.1 Transmission protocol

The interface supports a simple protocol where commands consist of a control byte, followed by a 32-bit address, followed by optional write data. Write access does not return any response, while a read access only returns the read data. Data is sent on 8-bit basis as shown below.

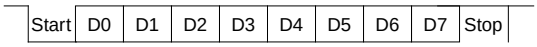


Figure 23. Data frame

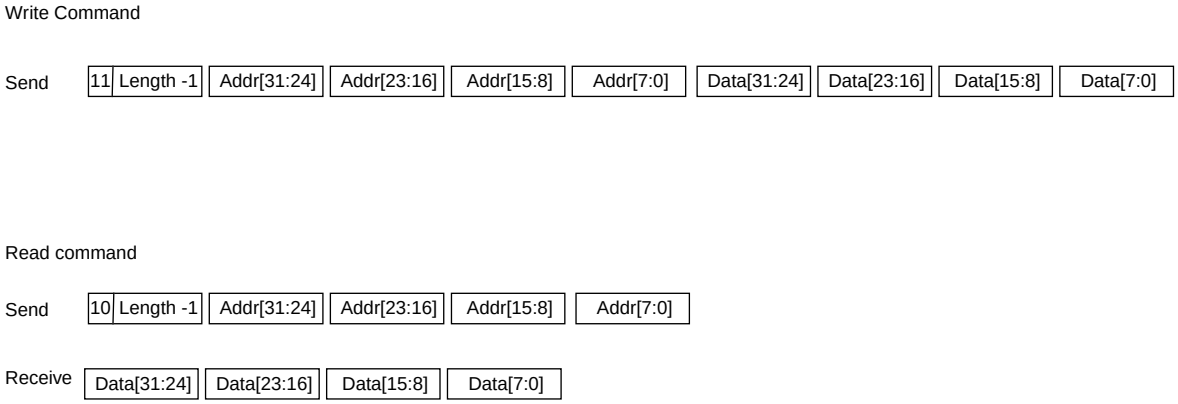


Figure 24. Commands

Block transfers can be performed be setting the length field to *n*-1, where *n* denotes the number of transferred words. For write accesses, the control byte and address is sent once, followed by the number of data words to be written. The address is automatically incremented after each data word. For read accesses, the control byte and address is sent once and the corresponding number of data words is returned.

7.2.2 Baud rate generation

The UART contains a 18-bit down-counting scaler to generate the desired baud-rate. The scaler is clocked by the system clock and generates a UART tick each time it underflows. The scaler is reloaded with the value of the UART scaler reload register after each underflow. The resulting UART tick frequency should be 8 times the desired baud-rate.

If not programmed by software, the baud rate will be automatically discovered. This is done by searching for the shortest period between two falling edges of the received data (corresponding to two bit periods). When three identical two-bit periods has been found, the corresponding scaler reload value is latched into the reload register, and the BL bit is set in the UART control register. If the BL bit is reset by software, the baud rate discovery process is restarted. The baud-rate discovery is also restarted when a ‘break’ or framing error is detected by the receiver, allowing to change to baudrate from the external transmitter. For proper baudrate detection, the value 0x55 should be transmitted to the receiver after reset or after sending break.

The best scaler value for manually programming the baudrate can be calculated as follows:

```
scaler = (((system_clk*10)/(baudrate*8))-5)/10
```

7.3 Registers

The core is programmed through registers mapped into APB address space.

Table 69. AHB UART registers

APB address offset	Register
0x4	AHB UART status register
0x8	AHB UART control register
0xC	AHB UART scaler register

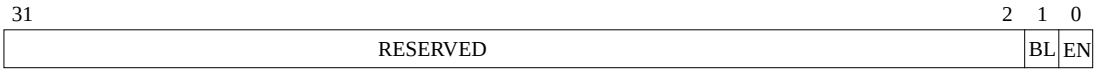


Figure 25. AHB UART control register

0: Receiver enable (EN) - if set, enables both the transmitter and receiver. Reset value: ‘0’.

1: Baud rate locked (BL) - is automatically set when the baud rate is locked. Reset value: ‘0’.

Figure 26. AHB UART status register

0: Data ready (DR) - indicates that new data has been received by the AMBA AHB master interface. Read only. Reset value: ‘0’.

1: Transmitter shift register empty (TS) - indicates that the transmitter shift register is empty. Read only. Reset value: ‘1’.

2: Transmitter hold register empty (TH) - indicates that the transmitter hold register is empty. Read only. Reset value: ‘1’.
3: Break (BR) - indicates that a BREAKE has been received. Reset value: ‘0’.
4: Overrun (OV) - indicates that one or more character have been lost due to overrun. Reset value: ‘0’.
6: Framing error (FE) - indicates that a framing error was detected. Reset value: ‘0’.

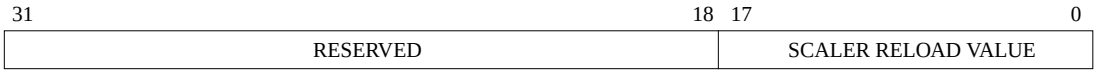


Figure 27. AHB UART scaler reload register

17:0 Baudrate scaler reload value = (((system_clk*10)/(baudrate*8))-5)/10. Reset value: “3FFFF“.

7.4 Signal definitions and reset values

The signals and their reset values are described in table 70.

Table 70. Signal definitions and reset values

Signal name	Type	Function	Active	Reset value
dsutx	Output	UART transmit data line	-	Logical 1
dsurx	Input	UART receive data line	-	-

7.5 Timing

The timing waveforms and timing parameters are shown in figure 28 and are defined in table 71.

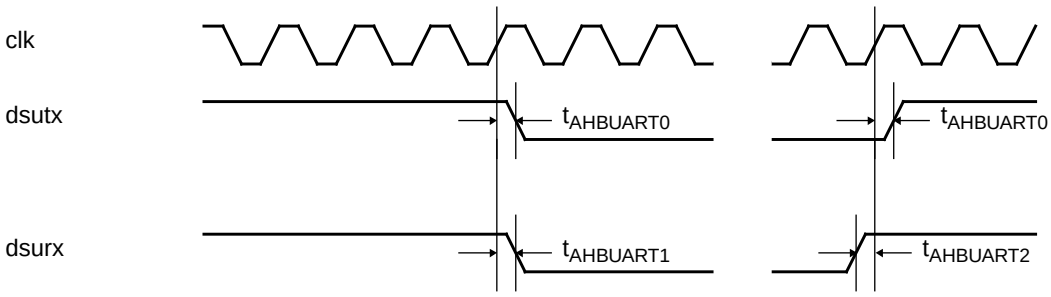


Figure 28. Timing waveforms

Table 71. Timing parameters

Name	Parameter	Reference edge	Min	Max	Unit
$t_{AHBUART0}$	clock to output delay	rising *clk* edge	0	25	ns
$t_{AHBUART1}$	input to clock hold	rising *clk* edge	-	-	ns
$t_{AHBUART2}$	input to clock setup	rising *clk* edge	-	-	ns

Note: The *dsurx* input is re-synchronized internally. The signal does not have to meet any setup or hold requirements.

8 JTAG Debug Interface

8.1 Overview

The JTAG debug interface provides access to on-chip AMBA AHB bus through JTAG. The JTAG debug interface implements a simple protocol which translates JTAG instructions to AHB transfers. Through this link, a read or write transfer can be generated to any address on the AHB bus.

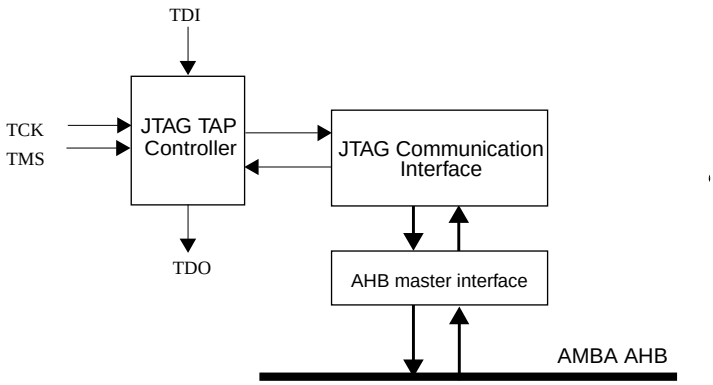


Figure 29. JTAG Debug link block diagram

8.2 Operation

8.2.1 Transmission protocol

The JTAG Debug link decodes two JTAG instructions and implements two JTAG data registers: the command/address register and data register. A read access is initiated by shifting in a command consisting of read/write bit, AHB access size and AHB address into the command/address register. The AHB read access is performed and data is ready to be shifted out of the data register. Write access is performed by shifting in command, AHB size and AHB address into the command/data register followed by shifting in write data into the data register. Sequential transfers can be performed by shifting in command and address for the transfer start address and shifting in SEQ bit in data register for following accesses. The SEQ bit will increment the AHB address for the subsequent access. Sequential transfers should not cross a 1 kByte boundary. Sequential transfers are always word based.

Table 72. JTAG debug link Command/Address register

34	33 32	31 ... 0
W	SIZE	AHB ADDRESS

34	Write (W) - ‘0’ - read transfer, ‘1’ - write transfer
33 32	AHB transfer size - “00” - byte, “01” - half-word, “10” - word, “11”- reserved
31 30	AHB address

Table 73. JTAG debug link Data register

32	31 … 0
SEQ	AHB DATA

32	Sequential transfer (SEQ) - If ‘1’ is shifted in this bit position when read data is shifted out or write data shifted in, the subsequent transfer will be to next word address.
31 30	AHB Data - AHB write/read data. For byte and half-word transfers data is aligned according to big-endian order where data with address offset 0 data is placed in MSB bits.

8.3 Registers

The core does not implement any registers mapped in the AMBA AHB or APB address space.

8.4 Signal definitions and reset values

The signals and their reset values are described in table 74.

Table 74. Signal definitions and reset values

Signal name	Type	Function	Active	Reset value
dsutck	Input	JTAG clock	-	-
dsutms	Input	JTAG TMS	High	-
dsutdi	Input	JTAG TDI	High	-
dsutdo	Output	JTAG TDO	High	undefined

8.5 Timing

The timing waveforms and timing parameters are shown in figure 30 and are defined in table 75.

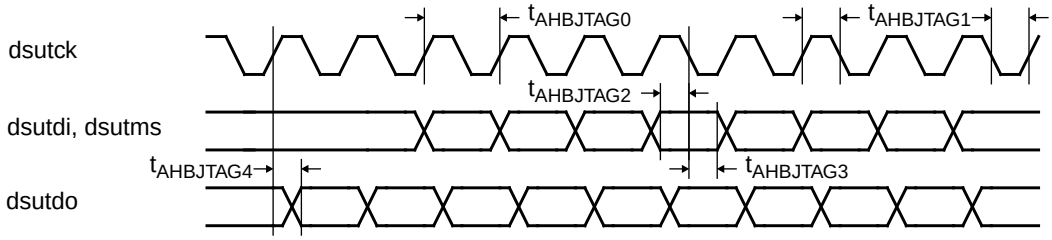


Figure 30. Timing waveforms

Table 75. Timing parameters

Name	Parameter	Reference edge	Min	Max	Unit
$t_{AHBJTAG0}$	clock period	-	100	-	ns
$t_{AHBJTAG1}$	clock low/high period	-	40	-	ns
$t_{AHBJTAG2}$	data input to clock setup	falling *dsutck* edge	15	-	ns
$t_{AHBJTAG3}$	data input from clock hold	falling *dsutck* edge	0	-	ns
$t_{AHBJTAG4}$	clock to data output delay	rising *dsutck* edge	-	25	ns

9 Clock generation

9.1 Overview

The clock generator implements internal clock generation and buffering.

9.2 Signal definitions and reset values

The signals and their reset values are described in table 76.

Table 76. Signal definitions and reset values

Signal name	Type	Function	Active	Reset value
clk	Input	System clock	Rising edge	-

9.3 Timing

The timing waveforms and timing parameters are shown in figure 31 and are defined in table 77.

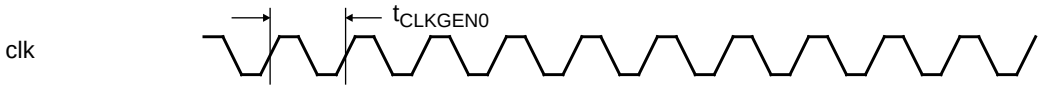


Figure 31. Timing waveforms

Table 77. Timing parameters

Name	Parameter	Reference edge	Min	Max	Unit
$t_{CLKGEN0}$	clock period (CID 1)	-	40	-	ns *1)*
$t_{CLKGEN0}$	clock period (CID 2)	-	30	-	ns *1)*

Note 1: The minimum clock period, and the resulting maximum clock frequency, is dependent on the manufacturing lot for the Actel RTAX2000S/SL parts and expected radiation levels.

The degradation criterion for Propagation Delays for the RTAX2000S/SL parts is according to Actel Total Ionizing Dose Test Report 10% at 300 krad (Si). The above specified timing values are guaranteed up to 50 krad (Si).

If a higher total ionizing dose is expected than 50 krad (Si), the corresponding post-annealing propagation delays that can be found in the Actel Total Ionizing Dose Test Report can be used to degrade the timing more accurately (typical degradation is within 1% after 300 krad (Si)).

The functional behavior of the part is guaranteed up to 300 krad (Si).

10 Reset generation

10.1 Overview

The reset generator implements input reset signal synchronization with glitch filtering and generates the internal reset signal. The input reset signal can be asynchronous.

10.2 Signal definitions and reset values

The signals and their reset values are described in table 78.

Table 78. Signal definitions and reset values

Signal name	Type	Function	Active	Reset value
resetn	Input	Reset	Low	

10.3 Timing

The timing waveforms and timing parameters are shown in figure 32 and are defined in table 79.

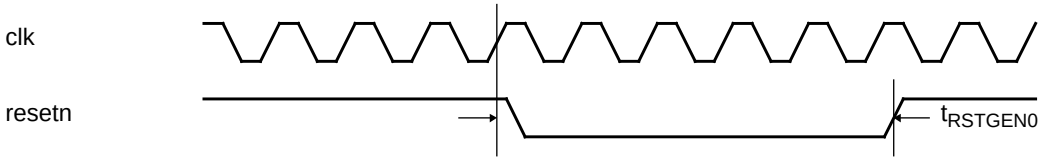


Figure 32. Timing waveforms

Table 79. Timing parameters

Name	Parameter	Reference edge	Min	Max	Unit
$t_{RSTGEN0}$	asserted period	-	100	-	ns

Note: The *resetn* input is re-synchronized internally. The signals does not have to meet any setup or hold requirements.

11 AMBA AHB controller with plug&play support

11.1 Overview

The AMBA AHB controller is a combined AHB arbiter, bus multiplexer and slave decoder according to the AMBA 2.0 standard.

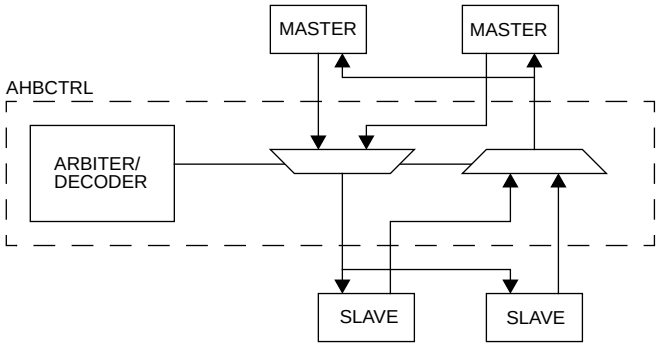


Figure 33. AHB controller block diagram

11.2 Operation

11.2.1 Arbitration

In round-robin mode, priority is rotated one step after each AHB transfer. If no master requests the bus, the last owner will be granted (bus parking).

11.2.2 Decoding

Decoding of AHB slaves is done using the plug&play method explained in the GRLIB User's Manual. A slave can occupy any binary aligned address space with a size of 1 - 4096 Mbyte. A specific I/O area is also decoded, where slaves can occupy 256 byte - 1 Mbyte. The default address of the I/O area is 0xFFF00000. Access to unused addresses will cause an AHB error response.

11.2.3 Plug&play information

The plug&play information is mapped on a read-only address area. By default, the area is mapped on address 0xFFFFF000 - 0xFFFFFFFF. The master information is placed on the first 2 kbyte of the block (0xFFFFF000 - 0xFFFFF800), while the slave information id placed on the second 2 kbyte block. Each unit occupies 32 bytes, which means that the area has place for 64 masters and 64 slaves. The address for masters is thus 0xFFFFF000 + n*32, and 0xFFFFF800 + n*32 for slaves.

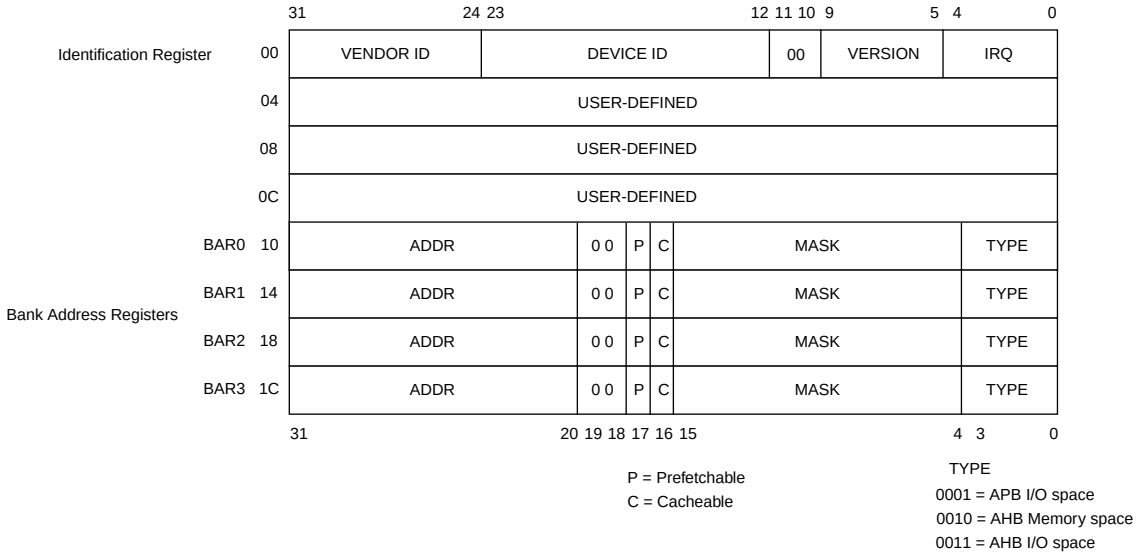


Figure 34. AHB plug&play information record

11.3 Registers

The core does not implement any registers.

12 AMBA AHB/APB bridge with plug&play support

12.1 Overview

The AMBA AHB/APB bridge is a APB bus master according the AMBA 2.0 standard.

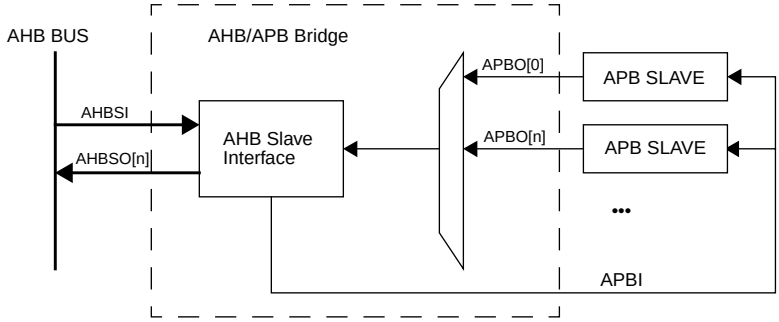


Figure 35. AHB/APB bridge block diagram

12.2 Operation

12.2.1 Decoding

Decoding of APB slaves is done using the plug&play method explained in the GRLIB IP Library User's Manual. A slave can occupy any binary aligned address space with a size of 256 bytes - 1 Mbyte.

12.2.2 Plug&play information

The plug&play information is mapped on a read-only address area at the top 4 kbytes of the bridge address space. Each plug&play block occupies 8 bytes. If the bridge is mapped on AHB address 0xFFE00000, the address for the plug&play records is thus 0xFFEFF000 + n*8.

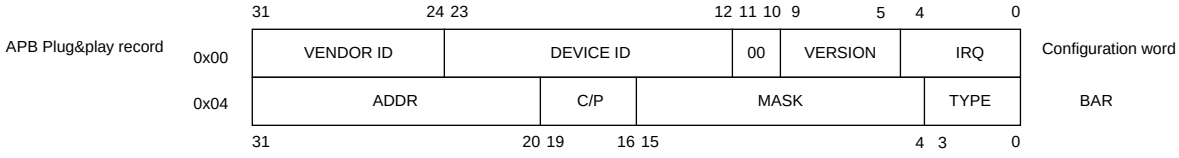


Figure 36. APB plug&play information

13 Electrical description

13.1 Absolute maximum ratings

According to Actel data sheet [RTAX] and [RT3PE].

13.2 Operating conditions

According to Actel data sheet [RTAX] and [RT3PE].

13.3 Input voltages, leakage currents and capacitances

According to Actel data sheet [RTAX] and [RT3PE].

13.4 Output voltages, leakage currents and capacitances

According to Actel data sheet [RTAX] and [RT3PE].

All output timing parameters are defined at 35 pF capacitive load, except for the LVDS and PCI electrical standards.

13.5 Clock Input voltages, leakage currents and capacitances

According to Actel data sheet [RTAX] and [RT3PE].

13.6 Power supplies

According to Actel data sheet [RTAX] and [RT3PE].

14 Mechanical description

14.1 Component and package

All configurations are implemented in the ACTEL RTAX2000S/SL or RT3PE3000L FPGAs. Configuration 1 is provided in either CQ352 or CG624 package, or CG484 package, as per Actel data sheets [RTAX], [RT3PE] and [PACK]. Configuration 2 is only available in CQ352 or CG624 package.

14.2 Pin assignment

The pin assignment in table 80 shows the implementation characteristics of each signal according to the Actel data sheet [RTAX] and [RT3PE], indicating how each pin has been configured in terms of electrical levels, voltage, slew rate, drive capability and internal pull-up or pull-down in the FPGA device.High

Table 80. Pin assignment

Name	I/O	Pin CG484	Pin CQ352	Pin CG624	Level	Volt.	Slew	Drive	Load [pF]	Pull	Pol-arity	Note	CID
clk	in			C12	LVTTL	3.3				None		System clock	All
resetn	in			D24	LVTTL	3.3				None	Low	Reset	All
dsutx	out			F23	LVTTL	3.3	Low	12	35	None	Low	Debug UART data transmit	2
dsurx	in			E23	LVTTL	3.3				None	Low	Debug UART data receive	2
dsutck	in			U23	LVCMOS	2.5				None		Debug JTAG clock	
dsutms	in			V23	LVCMOS	2.5				None	High	Debug JTAG mode	
dsutdi	in			Y24	LVCMOS	2.5				None	High	Debug JTAG input	
dsutdo	out			V22	LVCMOS	2.5	Low	12	35	None	High	Debug JTAG output	
rxclk	in			P23	LVTTL	3.3				None		SpaceWire receiver clock	All
txclk	in			P19	LVTTL	3.3				None		SpaceWire transmitter clock	All
clk2	out			R18	LVTTL	3.3	High	12	35	None		Clock with 1/2 the freq of rxclk	All
clk4	out			R24	LVTTL	3.3	High	12	35	None		Clock with 1/4 the freq of rxclk	All
clk8	out			T19	LVTTL	3.3	High	12	35	None		Clock with 1/8 the freq of rxclk	All
spw_rxdp[0]	in			P19	LVDS	2.5					High	SpaceWire data	All 1)
spw_rxdn[0]	in			P20	LVDS	2.5					Low		All 1)
spw_rxsp[0]	in			P23	LVDS	2.5					High	SpaceWire strobe	All 1)
spw_rxsn[0]	in			R23	LVDS	2.5					Low		All 1)
spw_txdp[0]	out			P25	LVDS	2.5					High	SpaceWire data	All 1)
spw_txdn[0]	out			R25	LVDS	2.5					Low		All 1)
spw_txsp[0]	out			P22	LVDS	2.5					High	SpaceWire strobe	All 1)
spw_txsn[0]	out			R22	LVDS	2.5					Low		All 1)
spw_rxdp[1]	in			R18	LVDS	2.5					High	SpaceWire data	All 1)
spw_rxdn[1]	in			T18	LVDS	2.5					Low		All 1)
spw_rxsp[1]	in			R24	LVDS	2.5					High	SpaceWire strobe	All 1)
spw_rxsn[1]	in			T24	LVDS	2.5					Low		All 1)
spw_txdp[1]	out			T25	LVDS	2.5					High	SpaceWire data	All 1)
spw_txdn[1]	out			U25	LVDS	2.5					Low		All 1)
spw_txsp[1]	out			R20	LVDS	2.5					High	SpaceWire strobe	All 1)

Table 80. Pin assignment

Name	I/O	Pin CG484	Pin CQ352	Pin CG624	Level	Volt.	Slew	Drive	Load [pF]	Pull	Pol-arity	Note	CID
spw_txsn[1]	out			T20	LVDS	2.5					Low		All 1)
spw_rxdp[2]	in			T19	LVDS	2.5					High	SpaceWire data	All 1)
spw_rxdn[2]	in			U19	LVDS	2.5					Low		All 1)
spw_rxsp[2]	in			W25	LVDS	2.5					High	SpaceWire strobe	All 1)
spw_rxsn[2]	in			Y25	LVDS	2.5					Low		All 1)
spw_txdp[2]	out			AA25	LVDS	2.5					High	SpaceWire data	All 1)
spw_txdn[2]	out			AB25	LVDS	2.5					Low		All 1)
spw_txsp[2]	out			U20	LVDS	2.5					High	SpaceWire strobe	All 1)
spw_txsn[2]	out			V20	LVDS	2.5					Low		All 1)
spw_rxdp[3]	in			U23	LVDS	2.5					High	SpaceWire data	All 1)
spw_rxdn[3]	in			U24	LVDS	2.5					Low		All 1)
spw_rxsp[3]	in			Y24	LVDS	2.5					High	SpaceWire strobe	All 1)
spw_rxsn[3]	in			AA24	LVDS	2.5					Low		All 1)
spw_txdp[3]	out			V24	LVDS	2.5					High	SpaceWire data	All 1)
spw_txdn[3]	out			V23	LVDS	2.5					Low		All 1)
spw_txsp[3]	out			U22	LVDS	2.5					High	SpaceWire strobe	All 1)
spw_txsn[3]	out			V22	LVDS	2.5					Low		All 1)
spw_rxdp[4]	in				LVDS	2.5					High	SpaceWire data	1 1)
spw_rxdn[4]	in				LVDS	2.5					Low		1 1)
spw_rxsp[4]	in				LVDS	2.5					High	SpaceWire strobe	1 1)
spw_rxsn[4]	in				LVDS	2.5					Low		1 1)
spw_txdp[4]	out				LVDS	2.5					High	SpaceWire data	1 1)
spw_txdn[4]	out				LVDS	2.5					Low		1 1)
spw_txsp[4]	out				LVDS	2.5					High	SpaceWire strobe	1 1)
spw_txsn[4]	out				LVDS	2.5					Low		1 1)
spw_rxdp[5]	in				LVDS	2.5					High	SpaceWire data	1 1)
spw_rxdn[5]	in				LVDS	2.5					Low		1 1)
spw_rxsp[5]	in				LVDS	2.5					High	SpaceWire strobe	1 1)
spw_rxsn[5]	in				LVDS	2.5					Low		1 1)
spw_txdp[5]	out				LVDS	2.5					High	SpaceWire data	1 1)
spw_txdn[5]	out				LVDS	2.5					Low		1 1)
spw_txsp[5]	out				LVDS	2.5					High	SpaceWire strobe	1 1)
spw_txsn[5]	out				LVDS	2.5					Low		1 1)
spw_rxdp[6]	in				LVDS	2.5					High	SpaceWire data	1 1)
spw_rxdn[6]	in				LVDS	2.5					Low		1 1)
spw_rxsp[6]	in				LVDS	2.5					High	SpaceWire strobe	1 1)
spw_rxsn[6]	in				LVDS	2.5					Low		1 1)

Table 80. Pin assignment

Name	I/O	Pin CG484	Pin CQ352	Pin CG624	Level	Volt.	Slew	Drive	Load [pF]	Pull	Pol-arity	Note	CID
spw_txdp[6]	out				LVDS	2.5					High	SpaceWire data	1 1)
spw_txdn[6]	out				LVDS	2.5					Low		1 1)
spw_txsp[6]	out				LVDS	2.5					High	SpaceWire strobe	1 1)
spw_txsn[6]	out				LVDS	2.5					Low		1 1)
spw_rxdp[7]	in				LVDS	2.5					High	SpaceWire data	1 1)
spw_rxdn[7]	in				LVDS	2.5					Low		1 1)
spw_rxsp[7]	in				LVDS	2.5					High	SpaceWire strobe	1 1)
spw_rxsn[7]	in				LVDS	2.5					Low		1 1)
spw_txdp[7]	out				LVDS	2.5					High	SpaceWire data	1 1)
spw_txdn[7]	out				LVDS	2.5					Low		1 1)
spw_txsp[7]	out				LVDS	2.5					High	SpaceWire strobe	1 1)
spw_txsn[7]	out				LVDS	2.5					Low		1 1)
spw_rxd[0]	in			H24	LVTTL	3.3				None	High	SpaceWire data	All 2)
spw_rxs[0]	in			N16	LVTTL	3.3				None	High	SpaceWire strobe	All 2)
spw_txd[0]	out			M19	LVTTL	3.3	High	8	35	None	High	SpaceWire data	All 2)
spw_txs[0]	out			M18	LVTTL	3.3	High	8	35	None	High	SpaceWire strobe	All 2)
spw_rxd[1]	in			K22	LVTTL	3.3				None	High	SpaceWire data	All 2)
spw_rxs[1]	in			F25	LVTTL	3.3				None	High	SpaceWire strobe	All 2)
spw_txd[1]	out			L21	LVTTL	3.3	High	8	35	None	High	SpaceWire data	All 2)
spw_txs[1]	out			L20	LVTTL	3.3	High	8	35	None	High	SpaceWire strobe	All 2)
spw_rxd[2]	in			K24	LVTTL	3.3				None	High	SpaceWire data	All 2)
spw_rxs[2]	in			M21	LVTTL	3.3				None	High	SpaceWire strobe	All 2)
spw_txd[2]	out			N17	LVTTL	3.3	High	8	35	None	High	SpaceWire data	All 2)
spw_txs[2]	out			J25	LVTTL	3.3	High	8	35	None	High	SpaceWire strobe	All 2)
spw_rxd[3]	in			N18	LVTTL	3.3				None	High	SpaceWire data	All 2)
spw_rxs[3]	in			M24	LVTTL	3.3				None	High	SpaceWire strobe	All 2)
spw_txd[3]	out			L25	LVTTL	3.3	High	8	35	None	High	SpaceWire data	All 2)
spw_txs[3]	out			K25	LVTTL	3.3	High	8	35	None	High	SpaceWire strobe	All 2)
spw_rxd[4]	in			M22	LVTTL	3.3				None	High	SpaceWire data	1 2)
spw_rxs[4]	in			M23	LVTTL	3.3				None	High	SpaceWire strobe	1 2)
spw_txd[4]	out			P17	LVTTL	3.3	High	8	35	None	High	SpaceWire data	1 2)
spw_txs[4]	out			P18	LVTTL	3.3	High	8	35	None	High	SpaceWire strobe	1 2)
spw_rxd[5]	in			U21	LVTTL	3.3				None	High	SpaceWire data	1 2)
spw_rxs[5]	in			AA23	LVTTL	3.3				None	High	SpaceWire strobe	1 2)
spw_txd[5]	out			W24	LVTTL	3.3	High	8	35	None	High	SpaceWire data	1 2)
spw_txs[5]	out			AB24	LVTTL	3.3	High	8	35	None	High	SpaceWire strobe	1 2)
spw_rxd[6]	in			U22	LVTTL	3.3				None	High	SpaceWire data	1 2)

Table 80. Pin assignment

Name	I/O	Pin CG484	Pin CQ352	Pin CG624	Level	Volt.	Slew	Drive	Load [pF]	Pull	Pol-arity	Note	CID
spw_rxs[6]	in			V24	LVTTL	3.3				None	High	SpaceWire strobe	1 [2)]
spw_txd[6]	out			R21	LVTTL	3.3	High	8	35	None	High	SpaceWire data	1 [2)]
spw_txs[6]	out			T22	LVTTL	3.3	High	8	35	None	High	SpaceWire strobe	1 [2)]
spw_rxd[7]	in			W23	LVTTL	3.3				None	High	SpaceWire data	1 [2)]
spw_rxs[7]	in			Y21	LVTTL	3.3				None	High	SpaceWire strobe	1 [2)]
spw_txd[7]	out			R19	LVTTL	3.3	High	8	35	None	High	SpaceWire data	1 [2)]
spw_txs[7]	out			P21	LVTTL	3.3	High	8	35	None	High	SpaceWire strobe	1 [2)]
timecodeen	in			D10	LVTTL	3.3				Up	High	Enable timecode function	1
tickin[0]	in			U2	LVTTL	3.3				None	High	Tick input for FIFO 0	1
timein0[0]	in			R5	LVTTL	3.3				None		Time input for FIFO 0, LSB	1
timein0[1]	in			H5	LVTTL	3.3				None			1
timein0[2]	in			AA1	LVTTL	3.3				None			1
timein0[3]	in			T7	LVTTL	3.3				None			1
timein0[4]	in			AB1	LVTTL	3.3				None			1
timein0[5]	in			U7	LVTTL	3.3				None			1
timein0[6]	in			R8	LVTTL	3.3				None			1
timein0[7]	in			V2	LVTTL	3.3				None		Time input for FIFO 0, MSB	1
tickout[0]	out			T2	LVTTL	3.3	High	8	35	None	High	Tick output for FIFO 0	1
timeout0[0]	out			M3	LVTTL	3.3	High	8	35	None		Time output for FIFO 0, LSB	1
timeout0[1]	out			N9	LVTTL	3.3	High	8	35	None			1
timeout0[2]	out			J3	LVTTL	3.3	High	8	35	None			1
timeout0[3]	out			J4	LVTTL	3.3	High	8	35	None			1
timeout0[4]	out			T4	LVTTL	3.3	High	8	35	None			1
timeout0[5]	out			H4	LVTTL	3.3	High	8	35	None			1
timeout0[6]	out			R3	LVTTL	3.3	High	8	35	None			1
timeout0[7]	out			H6	LVTTL	3.3	High	8	35	None		Time output for FIFO 0, MSB	1
tickin[1]	in			F3	LVTTL	3.3				None	High	Tick input for FIFO 1	1
timein1[0]	in			F2	LVTTL	3.3				None		Time input for FIFO 1, LSB	1
timein1[1]	in			J5	LVTTL	3.3				None			1
timein1[2]	in			E2	LVTTL	3.3				None			1
timein1[3]	in			J6	LVTTL	3.3				None			1
timein1[4]	in			H2	LVTTL	3.3				None			1
timein1[5]	in			F1	LVTTL	3.3				None			1
timein1[6]	in			M9	LVTTL	3.3				None			1
timein1[7]	in			L7	LVTTL	3.3				None		Time input for FIFO 1, MSB	1
tickout[1]	out			E3	LVTTL	3.3	High	8	35	None	High	Tick output for FIFO 1	1
timeout1[0]	out			J2	LVTTL	3.3	High	8	35	None		Time output for FIFO 1, LSB	1
timeout1[1]	out			L4	LVTTL	3.3	High	8	35	None			1
timeout1[2]	out			L2	LVTTL	3.3	High	8	35	None			1
timeout1[3]	out			L5	LVTTL	3.3	High	8	35	None			1
timeout1[4]	out			K1	LVTTL	3.3	High	8	35	None			1
timeout1[5]	out			E1	LVTTL	3.3	High	8	35	None			1

Table 80. Pin assignment

Name	I/O	Pin CG484	Pin CQ352	Pin CG624	Level	Volt.	Slew	Drive	Load [pF]	Pull	Pol-arity	Note	CID
timeout1[6]	out			H3	LVTTL	3.3	High	8	35	None			1
timeout1[7]	out			D1	LVTTL	3.3	High	8	35	None		Time output for FIFO 1, MSB	1
rxcharav[0]	out			F19	LVTTL	3.3	High	8	35	None	High	Rx data available for FIFO 0	1
rxaempty[0]	out			K19	LVTTL	3.3	High	8	35	None	High	Rx empty for FIFO 0	1
rxread[0]	in			J22	LVTTL	3.3				None	High	Rx read for FIFO 0	1
rxchar0[0]	out			G22	LVTTL	3.3	High	8	35	None		Rx character for FIF0 0	1
rxchar0[1]	out			M17	LVTTL	3.3	High	8	35	None			1
rxchar0[2]	out			H20	LVTTL	3.3	High	8	35	None			1
rxchar0[3]	out			H19	LVTTL	3.3	High	8	35	None			1
rxchar0[4]	out			E18	LVTTL	3.3	High	8	35	None			1
rxchar0[5]	out			F18	LVTTL	3.3	High	8	35	None			1
rxchar0[6]	out			G21	LVTTL	3.3	High	8	35	None			1
rxchar0[7]	out			G20	LVTTL	3.3	High	8	35	None			1
rxchar0[8]	out			F20	LVTTL	3.3	High	8	35	None			1
txfull[0]	out			E25	LVTTL	3.3	High	8	35	None	High	Tx full for FIFO 0	1
txafull[0]	out			D25	LVTTL	3.3	High	8	35	None	High	Tx almost full for FIFO 0	1
txwrite[0]	in			K20	LVTTL	3.3				None	High	Tx write for FIFO 0	1
txchar0[0]	in			J21	LVTTL	3.3				None		Tx character for FIFO 0	1
txchar0[1]	in			J20	LVTTL	3.3				None			1
txchar0[2]	in			J23	LVTTL	3.3				None			1
txchar0[3]	in			L19	LVTTL	3.3				None			1
txchar0[4]	in			F24	LVTTL	3.3				None			1
txchar0[5]	in			G24	LVTTL	3.3				None			1
txchar0[6]	in			K18	LVTTL	3.3				None			1
txchar0[7]	in			L18	LVTTL	3.3				None			1
txchar0[8]	in			H22	LVTTL	3.3				None			1
rxcharav[1]	out			R2	LVTTL	3.3	High	8	35	None	High	Rx data available for FIFO 1	1
rxaempty[1]	out			N4	LVTTL	3.3	High	8	35	None	High	Rx empty for FIFO 1	1
rxread[1]	in			U1	LVTTL	3.3				None	High	Rx read for FIFO 1	1
rxchar1[0]	out			T8	LVTTL	3.3	High	8	35	None		Rx character for FIF0 0	1
rxchar1[1]	out			M5	LVTTL	3.3	High	8	35	None			1
rxchar1[2]	out			N10	LVTTL	3.3	High	8	35	None			1
rxchar1[3]	out			L3	LVTTL	3.3	High	8	35	None			1
rxchar1[4]	out			N7	LVTTL	3.3	High	8	35	None			1
rxchar1[5]	out			M6	LVTTL	3.3	High	8	35	None			1
rxchar1[6]	out			P9	LVTTL	3.3	High	8	35	None			1
rxchar1[7]	out			P7	LVTTL	3.3	High	8	35	None			1
rxchar1[8]	out			N1	LVTTL	3.3	High	8	35	None			1
txfull[1]	out			R1	LVTTL	3.3	High	8	35	None	High	Tx full for FIFO 1	1
txafull[1]	out			U3	LVTTL	3.3	High	8	35	None	High	Tx almost full for FIFO 1	1
txwrite[1]	in			T1	LVTTL	3.3				None	High	Transmitter write for FIFO 1	1
txchar1[0]	in			Y1	LVTTL	3.3				None		Tx character for FIFO 1	1
txchar1[1]	in			P2	LVTTL	3.3				None			1

Table 80. Pin assignment

Name	I/O	Pin CG484	Pin CQ352	Pin CG624	Level	Volt.	Slew	Drive	Load [pF]	Pull	Pol-arity	Note	CID
txchar1[2]	in			K7	LVTTL	3.3				None			1
txchar1[3]	in			K6	LVTTL	3.3				None			1
txchar1[4]	in			V4	LVTTL	3.3				None			1
txchar1[5]	in			Y5	LVTTL	3.3				None			1
txchar1[6]	in			W5	LVTTL	3.3				None			1
txchar1[7]	in			U6	LVTTL	3.3				None			1
txchar1[8]	in			U5	LVTTL	3.3				None			1
enbridge[0]	in			E24	LVTTL	3.3				None	High	Enables bridge mode for FIFO 0	1
enbridge[1]	in			P5	LVTTL	3.3				None	High	Enables bridge mode for FIFO 1	1
linkrun[0]	out			P8	LVTTL	3.3	Low	8	35	None	Low	SpaceWire link in run state	All
linkrun[1]	out			G2	LVTTL	3.3	Low	8	35	None	Low		All
linkrun[2]	out			G4	LVTTL	3.3	Low	8	35	None	Low		All
linkrun[3]	out			P3	LVTTL	3.3	Low	8	35	None	Low		All
linkrun[4]	out			D2	LVTTL	3.3	Low	8	35	None	Low		1
linkrun[5]	out			R4	LVTTL	3.3	Low	8	35	None	Low		1
linkrun[6]	out			W1	LVTTL	3.3	Low	8	35	None	Low		1
linkrun[7]	out			D20	LVTTL	3.3	Low	8	35	None	Low		1
gerror	out			G7	LVTTL	3.3	Low	8	35	None	Low	Global error	All
reload_ps[0]	in			D9	LVTTL	3.3				Up		Timer prescaler reset value	All
reload_ps[1]	in			A7	LVTTL	3.3				Up			All
reload_ps[2]	in			A6	LVTTL	3.3				Up			All
reload_ps[3]	in			G9	LVTTL	3.3				Up			All
reload_ps[4]	in			M4	LVTTL	3.3				Up			All
reload_ps[5]	in			P1	LVTTL	3.3				Up			All
reload_timer[0]	in			G8	LVTTL	3.3				Up		Port timer registers reset value	All
reload_timer[1]	in			A22	LVTTL	3.3				Up			All
reload_timer[2]	in			N8	LVTTL	3.3				Up			All
reload_timer[3]	in			W4	LVTTL	3.3				Up			All
timeren	in			P6	LVTTL	3.3				Down	High	Rreset value timer enable bit	All
selfaddren	in			G3	LVTTL	3.3				Up	High	Reset value selfaddren bit	All
linkstartreq	in			V6	LVTTL	3.3				Down	High	Reset value linkstartreq bit	All
autodconnect	in			AB2	LVTTL	3.3				Down	High	Reset value autodconnect bit	All
instanceid[0]	in			A21	LVTTL	3.3				Down		Instance ID reset value	All
instanceid[1]	in			F16	LVTTL	3.3				Down			All
instanceid[2]	in			R7	LVTTL	3.3				Down			All
instanceid[3]	in			U4	LVTTL	3.3				Down			All
idivisor[0]	in			AA3	LVTTL	3.3				Up		Initialization divisor reset value	All
idivisor[1]	in			L8	LVTTL	3.3				Up			All
idivisor[2]	in			K8	LVTTL	3.3				Down			All
idivisor[3]	in			AA2	LVTTL	3.3				Down			All
idivisor[4]	in			J18	LVTTL	3.3				Up			All
cfglock	in			N6	LVTTL	3.3				Down	High	Lock configuration port	All

Table 80. Pin assignment

Name	I/O	Pin CG484	Pin CQ352	Pin CG624	Level	Volt.	Slew	Drive	Load [pF]	Pull	Pol-arity	Note	CID
pci_clk	in			G14	PCI	3.3						PCI clock	2
pci_rst	in			AB8	PCI	3.3						PCI reset	2
pci_frame	inout			AC9	PCI	3.3							2
pci_irdy	inout			AA9	PCI	3.3							2
pci_trdy	inout			AD6	PCI	3.3							2
pci_stop	inout			AD5	PCI	3.3							2
pci_idsel	in			U8	PCI	3.3							2
pci_devsel	inout			AB9	PCI	3.3							2
pci_par	inout			V10	PCI	3.3							2
pci_perr	inout			V9	PCI	3.3							2
pci_serr	inout			AD8	PCI	3.3							2
pci_req	inout			AE11	PCI	3.3							2
pci_gnt	in			AE10	PCI	3.3							2
pci_int	out			AA3	PCI	3.3						PCI interrupt	2
pci_cbe[0]	inout			W8	PCI	3.3						PCI byte enable, LSB	2
pci_cbe[1]	inout			W9	PCI	3.3							2
pci_cbe[2]	inout			AE4	PCI	3.3							2
pci_cbe[3]	inout			AE5	PCI	3.3						PCI byte enable, MSB	2
pci_ad[0]	inout			W11	PCI	3.3						PCI address/data, LSB	2
pci_ad[1]	inout			W12	PCI	3.3							2
pci_ad[2]	inout			AA11	PCI	3.3							2
pci_ad[3]	inout			Y11	PCI	3.3							2
pci_ad[4]	inout			AE9	PCI	3.3							2
pci_ad[5]	inout			AE6	PCI	3.3							2
pci_ad[6]	inout			AE7	PCI	3.3							2
pci_ad[7]	inout			Y10	PCI	3.3							2
pci_ad[8]	inout			W10	PCI	3.3							2
pci_ad[9]	inout			T13	PCI	3.3							2
pci_ad[10]	inout			AB10	PCI	3.3							2
pci_ad[11]	inout			AB11	PCI	3.3							2
pci_ad[12]	inout			AD9	PCI	3.3							2
pci_ad[13]	inout			AD10	PCI	3.3							2
pci_ad[14]	inout			V11	PCI	3.3							2
pci_ad[15]	inout			AD7	PCI	3.3							2

Table 80. Pin assignment

Name	I/O	Pin CG484	Pin CQ352	Pin CG624	Level	Volt.	Slew	Drive	Load [pF]	Pull	Pol-arity	Note	CID
pci_ad[16]	inout			AC8	PCI	3.3							2
pci_ad[17]	inout			AB7	PCI	3.3							2
pci_ad[18]	inout			AC7	PCI	3.3							2
pci_ad[19]	inout			AA8	PCI	3.3							2
pci_ad[20]	inout			Y8	PCI	3.3							2
pci_ad[21]	inout			V8	PCI	3.3							2
pci_ad[22]	inout			V7	PCI	3.3							2
pci_ad[23]	inout			Y7	PCI	3.3							2
pci_ad[24]	inout			W7	PCI	3.3							2
pci_ad[25]	inout			AC5	PCI	3.3							2
pci_ad[26]	inout			AC6	PCI	3.3							2
pci_ad[27]	inout			Y6	PCI	3.3							2
pci_ad[28]	inout			W6	PCI	3.3							2
pci_ad[29]	inout			AB6	PCI	3.3							2
pci_ad[30]	inout			AA6	PCI	3.3							2
pci_ad[31]	inout			Y3	PCI	3.3						PCI address/data, MSB	2
reserved	out		N/A	N2	LVTTL	3.3						Driven to logical 1	All
reserved	out		N/A	M2	LVTTL	3.3						Driven to logical 1	All
reserved	out		N/A	P4	LVTTL	3.3						Driven to logical 1	All
reserved	out		N/A	M1	LVTTL	3.3						Driven to logical 1	All

Pins not used in a given configuration (see CID column) should be left unconnected (or tied to ground). The only exception to this are unused pins mapped on a hardwired clock input or a routed clock input, which should then be tied to ground. The following signals use hardwired clocks in the router configurations: clk, pci_clk.

All the global clock pins can be found here (note they include *all* clock pins both used and unused):

CQ352 package: TBS

CG624 package: CLK and HCLK rows in table 82.

CG484 package: TBD

Pins marked as *{reserved}* must be left unconnected.

For the usage of all other pins, please refer to the specific pins of the selected package, as described in the next sections and the Actel data sheets [RTAX] and [RT3PE].

Note 1: These signals are only used in configuration with on-chip LVDS drivers.

Note 2: These signals are only used in configuration with off-chip LVDS drivers.

14.3 RTAX2000S/SL specific pins - CQ352 package

The Actel RTAX2000S/SL FPGA device has special pins that need to be correctly connected on the printed circuit board, as shown in table 81. Please refer to the Actel data sheet [RTAX] for details.

Table 81. RTAX2000S/SL special pins - CQ352 package

Name	Pin CQ352	Note	CID
GND	1, 9, 15, 21, 27, 33, 39, 45, 51, 57, 63, 69, 75, 81, 88, 89, 97, 103, 109, 115, 121, 133, 145, 151, 157, 163, 169, 176, 177, 186, 192, 198, 204, 210, 216, 222, 228, 234, 240, 246, 252, 258, 264, 265, 274, 280, 286, 292, 298, 310, 322, 330, 334, 340, 345, 352	Low supply voltage, ground	All
VCCA	3, 14, 32, 56, 74, 87, 102, 114, 150, 162, 175, 191, 209, 233, 251, 263, 279, 291, 329, 339	1.5 V supply voltage for array	All
VCCDA	2, 44, 90, 91, 116, 117, 130, 131, 132, 148, 149, 174, 178, 221, 266, 268, 293, 294, 307, 308, 309, 327, 328, 346	3.3 V supply voltage for I/O differential amplifier and JTAG and probe interfaces.	All
VCCIB0	321, 333, 344	3.3 V supply voltage for I/O	All
VCCIB1	273, 285, 297	3.3 V supply voltage for I/O	All
VCCIB2	227, 239, 245, 257	3.3 V supply voltage for I/O	All
VCCIB3	185, 197, 203, 215	3.3 V supply voltage for I/O	All
VCCIB4	144, 156, 168	3.3 V supply voltage for I/O	All
VCCIB5	96, 108, 120	3.3 V supply voltage for I/O	All
VCCIB6	50, 62, 68, 80	3.3 V supply voltage for I/O	All
VCCIB7	8, 20, 26, 38	2.5 V supply voltage for I/O (TBD)	All
VCCIB7	8, 20, 26, 38	2.5 V supply voltage for I/O (TBD)	All
VPUMP	267	Voltage External Pump. In normal operation, using the internal charge pump, should be tied to ground.	All
TRST	351	JTAG Test Clock. Actel recommends this pin be hardwired to ground for flight.	All
TCK	349	JTAG Test Clock. Actel recommends this pin be hardwired to ground. Must not be left unconnected.	All
TDI	348	JTAG Test Data Input. Actel recommends this pin be hardwired to VCCDA, or left unconnected.	All
TDO	347	JTAG Test Data Output. Must be left unconnected.	All
TMS	350	JTAG Test Mode Select. Actel recommends that this pin be hardwired to VCCDA, or left unconnected.	All
PRA	312	Test Probe. The pins' probe capabilities are disabled to protect programmed design confidentiality. These pins must be left unconnected.	All
PRB	311		All
PRC	135		All
PRD	134		All
NC	124, 125, 126, 127, 138, 139, 140, 141, 301, 302, 303, 304, 315, 316, 317, 318	No Connection. Pins are not connected to circuitry within the device. These pins can be driven to any voltage or can be left floating with no effect on the operation of the device.	All
CLK*	122, 123, 128, 129, 136, 137, 142, 143	Routed clocks. When pins are unused, Actel recommends they are tied to known state, preferably ground.	All
HCLK*	299, 300, 305, 306, 313, 314, 319, 320	Hardwired clocks. When pins are unused, it is recommended that they are tied to ground.	All

14.4 RTAX2000S/SL specific pins - CG624 package

The Actel RTAX2000S/SL FPGA device has special pins that need to be correctly connected on the printed circuit board, as shown in table 81. Please refer to the Actel data sheet [RTAX] for details.

Table 82. RTAX2000S/SL special pins - CG624 package

Name	Pin CG624	Note	CID
GND	A18, A24, A25, A2, A8, AA10, AA16, AA18, AA21, AA5, AB22, AB4, AC10, AC16, AC23, AC3, AD1, AD2, AD24, AD25, AE1, AE18, AE24, AE2, AE25, AE8, B1, B2, B25, B24, C10, C16, C23, C3, D22, D4, E10, E16, E21, E5, E8, H1, H21, H25, K21, K23, K3, K5, L11, L12, L13, L14, L15, M11, M12, M13, M14, M15, N11, N12, N13, N14, N15, P11, P12, P13, P14, P15, R11, R12, R13, R14, R15, T21, T23, T3, T5, V1, V25, V5	Low supply voltage, ground	All
VCCA	AB20, F22, F4, J17, J9, K10, K11, K15, K16, L10, L16, R10, R16, T10, T11, T15, T16, U17, U9, Y4	1.5 V supply voltage for array	All
VCCDA	A12, A14, AA13, AA15, AA20, AA7, AB13, AC11, AD11, AD4, AE12, AE17, B15, C15, C6, D13, E13, E19, F21, G10, G5, N21, N5, W21	3.3 V supply voltage for I/O differential amplifier and JTAG and probe interfaces.	All
VCCIB0	A3, B3, C4, D5, J10, J11, K12	3.3 V supply voltage for I/O	All
VCCIB1	A23, B23, C22, D21, J15, J16, K14	3.3 V supply voltage for I/O	All
VCCIB2	C24, C25, D23, E22, K17, L17, M16	3.3 V supply voltage for I/O	All
VCCIB3	AA22, AB23, AC24, AC25, P16, R17, T17	3.3 V supply voltage for I/O	All
VCCIB3	AA22, AB23, AC24, AC25, P16, R17, T17	2.5 V supply voltage for I/O (TBD)	All
VCCIB4	AB21, AC22, AD23, AE23, T14, U15, U16	2.5 V supply voltage for I/O (TBD)	All
VCCIB5	AB5, AC4, AD3, AE3, T12, U10, U11	3.3 V supply voltage for I/O	All
VCCIB6	AA4, AB3, AC1, AC2, P10, R9,T9	3.3 V supply voltage for I/O	All
VCCIB7	C1, C2, D3, E4, K9, L9, M10	3.3 V supply voltage for I/O	All
VPUMP	E20	Voltage External Pump. In normal operation, using the internal charge pump, should be tied to ground.	All
TRST	E6	JTAG Test Clock. Actel recommends this pin be hardwired to ground for flight.	All
TCK	F5	JTAG Test Clock. Actel recommends this pin be hardwired to ground. Must not be left unconnected.	All
TDI	C5	JTAG Test Data Input. Actel recommends that this pin be hardwired to VCCDA, or left unconnected.	All
TDO	F6	JTAG Test Data Output. Must be left unconnected.	All
TMS	D6	JTAG Test Mode Select. Actel recommends this pin be hardwired to VCCDA, or left unconnected.	All
PRA	F13	Test Probe. The pins' probe capabilities are disabled to protect programmed design confidentiality. These pins must be left unconnected.	All
PRB	A13		All
PRC	AB12		All
PRD	AE13		All
NC	AA12, AA14, E12, E14, F12, F14, H12, H14, J12, J14, U12, U14, V12, V14, Y12, Y14	No Connection. Pins are not connected to circuitry within the device. These pins can be driven to any voltage or can be left floating with no effect on the operation of the device.	All
CLK*	AC12, AC13, AD12, AD13, W14, W15, W13, Y13	Global clocks. When pins are unused, Actel recommends they are tied to known state, preferably ground.	All
HCLK*	B13, B14, C12, C13, G12, G13, G14, G15	Hardwired clocks. When pins are unused, it is recommended that they are tied to ground.	All

14.5 RT3PE3000L specific pins - CG484 package

The Actel RT3PE3000L FPGA device has special pins that need to be correctly connected on the printed circuit board, as shown in table 81. Please refer to the Actel data sheet [RT3PE] for details.

Table 83. RT3PE3000L special pins - CG484 package

Name	Pin CG484	Note	CID
GND	See [RT3PE] for pin listings	Low supply voltage, ground	1
GNDQ		Low supply voltage, ground, quiet	1
VCC		1.2 V or 1.5 V Core supply voltage	1
VCCIBX		I/O Supply Voltage	1
VMVX		I/O Supply Voltage, quiet	1
VCCPLA		PLL Supply Voltage	1
VCCPLB		PLL Supply Voltage	1
VCCPLC		PLL Supply Voltage	1
VCCPLD		PLL Supply Voltage	1
VCCPLE		PLL Supply Voltage	1
VCCPLF		PLL Supply Voltage	1
VCOMPLA		PLL Ground	1
VCOMPLB		PLL Ground	1
VCOMPLC		PLL Ground	1
VCOMPLD		PLL Ground	1
VCOMPLE		PLL Ground	1
VCOMPLF		PLL Ground	1
VPUMP		Voltage External Pump	1
VREF		I/O Voltage Reference	1
GL*		Global	1
FF		Flash*Freeze Mode Activation Pin	1
TRST		JTAG Test Clock	1
TCK		JTAG Test Clock	1
TDI		JTAG Test Data Input	1
TDO		JTAG Test Data Output	1
TMS		JTAG Test Mode Select	1
NC		No Connect. Pins are not connected to circuitry within the device. These pins can be driven to any voltage or can be left floating with no effect on the operation of the device.	1
DC		Do Not Connect. This pin should not be connected to any signal on the PCB. These pins should be left unconnected.	1

15 Reference documents

[AMBA] AMBA^{TM} Specification, Rev 2.0, ARM IHI 0011A, 13 May 1999, Issue A, first release, ARM Limited

[GRLIB] GRLIB IP Library User's Manual, Aeroflex Gaisler, www.aeroflex.com/gaisler

[GRIP] GRLIB IP Core User's Manual, Aeroflex Gaisler, www.aeroflex.com/gaisler

[SPW] ECSS - Space Engineering, SpaceWire - Links, Nodes, Routers and Networks, ECSS-E-ST-50-12C, July 2008

[RMAPID] ECSS - Space Engineering, SpaceWire Protocols, ECSS-E-ST-50-51C, February 2010

[RMAP] ECSS - Space Engineering, SpaceWire Protocols, ECSS-E-ST-50-52C, February 2010

[RTAX] RTAX-S/SL RadTolerant FPGAs, 5172169-13/8.10, Revision 13, August 2010, Actel Corporation

[RT3PE] Radiation-Tolerant ProASIC3 Low Power Space-Flight Flash FPGAs with Flash*Freeze Technology, 51700107-1/11.09, Revision 1, November 2009, Actel Corporation

[PACK] Package Mechanical Drawings, 5193068-39/8.10, Revision 39, August 2010, Actel Corporation

[PCIF] CorePCIF Handbook, 50200087-1 /2.07, v2.1, February 2007, Actel Corporation

CorePCIF PCI Interface Core, 51700057-3/02.06, V 5.0, February 2006, Actel Corporation

CorePCIF User's Guide Actel IP Solutions, 50200051-1/3.06, March 2006, Actel Corporation

CorePCIF v2.03 Release Notes, 51300025-2/2.06, February 2006, Actel Corporation

16 Ordering information

Ordering information is provided in table 84 and a legend is provided in table 85.

Table 84. Ordering information

Product	CID	Device	SpaceWire Interface		Speed Grade		Package Type	Lead Count	Screening Level		
			LVTTL	LVDS	STD	-1			EV	E	B
RT-SPW-ROUTER-10X-RTAX	1	RTAX2000S/SL	LVTTL	LVDS		-1	CQ	352	EV	E	B
RT-SPW-ROUTER-10X-RTAX	1	RTAX2000S/SL	LVTTL	LVDS		-1	CGS	624	EV	E	B
RT-SPW-ROUTER-10X-RT3PE	1	RT3PE3000L	LVTTL	LVDS		-1	CG	484			B
RT-SPW-ROUTER-6X-RTAX	2	RTAX2000S/SL	LVTTL	LVDS		-1	CQ	352	EV	E	B
RT-SPW-ROUTER-6X-RTAX	2	RTAX2000S/SL	LVTTL	LVDS		-1	CGS	624	EV	E	B

Table 85. Ordering legend

Designator	Option	Description
Product	RT-SPW-ROUTER-10X	Radiation-Tolerant 10x SpaceWire Router
	RT-SPW-ROUTER-6X	Radiation-Tolerant 6x SpaceWire Router with PCI Interface
Device	RTAX2000S	2,000,000 Equivalent System Gates
	RTAX2000SL	2,000,000 Equivalent System Gates - Low-Power Option
	RT3PE3000L	3,000,000 Equivalent System Gates - Low-Power Option
Speed Grade	STD	Standard Speed
	-1	Approximately 15% Faster than Standard
Package Type	CQ	Ceramic Quad Flat Pack
	CG	Ceramic Column Grid Array (RTAX) (obsolete)
	CG	Ceramic Column Grid Array (1.0 mm pitch), Six Sigma Column (RT3PE)
	CGS	Ceramic Column Grid Array, Six Sigma Column (RTAX)
	LG	Land Grid Array
Screening Level	B	MIL-STD-883 Class B
	E	Actel Extended Flow (Actel Space-Level Flow)
	EV	Class V Equivalent Flow Processing Consistent with MIL-PRF 38535
	PROTO	Prototype Unit; not for Space Flight or Qualification of Space-Flight Hardware

Commercial quality components can be supplied for prototyping.

For Actel AX2000, the footprint corresponds to either a CQ352 or a CG624 package, depending on the configuration identifier (CID) as listed in table 84 above. The prototyping component is supplied in a FG896 package and should be used with an FG896-to-CQ352 or an FG896-to-CG624 adapter, respectively. This allows the use of the same footprint on prototyping and flight boards. The pinout of the prototyping component (i.e. the FG896 package) can be provided upon request. Prototyping components also can be supplied in CQ352 and CG624 packages.

For Actel A3PE3000L, the footprint corresponds to the FG484 package.

17 Change record

Change record information is provided in table 86.

Table 86. Change record

Issue	Date	Sections	Note
1.2	2012 June	3.7	Clarified how timeout period is related to register value
1.1	2011 June	2.3	Router configuration area memory map corrected
		3.2.6	Port disable clarified
		3.2.8	On-chip memory handling clarified
		3.2.8.1	Auto-scrub clarified
		3.2.10	Invalid address error clarified
		3.2.11	Global configuration clarified
		3.9	Timining adjusted
		3.7	Register bits added
		14.2	Pin numbers and drive capability updated
1.0	2010 October	All	New document

Aeroflex Gaisler AB
Kungsgatan 12
411 19 Göteborg
Sweden

tel +46 31 7758650
fax +46 31 421407
sales@gaisler.com
www.aeroflex.com/gaisler