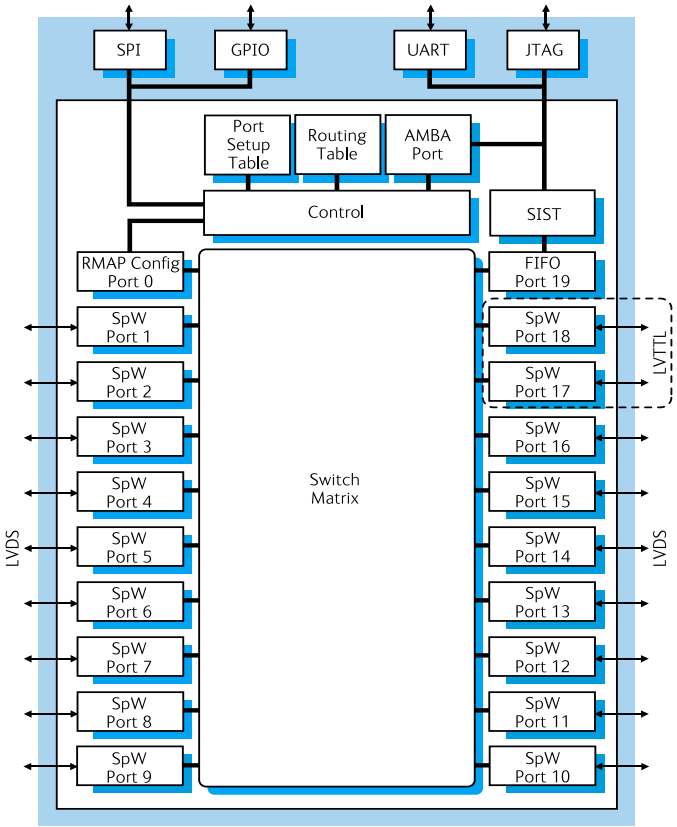
Data Sheet and User's Manual

Radiation-Tolerant 18x SpaceWire Router
GR718B

Features

- Fault tolerant 18x SpaceWire Router compatible with ECSS-E-ST-50-12C
- Supports ECSS-E-ST-50-12C Rev 1 features
- Non-blocking switch-matrix connecting any input port to any output port
- Packet Distribution, Group Adaptive Routing
- Path, Logical and Regional Logical addressing
- Two priority levels for output port arbitration
- 16x SpaceWire ports with on-chip LVDS
- 2x SpaceWire ports with LVTTL
- Configuration port using RMAP, compliant with ECSS-E-ST-50-52C
- Support functions for SpaceWire-D
- Distributed Interrupt support
- System-time distribution via all ports
- SpaceWire Plug-and-Play support
- Timers on all ports to recover from deadlock
- UART and JTAG interfaces to configuration port
- GPIO and SPI interfaces
- Do not fully support failsafe operation of LVDS as defined by ECSS-E-ST-50-12C

- Does not support cold sparing

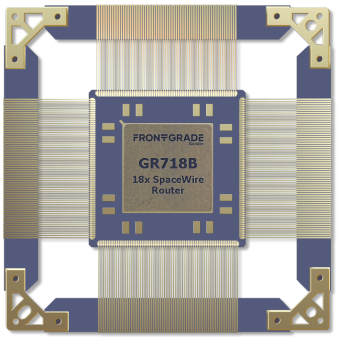


Description

The Radiation-Tolerant 18x SpaceWire Router device is ideally suited for space and other high-rel applications, providing a high-throughput worm-hole router functionality.

Specification

- CQFP256 hermetically sealed ceramic package
- Total Ionizing Dose (TID) up to 300 krad (Si)
- Single-Event Latch-Up Immunity (SEL) to LET_{TH} > 118 MeV-cm^2 mg
- Single-Event Upset (SEU) below 10^{-12} bit error rate in space environment
- 1.8 V and 3.3 V supply
- 2.8 W consumption (typical)
- Up to 200 Mbps on all 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 eighteen external routing ports, an internal configuration port, and an internal port for system level test support.

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.

Frontgrade Gaisler AB
Kungsgatan | SE-411 19 | Goteborg | Sweden
+46 31 7758650 | frontgrade.com/gaisler

Table of contents

1 Introduction

1.1 Scope

This document is the data sheet and user's manual for revision B of the GR718 - Radiation Tolerant 18x SpaceWire Router. The GR718 was developed in an activity initiated by the European Space Agency under ESTEC contract 4000105402/12/NL/CBi.

1.2 Reference documents

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

[SPW] ECSS - Space Engineering, SpaceWire - Links, nodes, routers and networks, ECSS-E-ST-50-12C, July 2008

[SPWID] ECSS - Space Engineering, SpaceWire protocol identification, ECSS-E-ST-50-51C, February 2010

[RMAP] ECSS - Space Engineering, SpaceWire - Remote memory access protocol, ECSS-E-ST-50-52C, February 2010

[SPWPNP] ECSS - Space Engineering, SpaceWire Plug-and-Play protocol, ECSS-E-ST-50-54C Draft, March 2013

[SPWD] SpaceWire-D - Deterministic Control and Data Delivery over SpaceWire Networks, Draft B, April 2010, ESA Contract Number 220774-07-NL/LvH

[SPWINT] Yuriy Sheynin, Distributed Interrupts in SpaceWire Interconnections, International SpaceWire Conference, Nara, November 2008 (outdated)

1.3 Document revision history

Change record information is provided in table 1.

Table 1. Change record

Version	Date	Sections	Note
1.5	2013 September	All	Document updated after tape-out. All sections affected.
1.6	2013 October	All	Created new section 1, and moved Reference documents and Document revision history to section 1. Old section 1 was split into section 2, 3, 4, 5. Moved section 2.8 to section 6.2.20 Moved section 2.9 to section 6.2.22 Changed name of SPW_CLK pin to SPWCLK. Added complete register overview with acronyms for all registers. Changed name of section General Purpose Register to System Level Test Configuration. Moved description of loop-back to this section. Updated block diagram (figure 1). Added acronyms for UART, SPI, GPIO, and GPREG registers to respective section's register table, as well as both AMBA and RMAP address. Clarified the memory map and AMBA plug & play Added reset value column to I/O pins (table 12). Changed complete GPIO pin multiplexing section to make it clearer. Rewrote Interrupt distribution subsection (in SpaceWire router section). Now named Distributed interrupt support, and includes more details. Added SpaceWire routing introduction to section 1.5. Added timing diagram for LOCK pin to AC characteristics section, as well as various fixes to other timing diagrams in order to make them more clear. SPW_RXSn and SPW_TXSn were mapped to the wrong pins in the pin list. Changed so that all registers use the same layout. Added register layout example to the Register overview section. Fixed various spelling errors throughout the document. Minor clarifications throughout the document.

Table 1. Change record

Version	Date	Sections	Note
1.7	2014 February	All	Changed affix for package from -CG256 to -CQ256. Register description for Port status register (table 28) showed wrong bit index for PB and PR fields. Also, description for RE bit was wrong. SpW PnP address range in table 6 was defined as 0x0000-0x8001. Changed to 0x00000000-0x0000C000. Changed RMAP addresses in register tables from 0xNNNN to 0xNNNNNNNN format. Moved static routing section to 6.2.6 Changed layout of overview table for SpaceWire Plug-and-Play registers. Replaced “packet timer” with “data character timer” in order to have a name that better describes the use of the timer. Clarified how link-rate is set by adding a formula. Clarified how the RMAP target handles write commands larger than four bytes (i.e. to more than one register) Clarified group adaptive routing and packet distribution. Fixed error claiming that RMAP commands denied access due to the access control features described in 6.5.1.3 were “silently discarded”. Moved section about packet length truncation from subsection of SpaceWire-D to its own section, because it’s not really part of SpaceWire-D. Added more details about AMBA plug & play in order to remove reference to GRLIB. Changed layout for description of SIST register fields to match all other registers. Clarified timing diagram for AUXTICKIN and AUXTICK-OUT signals. Updated electrical parameters. Updated pin placement. Fixed various spelling errors throughout the document. Minor clarifications throughout the document.

Table 1. Change record

Version	Date	Sections	Note
1.8	2015 March	2.4, 6.5.3	Wrong start address for SpaceWire router's Credit counter registers (RTR.CREDCNT)
		6.5.3, 6.2.16	Register bit RTR.RTRCFG.IP (bit 11) had wrong acronym in table overview (SD instead of IP). Changed acronym for bits 9 and 10 in RTR.RTRCFG register from TA and IA to AT and AI respectively, due to conflict with other acronyms.
		2.4	Wrong start address for SpaceWire router's port control registers (RTR.PCTRL)
		7.3	Wrong name for register in table 92 (SIST.CTRL register)
		6.3.2	Clarified that link is allowed to enter the started state if link-start-on-request is trying to start the port.
		6.5.3, 6.2.10	Changed acronym for bit 18 in RTR.PCTRL register from LS to LR due to conflict with other acronym.
		15.1	Added note about common ground
		15.3	Added Core Standby Current
		15.4, 15.5	Corrected unit for parameters.
		15.6.6, 15.6.7	Updated data and strobe input skew, jitter and hold
		16.2	Merged ground and supply pin groups
2.0	2015 June	Throughout doc (no change bars)	Changed wording to conform with upcoming revision 1 of the SpaceWire standard: interrupt-code changed to interrupt code; interrupt-acknowledge-code changed to interrupt acknowledgement code; distributed interrupt code used when talking about either an interrupt code or an interrupt acknowledgement code.
		5 (table 14 to 20), 6.2.16, 6.5.3 (table 32), 15.6.11	Added possibility to have auxiliary time-code / distributed interrupt code interface inputs to be asynchronous to CLK.
		5 (table 20)	Changed time-code / distributed interrupt code reset configuration.
		6.5.3 (table 34), 6.5.4 (table 66)	Changed minor version number to 3.
		6.5.3 (table 46)	Changed ISR change timers to 10 bits.
		6.5.3 (table 23)	Changed description of routing table port map registers, since they now read zero after reset for logical addresses.
		2.4 (table 8), 5 (table 20), 6.2.15, 6.5.3 (tables 22, 31, 32, 41, 42, 43, 44)	Added support for distributed interrupts to operate in extended interrupt mode, i.e with 64 interrupt codes instead of 32 interrupt codes + 32 interrupt acknowledgement codes.
		6.5.3 (table 41)	Fixed error in description of RTR.ICODEGEN.UA bit. Functionality does not depend on RTR.ICODEGEN.AH bit.
		Throughout doc	Renamed device to GR718B

Table 1. Change record

Version	Date	Sections	Note
2.1	2015 October	16.2 (Table 149)	Corrected the direction for LVDS reference pins
		16.2 (Table 149)	Modified recommendation for decoupling of analog PLL supply
		15.2 (Table 132)	Note for recommended operating conditions for V_{DDIO}
		15.1 (Table 131)	Absolute maximum values for maximum voltage
		15.1 (Table 131)	Absolute maximum values for Analog PLL supply
		15.2 (Table 132)	Recommended values for Analog PLL supply
		15.3 (Table 133)	PLL supply current
		15.6.5 (Table 142), 15.6.8 (Table 145), 15.6.9 (Table 146), 15.6.11 (Table 148)	Updated timing parameters for JTAG, SPI, GPIO and AUX
		16.2 (Table 149)	Pin description and direction of LVDSREF[0] and LVDSREF[2] are updated.
2.2	2015 November	All	Updated page layout, no changes to the contents.
2.3	2016 April	15.1, 15.2, 15.3, 15.4, 15.6, 16, 16.3	Updated absolute maximum rating and recommended conditions. Updated power supply condition and ratings Updated input/output DC characteristics Added equivalent load model for AC test Updated AC parameters Changed names on LVDSREF[0] pin to SELVREF and LVDSREF[2] to VREFEXT, LVDSREF[1] to LVDSREF Updated tolerance on resistance for external LVDS reference Added mechanical drawing of package
2.4	2016 November	2.1, 3 (Table 12), 15.1 (Table 133), 15.2 (Table 134), 15.4 (Table 137), 15.5 (Table 139), 15.6, 16.2 (Table 151)	Clarification of LVDS properties. Added LVDS input and output simplified buffer schematics.
2.5	2017 January	16.2 (Table 151)	Clarification of tolerance for VREFEXT

Table 1. Change record

Version	Date	Sections	Note
3.0	2017 October	Front	Changed status of document to data sheet after qualification.
		Front	Clarified non-support for coldspare and failsafe and updated typical power consumption
		1.4 (Table 2)	Added missing acronyms
		3 (Table 12)	Clarified non-support for coldspare and failsafe
		15.1 (Table 133)	Updated absolute maximum rating
		15.2 (Table 134)	Updated recommended operating conditions
		15.3 (Table 135)	Updated maximum power consumption
		15.4 (Table 135)	Clarified High level input voltage for system reset
		15.7	Clarified load conditions for AC parameters
		16.2 (Table 151)	Clarified non-support for coldspare and failsafe
		16.3	Corrected package lid dimensions
3.1	2018 February	15.1 (Table 133)	Updated absolute maximum rating
		15.2 (Table 134)	Updated recommended operating conditions
		15.4	Updated input capacitance value
		15.5	Updated output capacitance value
		15.7.3	Updated PLL lock time
		Table 20	Minor corrections
		Table 18	Register affected by external GPIO are clarified
		Table 16	Register affected by external GPIO are clarified
3.2	2018 March	1.6	Added Section for errata
		1.6.1	Added errata about missing first byte due to EEP insertion in a character overrun situation
3.3	2018 July	1.6.2	Added errata about blocking output ports used for distributing multicast packets
3.4	2020 April	Table 20	Added missing RTR.RTRCFG.AI reset values
		Table 33	Clarified reset behavior of RTR.RTRCFG.AI
		Table 47	Corrected bit width and default value for RTR.ISRC-TIMER.RL
		Table 27	Corrected input signal for setting reset value of the clock divisor. Added bit description of Run-state clock divisor (RD) and Initialization clock divisor (ID)
		Table 36	
		6.2.20	Added reference reset values used in formulas
		Table 12	Added to reference to register initialized by SPWCLKDIV
3.5	2020 June	Table 44	Corrected address for RTR.ISR1 register
3.6	2020 August	Table 27	Corrected description for the RTR.PCTRL.PR bit. Bit is double mapped in RTR.RTACTRL.PR. Priority is explained in the description for the RTR.RTACTRL.PR bit.
3.7	2020 November	Table 152	Corrected names in the package dimension table.
3.8	2021 September	Table 133	Clarified absolute maximum current for VREFEXT pins
		Table 136	Corrected VREFEXT pin leakage description
		6.5.1.1	Defined implicit return address feature implemented.

Table 1. Change record

Version	Date	Sections	Note
3.9	2024 October	All	Updated document template
		Front Page	Updated photo of GR718B device
		1.6.3	Added errata about LVDS driver enabling too late leading to connection failure in AutoStart mode.
		2.3.1	Added the values of device ID and build ID that can be found in the Plug-and-Play area.
		6.5.4	Corrected which bit that shows if SpaceWire Plug-and-Play is enabled.
		6.5.4	Updated company name in Vendor ID and Product ID.
		9.1, 9.2.1	Clarified JTAG instruction register usage for connecting to the GR718B.
		16.2	Updated description for SPWEN[17] and SPWEN[18] to clarify the usage.
		16.3	Corrected clerical errors in package drawing; Measure A2 max corrected from 0.25 to 0.55 mm.

1.4 Acronyms

Table 2. Acronyms

Acronym	Comment
AHB	Advanced High-performance bus, part of [AMBA]
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus, part of [AMBA]
FIFO	First-In-First-Out, refers to buffer type
GPIO	General purpose input/output
I/O	Input/Output
JTAG	Joint Test Action Group (developer of IEEE Standard 1149.1-1990)
kB	Kilobyte, 10^3 bytes
KiB	Kibibyte, 2^{10} bytes, unit defined in IEEE 1541-2002
LSb	Least significant bit
LSB	Least significant byte
Mb, Mbit	Megabit, 10^6 bits
MB, Mbyte	Megabyte, 10^6 bytes
MiB	Mebibyte, 2^{20} bytes, unit defined in IEEE 1541-2002
MSb	Most significant bit
MSB	Most significant byte
PnP	Plug-and-Play
RMAP	Remote Memory Access Protocol
SEE	Single Event Effects
SEL/SEU/ SET	Single Event Latch up/Upset/Transient
SoC	System-on-Chip
SPI	Serial Peripheral Interface
TID	Total Ionizing Dose
UART	Universal Asynchronous Receiver/Transmitter

1.5 SpaceWire routing introduction

This section provides a short introduction to SpaceWire routing, and introduces the terminology used throughout this document. For more details regarding SpaceWire, please refer to ECSS-E-ST-50-12C [SPW]. Note that GR718B provides routing features not described in this section. For more details regarding these features, please see section 6.

When a SpaceWire packet arrives on an input port of a router the first byte of the packet is used as an address to determine which port the packet should be routed to. If the first byte has a value of 0x01-0x1F it is said to be a physical address (or path address), meaning that it corresponds to an actual physical port of a router. If the first byte has a value of 0x20 - 0xFF it is said to be a logical address. If the first byte has value 0x00, the packet is always routed to the router's internal configuration port.

A packet may contain more than one address byte, but each router only looks at the first address byte when determining how to route the packet. The router may or may not, depending on how it is configured, remove the first address byte before forwarding the packet. This is called header deletion. For physical addresses, header deletion is mandatory, which means that each router that the packet passes through will use a different address byte when routing the packet. The physical addresses thus creates a path for the packet through the network. For logical addresses, header deletion is optional, allowing both logical addresses to be used across the whole network, and multiple level logical addressing.

The information on how to route packets with a specific address is stored in the routing table. The routing table in GR718B is a look-up table with 255 entries, one for each physical and logical address. Each entry in the routing table contains information about which port(s) the packet should be routed to, information whether or not header deletion should be used, as well as some other features described in more detail in section 6.2.1.

Once the route for the packet has been determined, the router's switch matrix connects the input port (the port that the packet was received on) to the output port (the port that the packet should be transmitted on). This connection is called a wormhole. When the packet transfer is finished, the wormhole is closed.

The switch matrix in GR718B can create wormholes between any ports. A port can only be an input for one wormhole at a time, and only be an output for one wormhole at a time. However, a port can be both an input and an output simultaneously, and the port's two wormholes can be connected to different ports. For example, a wormhole can be created where port 1 is an input and port 2 is an output. At the same time another wormhole can be created where port 3 is an input and port 1 is an output. Note that wormholes are not created between the actual SpaceWire link interfaces, they are created only through the switch matrix. This means that the link interfaces on the different sides of a wormhole can operate at different link speeds. The interface between the link interfaces and the switch matrix in GR718B consists of transmit and receive FIFOs with 64 entries each. The wormhole stays open either until the last byte of the packet has been read from the input port's receive FIFO and written to the output port's transmit FIFO, or until the router decides that the packet should be spilled (see section 6 for details).

1.6 Errata

1.6.1 Missing first byte after EEP on links with downstream congestion

Input port overrun protection can cause the insertion of an Error End of Packet (EEP) to overwrite the first byte of the next packet, when transfer is resumed. For more information about overrun protection see 6.2.12.

Missing first byte after EEP on link with downstream congestion occurs when all of the following conditions are met:

- Input port is configured to use overrun time-out protection. See 6.2.12 for more information.
- Flow control causes output buffer to be entirely filled.
- Overrun time-out has occurred in the input buffer, the current packet is discarded in the input buffer, EEP is scheduled for insertion in the output buffer.
- Output port is resuming normal operation after receiving flow control token, i.e. the downstream receiver is able to accept data.
- After the output port resumes normal operation, the following packet is routed from the same input port as the packet that was discarded and terminated with EEP.

Workaround 1: Disable overrun protection, and if possible move the overrun protection to the end node.

1.6.2 Blocking output ports used for distributing multicast packets

When a subset of the output ports are busy sending a packet, an incoming multicast packet will allocate the available outputs ports and then wait for all output ports to become available. If another multicast packet with higher priority (lower input port number) that will be routed to the same set of ports is sent to the router, this packet will allocate the currently occupied output ports when they become available. This results in a blocking state where multiple multicast packets, each allocating a subset of the output ports, blocks each other.

Triggering a blocking state occurs when all of the following conditions are met:

- Sending multicast packets distributed to multiple output ports from multiple input ports while also sending packets to only a subset of the same output ports.
- Multicast packet with higher priority (lower input port number) is received at the router when multicast packet with lower priority only has successfully allocated a subset of the output ports.

Workaround 1: All packets routed to output ports within a packet distribution group should use the same routing configuration and packet distribution mapping.

1.6.3 LVDS driver enabled too late when in AutoStart mode

The first NULL transmitted by a GR718B port after reset, or when the port was previously disabled, may be corrupted since the LVDS driver is not guaranteed to be enabled during the first transmitted bit. Any subsequent link start attempt will be correct.

When a SpaceWire port is configured in AutoStart mode and in power-down mode, this results in link connection failing on the first attempt.

The cause of this is that the SpaceWire port LVDS drivers are enabled too late when the SpaceWire port leaves Ready state after receiving a NULL character, leading to the transmitted NULL character being corrupted. In AutoStart mode, only a single NULL character is transmitted, since the gotNULL condition is already true. The port will then move on to transmitting FCTs, leading to gotFCT before gotNULL in the device at the other end of the SpaceWire link, which in turn leads to it returning to ErrorReset.

If the link connection is retried a second time, the LVDS driver will still be enabled and the leading NULL character will be correctly transmitted, leading to the link connecting as expected.

Note that if the device on the other end of the SpaceWire link keeps LinkStart enabled, an automatic reconnection attempt will be made after 19.2µs (+-10%).

Power-down mode is entered in the following cases:

- when the router is reset (this includes in response to writes to RTR.RTRCFG.RE),
- when the port enters the ErrorReset state at the same time as RTR.PCTRL.LD=1 (LinkDisabled=1).

For further information about the power-down mode, refer to section 6.2.22.

2 Architecture

2.1 Overview

The GR718B - Radiation Tolerant 18x SpaceWire Router architecture is centered around 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 addressing are supported. Group adaptive routing is fully supported, meaning that both path and logical addresses can be individually configured to use one or more output ports. A unique feature is the support for packet distribution, which can be used to implement multicast and broadcast addressing. Output ports are arbitrated using two priority levels with a round-robin scheme within each level.

The 18x SpaceWire router implements 18 external SpaceWire ports, an internal configuration port, that gives access to configuration and status registers, and also an internal port for system level testing. Out of the external SpaceWire ports, 16 have on-chip non-cold-spare LVDS transceivers, and two have LVTTL interfaces to off-chip LVDS transceivers. For noisy environment or protection from various failures such as open inputs, floating inputs or shorted inputs an external fail-safe function should be considered for all active SpaceWire LVDS inputs.

There is also an on-chip AMBA Advanced High-speed Bus (AHB) which hosts UART and JTAG interfaces that are AHB masters. The router's configuration port contains an AHB slave interface which is connected to the AHB bus, and allows the configuration port registers to be accessed from the UART and JTAG interfaces.

An auxiliary time-code / distributed interrupt code interface is present, for sending and receiving time-codes / distributed interrupt codes through external pins. Parts of the interface use dedicated pins, while the rest are multiplexed on the GPIO[23:0] pins.

There are SPI and GPIO interfaces accessible through the configuration port, which allows SPI devices to be accessed, and general purpose signalling to be performed, directly through RMAP commands, or through the UART and JTAG interfaces.

The full SpaceWire router architecture includes the following modules: SpaceWire Router, SPI Controller, UART Interface, JTAG Interface, General Purpose I/O Interface, SpaceWire In-System Test (SIST), System Level Test Configuration, AMBA AHB controller and AMBA APB controller.

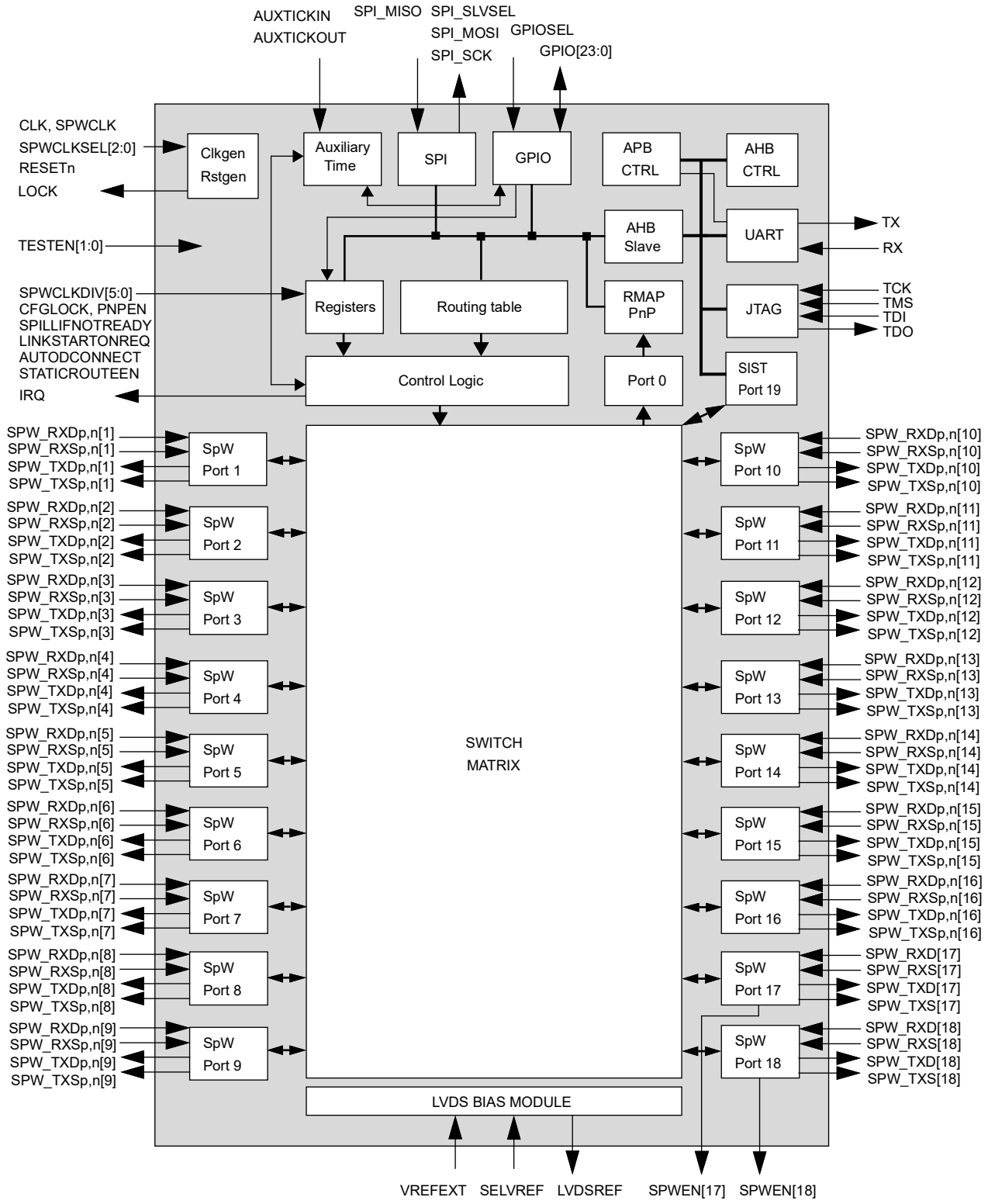


Figure 1. Block diagram and signal overview

2.2 Cores

GR718B is based on cores from the GRLIB IP library. The cores used are listed in table 3. The vendor and device identifiers shown in the last two columns can be extracted from the AMBA plug & play information.

Table 3. Used IP cores

Core	Function	Vendor	Device
AHBCTRL	AHB Arbiter & Decoder	0x01	-
APBCTRL	AHB/APB Bridge	0x01	0x006
GRSPWROUTER	SpaceWire Router	0x01	0x08B
GRGPIO	General Purpose I/O Interface	0x01	0x01A
AHBUART	UART Interface	0x01	0x007
AHBJTAG	JTAG Interface	0x01	0x01C
SPICTRL	SPI Controller	0x01	0x02D
GRSPW2_SIST	SpaceWire In-System Test	0x01	0x091
GRGPREG	System Level Test Configuration	0x01	0x087

2.3 Memory map

GR718B provides three different address spaces: AMBA address space, RMAP address space, and SpaceWire Plug-and-Play address space. The AMBA address space is accessible through the JTAG and UART interfaces. The RMAP address space is accessible through RMAP commands to the router's configuration port. The SpaceWire Plug-and-Play address space is accessible through SpaceWire Plug-and-Play commands to the router's configuration port. Table 4, 5, and 6 shows an overview of the three different address spaces, while section 2.4 contains a complete list of all the GR718B registers, showing the addresses for all three address spaces. The AMBA plug & play information is detailed in section 2.3.1.

Table 4. AMBA AHB address space

Core	Address range	Comment
APBCTRL	0xFFE00000 - 0xFFEFFFFC	AMBA APB bridge
AHBUART	0xFFE00000 - 0xFFE000FC	UART interface registers
GRGPREG	0xFFE00100 - 0xFFE001FC	System Level Test Configuration registers
GRSPW2_SIST	0xFFE00200 - 0xFFE002FC	SpaceWire In-System Test registers
APBCTRL	0xFFEFF000 - 0xFFE0FFFF	AMBA APB plug & play area
GRSPWROUTER	0xFFF20000 - 0xFFF21FFC	SpaceWire Router registers
SPICTRL	0xFFF22000 - 0xFFF220FC	SPI controller registers
GRPGIO	0xFFF22100 - 0xFFF221FC	General purpose I/O interface registers
AHBCTRL	0xFFFFF000 - 0xFFFFFFFF	AMBA AHB plug & play area

Note: An access to an undefined address will cause an AHB error response. A read access to a reserved address within a defined range will return zero, while a write access will have no effect.

Table 5. RMAP address space

Core	Address range	Comment
GRSPWROUTER	0x00000000 - 0x00001FFC	SpaceWire Router registers
SPICTRL	0x00002000 - 0x000020FC	SPI controller registers
GRPGIO	0x00002100 - 0x000021FC	General purpose I/O interface registers
N/A	0x00002200 - 0x00002FFC	Reserved range. Read return zero. Write has no effect.

Note: An RMAP access to an undefined address will result in an error. See section 6.5.1 for details.

Table 6. SpaceWire Plug-and-Play address space

Core	Address range	Comment
GRSPWROUTER	0x00000000 - 0x0000C000	SpaceWire Plug-and-Play registers

Note: See section 6.5.4 for details about accesses to undefined addresses.

2.3.1 AMBA plug & play information

Plug & play is used on the AMBA bus to identify all cores and their configurations. The AMBA plug & play memory map is shown in table 7, and the two different plug & play record types are detailed in figure 2 and 3. Both the AHB and APB plug & play information are mapped at read-only address areas. Note that AMBA plug & play is not available for the SPI controller or General purpose I/O interface, mapped in the SpaceWire router configuration port area.

Also note that AMBA plug & play is not to be confused with the SpaceWire-Plug-and-Play, which is used for identifying the SpaceWire characteristics through the configuration port of the router. SpaceWire Plug-and-Play is described in section 6.5.4.

Address 0xFFFFFFF0 contains the value 0x0718101a, of which the 16 most significant bits are the GR718B device identifier (0x718) and the 16 least significant bits are the GRLIB build ID (4122).

Table 7. AMBA plug & play memory map

Core	Function	Type	Index	Address range
AHBJTAG	JTAG interface	AHB master	0	0xFFFFF000 - 0xFFFFF01F
AHBUART	UART interface	AHB master	1	0xFFFFF020 - 0xFFFFF03F
		APB slave	0	0xFFEFF000 - 0xFFEFF007
APBCTRL	AHB/APB Bridge	AHB slave	0	0xFFFFF800 - 0xFFFFF81F
GRSPWROUTER	SpaceWire Router	AHB slave	1	0xFFFFF820 - 0xFFFFF83F
GRGPREG	System Level Test Configuration	APB slave	1	0xFFEFF008 - 0xFFEFF00F
GRSPW2_SIST	SpaceWire In-System Test	APB slave	2	0xFFEFF010 - 0xFFEFF017

Figure 2. AHB plug & play information record

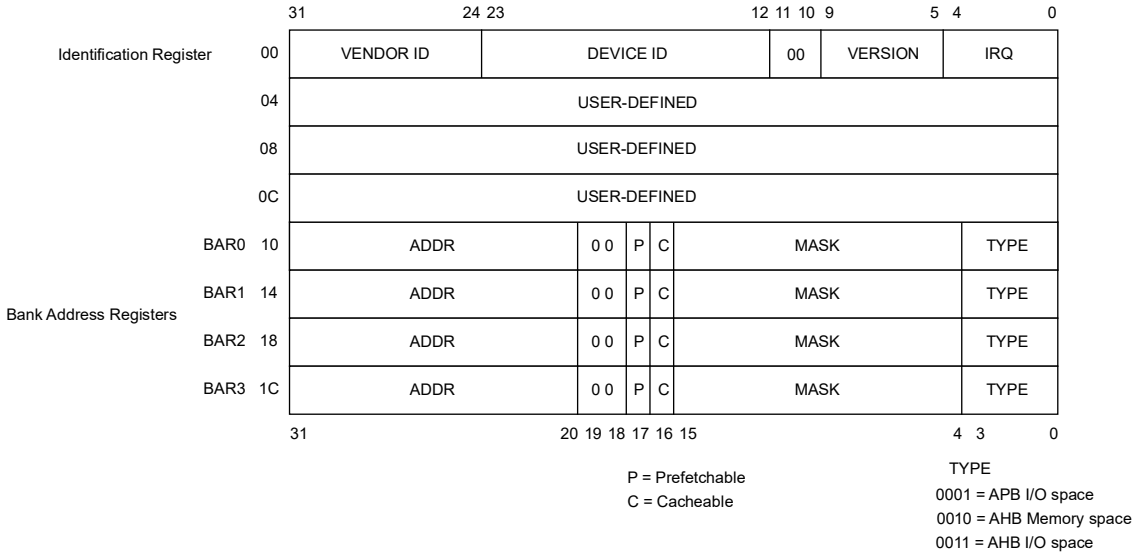
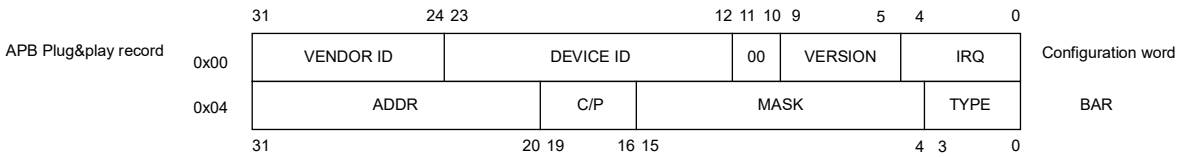


Figure 3. APB plug & play information record



2.4 Register overview

Table 8 contains an overview of all GR718B registers and shows which AMBA address, RMAP address, and SpaceWire Plug-and-Play address they are mapped to. Detailed descriptions of the registers are found in the section where the corresponding function is described. Note that unlisted addresses are reserved.

Table 8. GR718B register overview

Register name	Register acronym	AMBA address	RMAP address	SpW PnP address
SpaceWire Router registers (section 6.5.3)				
Routing table port mapping, physical addresses 1-19	RTR.RTPMAP	0xFFF20004 - 0xFFF2004C	0x00000004 - 0x0000004C	
Routing table port mapping, logical addresses 32-255	RTR.RTPMAP	0xFFF20080 - 0xFFF203FC	0x00000080 - 0x000003FC	
Routing table address control, physical addresses 1-19	RTR.RTACTRL	0xFFF20404 - 0xFFF2044C	0x00000404 - 0x0000044C	
Routing table address control, logical addresses 32-255	RTR.RTACTRL	0xFFF20480 - 0xFFF207FC	0x00000480 - 0x000007FC	
Port control, port 0 (configuration port)	RTR.PCTRLCFG	0xFFF20800	0x00000800	
Port control, ports 1-19 (SpaceWire ports and SIST port)	RTR.PCTRL	0xFFF20804 - 0xFFF2084C	0x00000804 - 0x0000084C	
Port status, port 0 (configuration port)	RTR.PSTSCFG	0xFFF20880	0x00000880	
Port status, ports 1-19 (SpaceWire ports and SIST port)	RTR.PSTS	0xFFF20884 - 0xFFF208CC	0x00000884 - 0x000008CC	
Port timer reload, ports 0-19	RTR.PTIMER	0xFFF20900 - 0xFFF2094C	0x00000900 - 0x0000094C	
Port control 2, port 0 (configuration port)	RTR.PCTRL2CFG	0xFFF20980	0x00000980	
Port control 2, ports 1-19 (SpaceWire ports and SIST port)	RTR.PCTRL2	0xFFF20984 - 0xFFF209CC	0x00000984 - 0x000009CC	
Router configuration / status	RTR.RTRCFG	0xFFF20A00	0x00000A00	
Time-code	RTR.TC	0xFFF20A04	0x00000A04	
Version / instance ID	RTR.VER	0xFFF20A08	0x00000A08	
Initialization divisor	RTR.IDIV	0xFFF20A0C	0x00000A0C	
Configuration write enable	RTR.CFGWE	0xFFF20A10	0x00000A10	
Timer prescaler reload	RTR.PRESCALER	0xFFF20A14	0x00000A14	
Interrupt mask	RTR.IMASK	0xFFF20A18	0x00000A18	
Interrupt port mask	RTR.IPMASK	0xFFF20A1C	0x00000A1C	
Port interrupt pending	RTR.PIP	0xFFF20A20	0x00000A20	
Interrupt code generation	RTR.ICODEGEN	0xFFF20A24	0x00000A24	
Interrupt code distribution ISR, interrupt 0-31	RTR.ISR0	0xFFF20A28	0x00000A28	
Interrupt code distribution ISR, interrupt 32-63	RTR.ISR1	0xFFF20A2C	0x00000A2C	
Interrupt code distribution ISR timer reload	RTR.ISRTIMER	0xFFF20A30	0x00000A30	
Interrupt code distribution ACK-to-INT timer reload	RTR.AITIMER	0xFFF20A34	0x00000A34	
Interrupt code distribution ISR change timer reload	RTR.ISRCTIMER	0xFFF20A38	0x00000A38	
SpaceWire link running status	RTR.LRUNSTS	0xFFF20A40	0x00000A40	

Table 8. GR718B register overview

Register name	Register acronym	AMBA address	RMAP address	SpW PnP address
Capability	RTR.CAP	0xFFF20A44	0x00000A44	
SpaceWire Plug-and-Play - Device Vendor and Product ID	RTR.PNPVEND	0xFFF20A50	0x00000A50	0x00000000
SpaceWire Plug-and-Play - Unit Vendor and Product ID	RTR.PNPUVEND	0xFFF20A54	0x00000A54	0x00000009
SpaceWire Plug-and-Play - Unit Serial Number	RTR.PNPUSN	0xFFF20A58	0x00000A58	0x0000000A
Outgoing character counter, ports 1-19	RTR.OCHARCNT	0xFFF20C10, 0xFFF20C20 ... 0xFFF20D30	0x00000C10, 0x00000C20 ... 0x00000D30	
Incoming character counter, ports 1-19	RTR.ICHARCNT	0xFFF20C14, 0xFFF20C24 ... 0xFFF20D34	0x00000C14, 0x00000C24 ... 0x00000D34	
Outgoing packet counter, ports 1-19	RTR.OPKTCNT	0xFFF20C18, 0xFFF20C28 ... 0xFFF20D38	0x00000C18, 0x00000C28 ... 0x00000D38	
Incoming packet counter, ports 1-19	RTR.IPKTCNT	0xFFF20C1C, 0xFFF20C2C ... 0xFFF20D3C	0x00000C1C, 0x00000C2C ... 0x00000D3C	
Maximum packet length, ports 0-19	RTR.MAXPLEN	0xFFF20E00 - 0xFFF20E4C	0x00000E00 - 0x00000E4C	
Credit counter, ports 1-18	RTR.CREDCNT	0xFFF20E84 - 0xFFF20EC8	0x00000E84 - 0x00000EC8	
General purpose out, bits 0-31	RTR.GPOA	0xFFF20F00	0x00000F00	
General purpose out, bits 32-48	RTR.GPOB	0xFFF20F04	0x00000F04	
General purpose in, bits 0-1	RTR.GPIA	0xFFF20F10	0x00000F10	
Routing table, combined port mapping and address control, addresses 1-19	RTR.RTCOMB	0xFFF21004 - 0xFFF2104C	0x00001004 - 0x0000104C	
Routing table, combined port mapping and address control, addresses 32-255	RTR.RTCOMB	0xFFF21080 - 0xFFF213FC	0x00001080 - 0x000013FC	
SpaceWire Plug-and-Play - Version	RTR.PNPVER			0x00000001
SpaceWire Plug-and-Play - Device Status	RTR.PNPDEVSTS			0x00000002
SpaceWire Plug-and-Play - Active Links	RTR.PNPACTLNK			0x00000003
SpaceWire Plug-and-Play - Link Information	RTR.PNPLNKINFO			0x00000004
SpaceWire Plug-and-Play - Owner Address 0	RTR.PNPOA0			0x00000005
SpaceWire Plug-and-Play - Owner Address 1	RTR.PNPOA1			0x00000006
SpaceWire Plug-and-Play - Owner Address 2	RTR.PNPOA2			0x00000007
SpaceWire Plug-and-Play - Device ID	RTR.PNPDEVID			0x00000008
SpaceWire Plug-and-Play - Vendor String Length	RTR.PNPVSTRL			0x00004000
SpaceWire Plug-and-Play - Product String Length	RTR.PNPPSTRL			0x00006000
SpaceWire Plug-and-Play - Protocol Count	RTR.PNPPCNT			0x00008000
SpaceWire Plug-and-Play - Application Count	RTR.PNPACNT			0x0000C000

Table 8. GR718B register overview

Register name	Register acronym	AMBA address	RMAP address	SpW PnP address
SpaceWire In-System Test registers (section 7.3)				
SpaceWire Address Register 0-7	SIST.ADDR[0-7]	0xFFE00200 - 0xFFE0021C		
Protocol ID and Polynomial Register	SIST.PID	0xFFE00220		
Seed Register	SIST.SEED	0xFFE00224		
Packet Length Register	SIST.LEN	0xFFE00228		
Control Register	SIST.CTRL	0xFFE0022C		
Error Register 0	SIST.ERROR0	0xFFE00230		
Error Register 1	SIST.ERROR1	0xFFE00234		
Error Register 2	SIST.ERROR2	0xFFE00238		
Packet Counter Register	SIST.PKTCNTR	0xFFE0023C		
Timer Register 0	SIST.TIMER0	0xFFE00240		
Timer Register 1	SIST.TIMER1	0xFFE00244		
Status Register	SIST.STAT	0xFFE00248		
State Register	SIST.STATE	0xFFE0024C		
Transmitter Byte Count Register	SIST.TXBYTECNTR	0xFFE00250		
Receiver Byte Count Register	SIST.RXBYTECNTR	0xFFE00254		
Time-Code Register	SIST.TIME	0xFFE00258		
Protection Register	SIST.PROT	0xFFE0025C		
Data Input Registers 0 - 7	SIST.DIN[0-7]	0xFFE00280 - 0xFFE0029C		
Data Output Registers 0 - 7	SIST.DOUT[0-7]	0xFFE002A0 - 0xFFE002BC		
UART Interface registers (section 8.3)				
AHB UART status register	UART.STS	0xFFE00004		
AHB UART control register	UART.CTRL	0xFFE00008		
AHB UART scaler register	UART.SCALE	0xFFE0000C		
SPI Controller registers (section 10.3)				
Capability register	SPI.CAP	0xFFF22000	0x00002000	
Mode register	SPI.MODE	0xFFF22020	0x00002020	
Event register	SPI.EVENT	0xFFF22024	0x00002024	
Mask register	SPI.MASK	0xFFF22028	0x00002028	
Command register	SPI.CMD	0xFFF2202C	0x0000202C	
Transmit register	SPI.TX	0xFFF22030	0x00002030	
Receive register	SPI.RX	0xFFF22034	0x00002034	

Table 8. GR718B register overview

Register name	Register acronym	AMBA address	RMAP address	SpW PnP address
Slave Select register (optional)	SPI.SLVSEL	0xFFF22038	0x00002038	
Automatic slave select register	SPI.ASLVSEL	0xFFF2203C	0x0000203C	
General Purpose I/O Interface registers (section 11.2)				
I/O port data register	GPIO.DATA	0xFFF22100	0x00002100	
I/O port output register	GPIO.OUT	0xFFF22104	0x00002104	
I/O port direction register	GPIO.DIR	0xFFF22108	0x00002108	
Capability register	GPIO.CAP	0xFFF2211C	0x0000211C	
System Level Test Configuration registers (section 12.1)				
System level test configuration 1	SYSTEST.CFG1	0xFFE00100		
System level test configuration 2	SYSTEST.CFG2	0xFFE00104		

An example register, showing the register layout used throughout this document, can be seen in table 9. The values used for the reset value fields are described in table 10, and the values used for the field type fields are described in table 11. Fields that are named RESERVED, RES, or R are read-only fields that always reads as zero.

Table 9. <Address> - <Register acronym> - <Register name>

31 24	23 16	15 8	7 0
EF3	EF2	EF1	EF0
<Reset value for EF3>	<Reset value for EF2>	<Reset value for EF1>	<Reset value for EF0>
<Field type for EF3>	<Field type for EF2>	<Field type for EF1>	<Field type for EF0>

31: 24 Example field 3 (EF3) - <Field description>
23: 16 Example field 2 (EF2) - <Field description>
15: 8 Example field 1 (EF1) - <Field description>
7: 0 Example field 0 (EF0) - <Field description>

Table 10. Reset value definitions

Value	Description
0	Reset value 0. Used for single-bit fields.
1	Reset value 1. Used for single-bit fields.
0xNN	Hexadecimal representation of reset value. Used for multi-bit fields.
N/R	Field not reseted
*	Special reset condition, described in textual description of the field. Used for example when reset value is taken from a pin.

Table 11. Field type definitions

Value	Description
r	Read-only. Writes have no effect.
w	Write-only. Used for a writable field in a register that is not readable.
rw	Readable and writable.
rw*	Readable and writable. Special condition for write, described in textual description of field.
wc	Write-clear. Readable, and cleared when written with a 1
cas	Readable, and writable through compare-and-swap. Only applies to SpaceWire Plug-and-Play registers.

3 I/O Pins

The GR718B has the external pins shown in table 12. The AC characteristic is found in section 15.6, and the pin assignments is found in section 16.2.

Table 12. External pins

Name	Usage	Direction	Active	Reset value
RESETn	System reset	In	Low	-
CLK	Main system clock	In	-	-
SPWCLK	External SpaceWire clock	In	-	-
SPWCLKSEL[2:0]	Selects internal SpaceWire clock. See section 4.	In	-	-
LOCK	Internal SpaceWire clock PLL lock. See section 4.	Out	-	1)
IRQ	Interrupt output, indicating internal error or status change	Out	High	0
TESTEN[1:0]	Test mode enable pins. Keep low for normal operation.	In	High	-
TX	UART transmit data	Out	-	1
RX	UART receive data	In	-	-
TCK	JTAG Clock	In	-	-
TMS	JTAG Mode	In	-	-
TDI	JTAG Input	In	-	-
TDO	JTAG Output	Out	-	0
SPW_RXDp[16:1]	SpaceWire data input for ports 16:1, positive (LVDS)	In	High	-
SPW_RXDn[16:1]	SpaceWire data input for ports 16:1, negative (LVDS)	In	Low	-
SPW_RXSp[16:1]	SpaceWire strobe input for ports 16:1, positive (LVDS)	In	High	-
SPW_RXSn[16:1]	SpaceWire strobe input for ports 16:1, negative (LVDS)	In	Low	-
SPW_TXDp[16:1]	SpaceWire data output for ports 16:1, positive (LVDS)	Out	High	0x0000
SPW_TXDn[16:1]	SpaceWire data output for ports 16:1, negative (LVDS)	Out	Low	0xFFFF
SPW_TXSp[16:1]	SpaceWire strobe output for ports 16:1, positive (LVDS)	Out	High	0x0000
SPW_TXSn[16:1]	SpaceWire strobe output for ports 16:1, negative (LVDS)	Out	Low	0xFFFF
SPW_RXD[18:17]	SpaceWire data input for ports 18:17 (LVTTL)	In,	High	-
SPW_RXS[18:17]	SpaceWire strobe input for ports 18:17 (LVTTL)	In	High	-
SPW_TXD[18:17]	SpaceWire data output for ports 18:17 (LVTTL)	Out	High	0
SPW_TXS[18:17]	SpaceWire strobe output for ports 18:17 (LVTTL)	Out	High	0
SPWEN[18:17]	SpaceWire transceiver enable for port 18:17	Out	High	1
GPIOSEL	Selects the function of the GPIO[23:0] pins. See section 5	In	-	-
GPIO[23:0]	General purpose input / output pins. See section 5	BiDir	-	- 2)
AUXTICKIN	Auxiliary time-code / distributed interrupt code tick in. See section 6.2.16	In	High	-
AUXTICKOUT	Auxiliary time-code / distributed interrupt code tick out. See section 6.2.16	Out	High	0
SPI_SCK	SPI clock	Out	-	0
SPI_MOSI	SPI master out slave in	Out	-	0
SPI_MISO	SPI master in slave out	In	-	-
SPI_SLVSEL	SPI slave select	Out	-	0

Table 12. External pins

Name	Usage	Direction	Active	Reset value
CFGLOCK	Locks configuration port (port 0) from accesses from all ports except port 1 and 2. See section 6.5.1.3 for details.	In	High	-
SPWCLKDIV[5:0]	Reset value for bits 5:0 of the SpaceWire port's clock divisor register (initialization and run-state). Reset value for bits 5:0 of the SpaceWire port's clock divisor registers, see RTR.IDIV.ID in Table 36 for initialization and RTR.PCTRL.RD in Table 27 for run-state, and section 6.2.20 for their interpretation.	In	-	-
SPILLIFNOTREADY	Sets the reset value for the ports' spill-if-not-ready feature. See section 6.2.7 for details.	In	High	-
PNPEN	Enables / disable SpaceWire Plug-and-Play at reset. See section 6.5.4 for details.	In	High	-
LINKSTARTONREQ	Reset value for the SpaceWire ports' link-start-on-request feature. See section 6.2.10 for details.	In	High	-
AUTODCONNECT	Reset value for the SpaceWire ports' auto-disconnect feature. See section 6.2.11 for details.	In	High	-
STATICROUTEEN	Enables / disables the static routing feature at reset. See section 6.2.6 for details.	In	High	-

Note 1: LOCK is low while RESETn is asserted, and while PLL has not yet acquired lock. LOCK is high if either PLL has acquired lock, or the PLL is bypassed (SPWCLKSEL = 000).

Note 2: During reset, the GPIO pins are in input mode. After reset, the value and direction is as defined by table 14.

Note 3: No on-chip support for cold spare or fail safe for any input pr output pins.

4 Clocks and reset

4.1 Clocks

There are three clock input pins: CLK, SPWCLK, and TCK.

The CLK pin is used as the main system clock, and directly drives the clock network without any PLLs, or muxes in between. The CLK pin can also be used to generate the internal SpaceWire clock (*int_spwclk)*, depending on the value of the SPWCLKSEL[2:0] pins.

The SPWCLK pin is the external SpaceWire clock, and it can be used as the internal SpaceWire clock (*int_spwclk*) either directly, or multiplied with a PLL, depending on the value of the SPWCLK-SEL[2:0] pins.

The TCK pin is used as the JTAG clock.

When generating the internal SpaceWire clock, the SPWCLKSEL[2:0] pins are used according to table 13 to select which clock to use.

The internal signal *spwclklock* (PLL output that indicates that it has acquired lock) is mapped to the LOCK pin, and can also be seen through the RTR.GPIA.CL register bit, together with a sticky bit (RTR.GPIA.LL) that indicates if PLL lock has been lost since the last time the sticky bit was cleared.

For information about the relationship between the internal SpaceWire clock and SpaceWire transmission rate, please see section 6.2.20.

Table 13. Internal SpaceWire clock muxing

SPWCLKSEL[2:0]	**Internal SpaceWire clock (*int_spwclk*)**
000	SPWCLK (PLL bypassed and powered down, and LOCK / *spwclklock* is constant 1)
001	2 x CLK (SPWCLK unused)
010	4 x CLK (SPWCLK unused)
011	8 x CLK (SPWCLK unused)
100	1 x SPWCLK
101	2 x SPWCLK
110	4 x SPWCLK
111	8 x SPWCLK

4.2 Reset

The reset circuitry used for the system clock domain is shown in figure 4. The reset signal that is routed to the internal flip-flops is asserted asynchronously and de-asserted synchronously with the system clock. The output from the reset circuitry is fed through combinatorial logic to the D input of the flip-flops in the design.

For critical external signals, such as the bi-directional GPIO[23:0] pins, the corresponding control signals are asynchronously disabled directly from the RESETn pin.

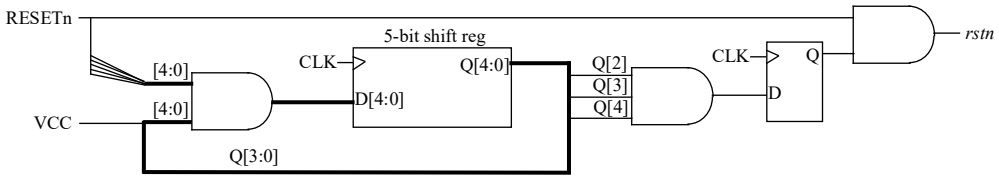


Figure 4. Reset generator circuit, system clock domain

The reset circuitry used for the SpaceWire clock domain is shown in figure 5. The functionality is the same as for the system clock domain, except that the internal signal *spwclklock* (PLL output that indicates that it has acquired lock) is used in the reset circuitry for the SpaceWire clock domain. The signal *int_spwclk* shown in figure 5 is the internal SpaceWire clock, derived according to table 13.

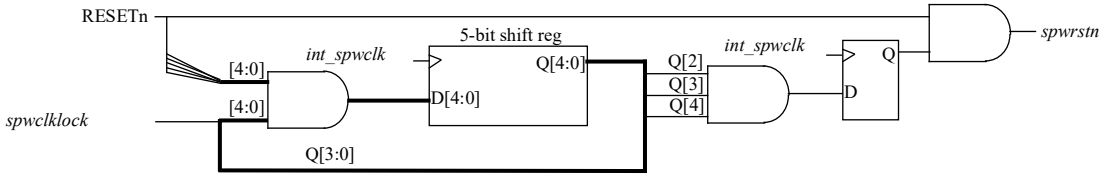


Figure 5. Reset generator circuit, SpaceWire clock domain

5 GPIO pin multiplexing

Table 14 shows how the function of the GPIO[23:0] pins is affected depending on the value of the RESETn and GPIOSEL pins, and the setting of the RTR.GPOA / RTR.GPOB registers. The Function column shows both how the pin is used during reset, and which internal signal the pin is mapped to while not in reset. The internal signals shown in table 14 are:

- *auxtimein[7:0]*, *auxtimeinen*, *auxtimeout[7:0]* - Signals to / from the router's auxiliary time-code / distributed interrupt code interface, described in section 6.2.16. Reset value of the *auxtime-out[7:0]* signals is 0x00.
- *gpio[23:0]* - Signals to / from the general purpose I/O interface, described in section 11.
- *linkrun[18:1]* - The *linkrun[18:1]* signals are high when the link interface for the corresponding SpaceWire port is in run-state. Reset value of the *linkrun[18:1]* signals is 0x00000.
- *porterr[19:0]* - The *porterr[19:0]* signals are high if an error has occurred in the corresponding port. See section 6.2.19 for information about which errors that set the *porterr[19:0]* signals. Reset value for the *porterr[19:0]* is 0x00000.
- *spwen[16:1]* - In addition to the two dedicated SPWEN[18:17] pins, these 16 extra SpaceWire transceiver enable signals are provided. See section 6.2.21 for more information. Reset value for the *spwen[16:1]* is 0xFFFF.
- *spi_slvsel[5:1]* - In addition to the dedicated SPI_SLVSEL pin, the SPI controller outputs these extra five slave select signals. See section 10 for more information. Reset value for the *spi_slv-sel[5:1]* signals is 0x00.

Table 14. GPIO pin function table

Pin	RESETn	GPIOSEL	RTR.GPO[A/B].GPO	Direction	Function
GPIO[0]	0	-	--	In	
	1	0	--	In	*auxtimein[0]*
		1	00, 01, 11	In/Out [1)]	*gpio[0]*
			10	Out	*porterr[0]*
GPIO[1]	0	-	--	In	
	1	0	--	In	*auxtimein[1]*
		1	00	In/Out [1)]	*gpio[1]*
			01	Out	*linkrun[1]*
			10	Out	*porterr[1]*
			11	Out	*spwen[1]*
GPIO[2]	0	-	--	In	
	1	0	--	In	*auxtimein[2]*
		1	00	In/Out [1)]	*gpio[2]*
			01	Out	*linkrun[2]*
			10	Out	*porterr[2]*
			11	Out	*spwen[2]*
GPIO[3]	0	-	--	In	
	1	0	--	In	*auxtimein[3]*
		1	00	In/Out [1)]	*gpio[3]*
			01	Out	*linkrun[3]*
			10	Out	*porterr[3]*
			11	Out	*spwen[3]*

Table 14. GPIO pin function table

Pin	RESETn	GPIOSEL	RTR.GPO[A/B].GPO	Direction	Function
GPIO[4]	0	-	--	In	
	1	0	--	In	*auxtimein[4]*
		1	00	In/Out [1)]	*gpio[4]*
			01	Out	*linkrun[4]*
			10	Out	*porterr[4]*
			11	Out	*spwen[4]*
GPIO[5]	0	-	--	In	
	1	0	--	In	*auxtimein[5]*
		1	00	In/Out [1)]	*gpio[5]*
			01	Out	*linkrun[5]*
			10	Out	*porterr[5]*
			11	Out	*spwen[5]*
GPIO[6]	0	-	--	In	
	1	0	--	In	*auxtimein[6]*
		1	00	In/Out [1)]	*gpio[6]*
			01	Out	*linkrun[6]*
			10	Out	*porterr[6]*
			11	Out	*spwen[6]*
GPIO[7]	0	-	--	In	
	1	0	--	In	*auxtimein[7]*
		1	00	In/Out [1)]	*gpio[7]*
			01	Out	*linkrun[7]*
			10	Out	*porterr[7]*
			11	Out	*spwen[7]*
GPIO[8]	0	-	--	In	
	1	0	--	In	*auxtimeinen*
		1	00	In/Out [1)]	*gpio[8]*
			01	Out	*linkrun[8]*
			10	Out	*porterr[8]*
			11	Out	*spwen[8]*
GPIO[9]	0	-	--	In	
	1	0	--	Out	*auxtimeout[0]*
		1	00	In/Out [1)]	*gpio[9]*
			01	Out	*linkrun[9]*
			10	Out	*porterr[9]*
			11	Out	*spwen[9]*
GPIO[10]	0	-	--	In	
	1	0	--	Out	*auxtimeout[1]*
		1	00	In/Out [1)]	*gpio[10]*
			01	Out	*linkrun[10]*
			10	Out	*porterr[10]*
			11	Out	*spwen[10]*

Table 14. GPIO pin function table

Pin	RESETn	GPIOSEL	RTR.GPO[A/B].GPO	Direction	Function
GPIO[11]	0	-	--	In	See table 15
	1	0	--	Out	*auxtimeout[2]*
		1	00	In/Out [1)]	*gpio[11]*
			01	Out	*linkrun[11]*
			10	Out	*porterr[11]*
			11	Out	*spwen[11]*
GPIO[12]	0	-	--	In	See table 16
	1	0	--	Out	*auxtimeout[3]*
		1	00	In/Out [1)]	*gpio[12]*
			01	Out	*linkrun[12]*
			10	Out	*porterr[12]*
			11	Out	*spwen[12]*
GPIO[13]	0	-	--	In	See table 16
	1	0	--	Out	*auxtimeout[4]*
		1	00	In/Out [1)]	*gpio[13]*
			01	Out	*linkrun[13]*
			10	Out	*porterr[13]*
			11	Out	*spwen[13]*
GPIO[14]	0	-	--	In	See table 16
	1	0	--	Out	*auxtimeout[5]*
		1	00	In/Out [1)]	*gpio[14]*
			01	Out	*linkrun[14]*
			10	Out	*porterr[14]*
			11	Out	*spwen[14]*
GPIO[15]	0	-	--	In	See table 16
	1	0	--	Out	*auxtimeout[6]*
		1	00	In/Out [1)]	*gpio[15]*
			01	Out	*linkrun[15]*
			10	Out	*porterr[15]*
			11	Out	*spwen[15]*
GPIO[16]	0	-	--	In	See table 17
	1	0	--	Out	*auxtimeout[7]*
		1	00	In/Out [1)]	*gpio[16]*
			01	Out	*linkrun[16]*
			10	Out	*porterr[16]*
			11	Out	*spwen[16]*
GPIO[17]	0	-	--	In	See table 17
	1	-	00	In/Out [1)]	*gpio[17]*
			01	Out	*linkrun[17]*
			10	Out	*porterr[17]*
			11	Out	*spi_slvsel[1]*

Table 14. GPIO pin function table

Pin	RESETn	GPIOSEL	RTR.GPO[A/B].GPO	Direction	Function
GPIO[18]	0	-	--	In	See table 18
	1	-	00	In/Out [1)]	*gpio[18]*
			01	Out	*linkrun[18]*
			10	Out	*porterr[18]*
			11	Out	*spi_slvsel[2]*
GPIO[19]	0	-	--	In	See table 18
	1	-	00, 01	In/Out [1)]	*gpio[19]*
			10	Out	*porterr[19]*
			11	Out	*spi_slvsel[3]*
GPIO[20]	0	-	--	In	See table 19
	1	-	00, 01, 10	In/Out [1)]	*gpio[20]*
			11	Out	*spi_slvsel[4]*
GPIO[21]	0	-	--	In	See table 19
	1	-	00, 01, 10	In/Out [1)]	*gpio[21]*
			11	Out	*spi_slvsel[5]*
GPIO[22]	0	-	--	In	See table 20
	1	-	--	In/Out [1)]	*gpio[22]*
GPIO[23]	0	-	--	In	See table 20
	1	-	--	In/Out [1)]	*gpio[23]*

Note 1: The direction of the GPIO pin is controlled by the corresponding bit in the GPIO.DIR register.

Table 15 through 20 shows how the GPIO pins are used for reset values.

Table 15. Auxiliary time-code / distributed interrupt code interface configuration

GPIO[11]	Register field	Reset value	Resulting functionality
0	RTR.RTRCFG.AA	0	The inputs for the auxiliary time-code / distributed interrupt code interface are synchronous to CLK. See section 6.2.16.
1	RTR.RTRCFG.AA	1	The inputs for the auxiliary time-code / distributed interrupt code interface are asynchronous to CLK. See section 6.2.16.

Table 16. Instance ID reset value

GPIO[15:12]	Register field	Resulting Instance ID
----	RTR.VER.ID RTR.PNPUSN	RTR.VER.ID[3:0] and RTR.PNPUSN[3:0] is reset from GPIO[15:12], while RTR.VER.ID[7:4] is always reset to 0x0.

Table 17. Prescaler reset values

GPIO[17:16]	Register field	Reset value	Resulting prescaler tick interval (with 50 MHz system clock)
00	RTR.PRESCALER.RL	49	1 us
01		499	10 us
10		4999	100 us
11		65535	1.3 ms

Table 18. Port timer reset values

GPIO[19:18]	Register field	Reset value	Resulting port timer settings
00	RTR.PTIMER.RL	1	2 x prescaler tick
	RTR.PCTRL.TR RTR.PCTRLCFG.TR	0	Port data character timer disabled
01	RTR.PTIMER.RL	1	2 x prescaler tick
	RTR.PCTRL.TR RTR.PCTRLCFG.TR	1	Port data character timer enabled
10	RTR.PTIMER.RL	511	512 x prescaler tick
	RTR.PCTRL.TR RTR.PCTRLCFG.TR	1	Port data character timer enabled
11	RTR.PTIMER.RL	1023	1024 x prescaler tick
	RTR.PCTRL.TR RTR.PCTRLCFG.TR	1	Port data character timer enabled

Table 19. Interrupt distribution timeout reset values

GPIO[21:20]	Register field	Reset value	Resulting timeout values
00	RTR.ISRTIMER.RL	1	2 x prescaler tick
	RTR.AITIMER.RL	0	1 x prescaler tick
01	RTR.ISRTIMER.RL	63	64 x prescaler tick
	RTR.AITIMER.RL	31	32 x prescaler tick
10	RTR.ISRTIMER.RL	511	512 x prescaler tick
	RTR.AITIMER.RL	255	256 x prescaler tick
11	RTR.ISRTIMER.RL	1023	1024 x prescaler tick
	RTR.AITIMER.RL	512	512 x prescaler tick

Table 20. Time-code / distributed interrupt code reset configuration.

GPIO[23:22]	Register field	Reset value	Resulting functionality
00	RTR.TC.EN	1	Time-codes are globally enabled.
	RTR.RTRCFG.TF	1	Control flag filtering enabled. Time-codes must have bits 7:6 = "00".
	RTR.RTRCFG.IE	1	Distributed interrupt codes globally enabled.
	RTR.PCTRL.IC	1	Distributed interrupt codes are enabled for each port.
	RTR.RTRCFG.EE	0	Distributed interrupts are operating in interrupt with acknowledgement mode (32 possible interrupts, with 32 acknowledgements).
	RTR.RTRCFG.AI	1	Distributed interrupt codes are can be sent and received on the auxiliary time-code / distributed interrupt code interface
01	RTR.TC.EN	1	Time-codes are globally enabled.
	RTR.RTRCFG.TF	1	Control flag filtering enabled.Time-codes must have bits 7:6 = "00".
	RTR.RTRCFG.IE	1	Distributed interrupt codes globally enabled.
	RTR.PCTRL.IC	1	Distributed interrupt codes are enabled for each port.
	RTR.RTRCFG.EE	1	Distributed interrupts are operating in extended interrupt mode (64 possible interrupts, without acknowledge-ments).
	RTR.RTRCFG.AI	1	Distributed interrupt codes are can be sent and received on the auxiliary time-code / distributed interrupt code interface

Table 20. Time-code / distributed interrupt code reset configuration.

GPIO[23:22]	Register field	Reset value	Resulting functionality
10	RTR.TC.EN	1	Time-codes are globally enabled.
	RTR.RTRCFG.TF	0	Control flag filtering is disabled. Time-codes are allowed to have bits 7:6 set to any value.
	RTR.RTRCFG.IE	0	Distributed interrupt codes globally disabled.
	RTR.PCTRL.IC	0	Distributed interrupt codes are disabled for each port.
	RTR.RTRCFG.EE	0	Distributed interrupts are operating in interrupt with acknowledgement mode (32 possible interrupts, with 32 acknowledgements).
	RTR.RTRCFG.AI	0	Distributed interrupt codes received on the auxiliary time-code / distributed interrupt code interface are silently discarded.
11	RTR.TC.EN	1	Time-codes are globally enabled
	RTR.RTRCFG.TF	1	Control flag filtering enabled. Time-codes must have bits 7:6 = "00".
	RTR.RTRCFG.IE	0	Distributed interrupt codes are globally disabled.
	RTR.PCTRL.IC	0	Distributed interrupt codes are disabled for each port.
	RTR.RTRCFG.EE	0	Distributed interrupts are operating in interrupt with acknowledgement mode (32 possible interrupts, with 32 acknowledgements).
	RTR.RTRCFG.AI	0	Distributed interrupt codes received on the auxiliary time-code / distributed interrupt code interface are silently discarded.

6 SpaceWire Router

6.1 Overview

The SpaceWire router implements a SpaceWire routing switch as defined in ECSS-E-ST-50-12C [SPW]. It provides a total of 20 ports, where port 0 is a configuration port, ports 1-18 are SpaceWire ports, and port 19 is a custom port called the SIST port. Each SpaceWire port contains a SpaceWire codec, and provides an external SpaceWire interface. The SIST port provides a FIFO interface which is internally connected to the SpaceWire In-System Test, described in section 7. The configuration port provides an RMAP target, and an AMBA AHB slave interface, both used for accessing internal configuration and status registers. The SPI and General purpose I/O interfaces, described in sections 10 and 11, are also accessible through the configuration port. The configuration port also provides a SpaceWire Plug-and-Play interface, allowing device identification.

The switch matrix can connect any input port to any output port. Access to each output port is arbitrated using a round-robin arbitration scheme based on the address of the incoming packet. A single routing-table is used for the whole router, where access to the table is arbitrated using a round-robin scheme based on the input port number.

Among the features supported by the router are: group adaptive routing, packet distribution, system time-distribution, distributed interrupts, port timers to recover from deadlock situations, and SpaceWire-D [SPWD] packet truncation based time-slot violations.

6.2 Operation

6.2.1 Routing table

A single routing table is provided. The access to the 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 do not arrive at the same time. The probability for this is higher when the traffic only consists 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 routing table is configured with either RMAP or AMBA AHB accesses to the configuration port. Configuration of the routing table does not introduce any extra latency for packets, since the configuration accesses have lower priority than packet traffic. The routing table is split into two parts, one which controls the port mapping for the address (RTR.RTPMAP registers), and one which controls properties for the address, such as priority and header deletion (RTR.RTACTRL registers).

6.2.1.1 Port mapping

For both physical and logical addresses it is possible to configure which port(s) the incoming packet should be routed to. This is done by programming the corresponding RTR.RTPMAP register. The RTR.RTPMAP registers also controls whether or not group adaptive routing or packet distribution should be used for the incoming packet. The RTR.RTPMAP registers are not initialized after reset / power-up. For physical addresses this has the effect that the incoming packet is routed to the port that matches the address in the packet, without any group adaptive routing or packet distribution. For logical addresses, an uninitialized RTR.RTPMAP register (or if the RTR.RTPMAP.PE field has been written with all zeros) has the effect that the incoming packet is spilled. See table 24 in section 6.5.3 for more details.

6.2.1.2 Address control

For both physical and logical addresses it is possible to configure the priority, and to enable the spill-if-not-ready feature (explained in section 6.2.7). For logical addresses it is also possible to enable / disable the address, and enable / disable header deletion. Physical addresses are always enabled, and always have header deletion enabled, as specified by ECSS-E-ST-50-12C [SPW]. This configuration for an address is done by programming the corresponding RTR.RTACTRL register. Logical addresses are disabled after reset / power-up. An incoming packet with a disabled logical address is spilled. See table 25 in section 6.5.3 for details.

6.2.2 Output port arbitration

Each output port is arbitrated individually based on the address of the incoming packet, using two priority levels, with round-robin at each level. Each physical address and logical address can be configured in the routing table (RTR.RTACTRL register) to be either high or low priority. Priority assignments can have large impact on the delays for packets, because packets can be large and the speed of the data consumer and link itself may not be known. This should therefore be considered when assigning priorities.

6.2.3 Group adaptive routing

Group adaptive routing can be used to allow incoming packets to be sent on different output ports depending on which of the output ports that are currently ready. It can be enabled for both physical and logical addresses, and is configured by programming the corresponding RTR.RTPMAP register.

When a packet arrives, and group adaptive routing is enabled for the packet's address, the router looks at the group of ports selected by the corresponding RTR.RTPMAP register and transmits the packet on the port with the lowest index that is currently ready. Ready in this context means that the port's link interface is in run-state and currently not sending any other packet. If none of the selected output ports are ready, the incoming packet will either be spilled or transmitted on the first port that becomes ready. The action taken depends on the setting of the input port's data character timer (see section 6.2.12), the spill-if-not-ready feature for the address (see section 6.2.7), and the link-start-on-request feature for the output ports (see section 6.2.10). See table 24 in section 6.5.3 for details on how to enable and configure group adaptive routing.

6.2.4 Packet distribution

Packet distribution can be used to implement multicast and broadcast addresses, and can be enabled for both physical and logical addresses. Packet distribution is enabled and configured by programming the corresponding RTR.RTPMAP register.

When a packet arrives, and packet distribution is enabled for the packet's address, the router looks at the group of ports selected by the corresponding RTR.RTPMAP register. If all of the selected ports are ready, the packet is transmitted on all the ports. Ready in this context means that the port's link interface is in run-state and currently not sending any other packet. If one or more of the selected ports are not ready, the incoming packet will either be spilled or transmitted once all ports are ready. The action taken depends on the setting of the input port's data character timer (section 6.2.12), the spill-if-not-ready feature for the address (section 6.2.7), and the link-start-on-request feature for the output ports (section 6.2.10). See table 24 in section 6.5.3 for details on how to enable and configure packet distribution.

6.2.5 Port disable

A port can be disabled for data traffic by setting the corresponding RTR.PCTRL.DI bit. Incoming packets on a disabled port are silently spilled, and no packets are routed to a disabled port. A disabled port will not be included in any group used for group adaptive routing or packet distribution, even if the corresponding bit in that address' RTR.RTPMAP.PE field is set. When routing packets that are

incoming on other ports, the router will simply behave as if the disabled port did not exist. The RTR.PCTRL.DI bit only affects routing of data, thus the transmission and reception of time-codes and distributed interrupt codes are not affected.

The link interface for a SpaceWire port is not affected solely by the RTR.PCTRL.DI bit. However, note that the RTR.PCTRL.DI bit, together with the RTR.PCTRL.LD bit, is used to clock-gate a SpaceWire port (see section 6.2.22).

6.2.6 Static routing

The router supports a feature called static routing, which can be enabled individually per port. When enabled, all incoming packets on the port are routed based on the physical address specified in the port's RTR.PCTRL2.SC field, and the setting of the corresponding RTR.PCTRL2.SC bit, instead of the address in the packets. Header deletion is not used for the incoming packets when static routing is enabled, which means that the first byte of the packets are always sent to the output port as well. Static routing to port 0 is not allowed, and will generate an invalid address error if attempted.

The STATICROUTEEN pin works as a global enable / disable for the static routing feature during reset. The feature is enabled if the pin is high during reset, and disabled if the pin is low during reset. The RTR.RTRCFG.SR bit shows if the static routing feature is globally enabled or disabled.

Note that when static routing is enabled for a port, it is not possible to access the configuration port from the same port.

6.2.7 Spill-if-not-ready

The spill-if-not-ready feature can be enabled individually for each physical and logical address by configuring the corresponding RTR.RTACTRL.SR bit. When enabled, an incoming packet is spilled if the selected output port's link interface is not in run-state. If group adaptive routing is used for the incoming packet then the packet is only spilled if none of the ports in the group is in run-state. If packet distribution is used for the incoming packet then the packet is spilled unless the link interfaces for all selected output ports are in run-state. The spill-if-not-ready feature has priority over the incoming port's data character timer (section 6.2.12) and the output port's link-start-on-request feature (section 6.2.10). This means that if the spill-if-not-ready feature is enabled, the packet is spilled before the timer starts, and the link-start-on-request feature will never be activated.

6.2.8 Self addressing

Self addressing occurs when a selected output port for a packet is the same port as the input port. Whether or not this is allowed is controlled by the RTR.RTRCFG.SA bit. If self addressing is not allowed, the incoming packet is spilled and an invalid address error occurs.

When group adaptive routing is used, and self addressing is not allowed, the input port is still allowed to be in the group of ports configured for the packet. The packet is not spilled until the router actually selects the input port as output port. If the router selects one of the other ports in the group, the packet is not spilled.

When packet distribution is used, and self addressing is not allowed, the input port is not allowed to be in the group of ports configured for the packet, since the packet should be sent to all ports in the group.

6.2.9 Invalid address error

An invalid address error occurs under the conditions listed below.

- When an incoming packet's address corresponds to a non-existing port (physical addresses 20-31).
- When an incoming packet's address is a logical address that is not enabled (RTR.RTACTRL.EN = 0).

- When an incoming packet's address is a logical address for which the corresponding RTR.RTPMAP register is not initialized, or the corresponding RTR.RTPMAP.PE field set to all zeroes.
- When only one output port is selected for an incoming packet, and that port is disabled (RTR.PCTRL.DI = 1).
- When self addressing occurs, and the router is configured not to allow self addressing (RTR.RTRCFG.SA = 0).
- When a packet is routed with the static routing feature, and the physical address programmed in RTR.PCTRL2.SD is 0 (static routing to port 0 is not allowed).

For all the invalid address cases above, the incoming packet is spilled, and the RTR.PSTS.IA bit corresponding to the input port will be set to 1.

6.2.10 Link-start-on-request

The link-start-on-request feature gives the possibility to automatically start a SpaceWire port's link interface when a packet is routed to the port (i.e, port is selected as output port). Each port can have the feature individually enabled by setting the corresponding RTR.PCTRL.LR bit to 1.

If a packet arrives, and the link interface of the selected output port is not in run-state, and the port has the link-start-on-request feature enabled, the router will try to start the link interface under the following conditions:

1. The link interface is not already trying to start (RTR.PCTRL.LS = 0).

2. The link is not disabled (RTR.PCTRL.LD = 0).

3. The spill-if-not-ready feature is not enabled for the packet being routed.

The link will continue to be started until either the RTR.PCTRL.LD bit is set to 1, or until the link is disabled through the auto-disconnect feature, described in section 6.2.11.

The link-start-on-request feature is only available for the SpaceWire ports, since the configuration port and SIST port does not have a link interface FSM, and are therefore always considered to be in run-state.

6.2.11 Auto-disconnect

The auto-disconnect feature gives the possibility to automatically disable the link interface of a SpaceWire port if the port has been inactive for a long enough period of time. Each port can have the feature individually enabled by setting their corresponding RTR.PCTRL.AD bit to 1. The amount of time the port needs to be inactive for is decided by the settings of the global prescaler register (RTR.PRESCALER), and the port's individual timer register (RTR.PTIMER). This time period is the same as the timeout period used by the port's data character timer when recovering from deadlock situations (see section 6.2.12). If the auto-disconnect feature is enabled, then a SpaceWire port will automatically disable its link interface under the following conditions:

1. The link interface entered run-state because it was started by the link-start-on-request feature, described in section 6.2.10.

2. The packet that caused the link interface to start has finished (either sent or spilled).

3. Nothing has been transmitted or received on the port for the duration of the time period specified by the RTR.PRESCALER register, and the corresponding RTR.PTIMER register.

4. The port's corresponding RTR.PCTRL.LS bit has not been set to 1.

The auto-disconnect feature is only available for the SpaceWire ports, since the configuration port and SIST port does not have a link interface FSM, and are therefore always considered to be in run-state.

6.2.12 Port data character timers

Each port has an individual data character timer, which can be used to timeout an ongoing data transfer in order to recover from a deadlock situation. There are two different timeouts defined: overrun timeout, and underrun timeout. An overrun timeout is when the input port has data available, but the output port(s) can not accept data fast enough. An underrun timeout is when the output port(s) can accept more data, but the input port can not provide data fast enough.

The timeout period for a port is set in its corresponding RTR.PTIMER register, and the timer is enabled through the corresponding RTR.PCTRL.TR bit. Timeouts due to overrun and underrun can also be individually enabled / disabled through the corresponding RTR.PCTRL2.OR and RTR.PCTRL2.UR bits.

It is always the input port's data character timer that is used for timing data transfers. When the timer is enabled, it counts down on every tick from the global prescaler (RTR.PRESCALER register). If a data character is transmitted from the input port to the output port(s), then the timer is restarted. If the timer expires, the ongoing packet is spilled, and an EEP is written to the transmit FIFO of the output port(s).

The range of the timeout period depends on the system clock frequency, and is calculated with the following formula:

<timeout period> = (<clock period> x (RTR.PRESCALER+1)) x RTR.PTIMER

Sub-sections 6.2.12.1 through 6.2.12.4 clarifies the behaviour of the timers for different scenarios that can occur when a packet arrives.

6.2.12.1 Timer disabled

If the data character timer for an input port is disabled, the incoming packet will wait indefinitely for the output port(s) to be ready, unless the spill-if-not-ready feature is enabled for the packet's address (see section 6.2.7).

6.2.12.2 Timer enabled, but output port(s) not in run state

If the spill-if-not-ready feature (see section 6.2.7) is disabled for the incoming packet's address, the input port's data character timer is started when the packet arrives, and if the output port's link interface has not entered run-state when the timer expires, the packet is spilled. When group adaptive routing is used for the incoming packet, it is enough for one of the possible output ports to enter run-state before the timer expires for the packet to be transmitted. If packet distribution is used, all the output ports must enter run-state before the timer expires, otherwise the packet is spilled.

If the link-start-on-request feature is enabled for an output port, that port will try to enter run-state when the packet arrives. However, the input port's timer is unaffected of this, and will still only wait for its configured timeout period, before spilling the packet.

A timeout due to the output port not being in run-state is classified as a overrun timeout, which means that the RTR.PCFG2.OR bit for the input port must be set in order for the packet to be spilled. If the RTR.PCFG2.OR bit is not set, the packet will wait indefinitely (unless spill-if-not-ready is enabled).

6.2.12.3 Timer enabled, output port(s) in run-state but busy with other transmission

The input port's data character timer will not start, and the incoming packet will wait indefinitely until the output port either becomes free or leaves run-state. When group adaptive routing is used for the incoming packet, it is enough for one of the possible output ports to be in run-state to prevent the timer from starting. If packet distribution is used, all the output ports must be in run-state to prevent the timer from starting.

6.2.12.4 Timer functionality when accessing the configuration port

The timer functionality is basically the same for the configuration port as for the other ports. When the command is being received, the configuration port is the output port of the data transfer, and when the reply is being sent, the configuration port is the input port of the data transfer. The differences between the configuration port and the other ports are:

- The configuration port can always accept data fast enough, which means that an overrun timeout will never occur when a command is being received.
- The configuration port can always send data fast enough, which means that an underrun timeout will never occur when a reply is being sent.

6.2.13 Packet length truncation

Packet length truncation monitors the length of an incoming packet, and increases a counter for each received data character. If the counter reaches a value larger than the input port's RTR.MAXPLEN register, and truncation is enabled for the input port (RTR.PCTRL.PL = 1), the rest of the packet is spilled, and an EEP is written to the FIFO of the output port(s). Each port has its own RTR.MAX-PLEN register and counter in order to allow different maximum lengths for different ports.

Packet length truncation can also be enabled for port 0. In that case, it is the length of the RMAP / SpaceWire Plug-and-Play reply packet that is monitored.

6.2.14 System time-distribution

The router supports system time distribution through time-codes, as defined in ECSS-E-ST-50-12C [SPW]. It contains a global time-counter register (RTR.TC) where the latest received time-code can be read. Both the SpaceWire ports and the SIST port support time-code transmission and reception. All incoming time-codes update the RTR.TC register. If the incoming time-code has a time value which is plus one (modulo 64) compared to the old RTR.TC value, the time-code is forwarded to all the other ports. The time-code is not sent out onto the port on which it arrived.

Time-codes can be globally enabled / disabled through the RTR.TC register, as well as individually enabled / disabled per port through respective RTR.PCTRL.TE bit. When time-codes are disabled for a port, all incoming time-codes on that port are discarded, and no time-codes will be forwarded to the port.

The router can be configured to either filter out all incoming time-codes that does not have the two control flags (bit 7:6) set to "00", or to discard the control flags and allow them to have any value. This configuration is done through the RTR.RTRCFG.TF bit. The value of the control flags for the last received time-code can also be read from the RTR.TC register. Note that if interrupt distribution is globally enabled (RTR.RTRCTRL.IE = 1), only control flags "00" are considered as time-codes, no matter the value of the RTR.RTRCFG.TF bit.

6.2.15 Distributed interrupt support

The router supports SpaceWire distributed interrupts. GR718 can be configured to operate in two modes, interrupt with acknowledgement mode, and extended interrupt mode. In the interrupt with acknowledgement mode, 32 interrupt numbers are supported, while in the extended interrupt mode there is support for 64 interrupt numbers. Which mode to operate in is configured through the RTR.RTRCFG.EE bit.

A distributed interrupt code is a control code that has the control flags (bits 7:6) set to "10". A distributed interrupt code that has bit 5 set to 0 is called an interrupt code, and bits 4:0 specify an interrupt number between 0 and 31. When operating in the interrupt with acknowledgement mode, a distributed interrupt code with bit 5 set to 1 is called an interrupt acknowledgement code, used to acknowledge the interrupt with the interrupt number specified by bits 4:0. When operating in the extended interrupt

mode, a distributed interrupt code with bit 5 set to 1 is called an extended interrupt code, and bits 4:0 specify an interrupt number between 32 and 63.

The interrupt codes and extended interrupt codes are generated by the source of the interrupt event, while the interrupt acknowledgement code is sent by the interrupt handler for the corresponding interrupt number.

The router has two 32-bit ISR register (RTR.ISR0 and RTR.ISR1) where each bit corresponds to one interrupt number. A bit in the ISR registers is set to 1 when an interrupt code, or extended interrupt code, with the corresponding interrupt number is received. A bit in the ISR registers is set to 0 when an interrupt acknowledgement code with the corresponding interrupt number is received. This way the ISR registers reflects the status for all interrupt numbers. Each interrupt number also has its own timer which is used to clear the ISR register bit if an interrupt acknowledgement code is not received before the timer expires (for example if operating in the extended interrupt mode), as well as an optional timer which is used to control how fast a bit in the RTR.ISR register is allowed to toggle. See section 6.2.15.2 for more details on the ISR timers.

6.2.15.1 Receiving and transmitting distributed interrupt codes

When a distributed interrupt code is received on a port, or the auxiliary time-code / distributed interrupt code interface, the following requirements must be fulfilled in order for the code to be distributed:

1. Interrupt distribution is globally enabled (RTR.RTRCTRL.IE = 1), and enabled for the port that received the code (corresponding RTR.PCTRL.IC = 1).

2. If the received code is an interrupt code, the RTR.PCTRL2.IR bit for the port must be set to 1. If the received code is an interrupt acknowledgement code or extended interrupt code, the RTR.PCTRL2.AR bit for the port must be set to 1.

3. If the code is an interrupt code or extended interrupt code, the interrupt number's corresponding bit in RTR.ISR0 or RTR.ISR1 must be 0. If the code is an interrupt acknowledgement code, the corresponding bit in RTR.ISR0 must be 1.

4. No previous distributed interrupt code with the same interrupt number is waiting to be distributed.

5. The ISR change timers (see section 6.2.15.2) are either globally disabled (RTR.RTRCFG.IC = 0) or the interrupt number's corresponding ISR change timer has expired.

If one of the requirements above is not fulfilled, then the received code is discarded. If all of the requirements above are fulfilled, then the received code is placed in a queue. The queue is then serviced in one of the four following ways, depending on the settings of the RTR.RTRCFG.IS and RTR.RTRCFG.IP bits:

1. All interrupt codes have priority over all interrupt acknowledgement codes / extended interrupt codes (RTR.RTRCFG.IP = 0), and the interrupt numbers are serviced through a round-robin scheme (RTR.RTRCFG.IS = 0). This is the default service scheme after reset / power-up.

2. All interrupt codes have priority over all interrupt acknowledgement codes / extended interrupt codes (RTR.RTRCFG.IP = 0), and the interrupt numbers are serviced with priority to lower interrupt numbers (RTR.RTRCFG.IS = 1).

3. All interrupt acknowledgement codes / extended interrupt codes have priority over all interrupt codes (RTR.RTRCFG.IP = 1), and the interrupt numbers are serviced through a round-robin scheme (RTR.RTRCFG.IS = 0).

4. All interrupt acknowledgement codes / extended interrupt codes have priority over all interrupt codes (RTR.RTRCFG.IP = 1), and the interrupt numbers are serviced with priority to lower interrupt numbers (RTR.RTRCFG.IS = 1).

When a distributed interrupt code has been selected from the queue, it is forwarded to all ports (except the port it was received on) that has interrupt distribution enabled (RTR.PCTRL.IC = 1), and that has

enabled transmission of interrupt codes or interrupt acknowledgement codes / extended interrupt codes (RTR.PCTRL2.IT and RTR.PCTRL2.AT respectively).

6.2.15.2 Interrupt distribution timers

Each interrupt number has two corresponding timers, called the ISR timer, and ISR change timer:

The ISR timer is started and reloaded with the value from the RTR.ISRTIMER register each time a received interrupt code / extended interrupt code sets the corresponding RTR.ISR0 / RTR.ISR1 bit to 1. If an interrupt acknowledgement code is received, the corresponding ISR timer is stopped. If the ISR timer expires before an interrupt acknowledgement code is received, the corresponding bit in the RTR.ISR0 or RTR.ISR1 register is cleared. The use of ISR timers is always enabled. In the interrupt with acknowledgement mode, the purpose of the timers is to recover from situations where an interrupt acknowledgement code is lost. In the extended interrupt mode, the purpose of the ISR timers is to limit the rate of which interrupt codes are forwarded. It is important to configure the reload value for the ISR timer correctly. In the interrupt with acknowledgement mode, the reload value must not be less than the worst propagation delay for the interrupt code, plus the maximum delay in the interrupt handler, plus the worst propagation delay for the interrupt acknowledgement code. In the extended interrupt mode, the reload value must not be less than the worst propagation delay for the interrupt code / extended interrupt code.

The ISR change timers are timers that optionally can be used to control the minimum delay between two consecutive changes to the same RTR.ISR0 / RTR.ISR1 bit. The purpose of the timers is to protect against unexpected code occurrences that could occur, for example, due to a network malfunction or a babbling idiot. If the use of ISR change timers is enabled (RTR.RTRCFG.IC = 1), then the ISR change timer for an RTR.ISR0 / RTR.ISR1 bit is started and reloaded with the value from the RTR.ISRCTIMER register each time a received distributed interrupt code makes the RTR.ISR0 / RTR.ISR1 bit change value. Until the timer has expired, the corresponding RTR.ISR0 / RTR.ISR1 bit is not allowed to change value, and any received distributed interrupt code with that interrupt number is discarded. In the extended interrupt mode, the ISR change timers are not used, and should be disabled.

6.2.15.3 Interrupt code generation

In addition to distributing interrupt codes received on the ports, the router can also generate an interrupt code / extended interrupt code when an internal error event occurs, such that the IRQ pin is set to 1. See section 6.2.19 for information about how to control which errors that set the IRQ pin. In addition to the errors described in 6.2.19 the IRQ pin can also be configured to be set when a SpaceWire port's link interface enters run-state.

Everything in sections 6.2.15.1 and 6.2.15.2 also applies when the distributed interrupt code is generated by the router. The only difference is that a distributed interrupt code generated by the router will not be discarded if it is not allowed to be distributed. Instead, the distributed interrupt code will be distributed later, as soon as it is allowed. The only time a distributed interrupt code generated by the router is not distributed is if the bits in RTR.PIP are cleared by software before the interrupt code is allowed to be sent.

The interrupt code generation is controlled through the RTR.ICODEGEN register, and in addition to the enable / disable bit (RTR.ICODEGEN.EN), the following features are available:

- The interrupt number to use for a generated distributed interrupt code is programmable through the RTR.ICODEGEN.IN field.
- The generated distributed interrupt code can be configured to be either level type, or edge type. Level type means that a new distributed interrupt code will be sent as long as the IRQ pin is set (i.e. as long as any bit in the RTR.PIP register is set). Edge type means that a new distributed interrupt code will only be sent when a new error event occurs (an RTR.PIP bit toggles from 0 to 1). The type is selected by the RTR.ICODEGEN.IT bit.

- A timer can be enabled through the RTR.ICODEGEN.TE bit. This timer controls the minimum time between a received interrupt acknowledgement code and the distribution of a new generated interrupt code. The timer is started and reloaded with the value from the RTR.AITIMER register when an interrupt acknowledgement code is received, if the router was the source of the corresponding interrupt code. Until the timer has expired, a new generated interrupt code will not be distributed. The reload value should not be less than the worst propagation delay for the interrupt acknowledgement code. This timer is unused when operating in the extended interrupt mode.
- Through the RTR.ICODEGEN.AH and RTR.ICODEGEN.UA bits the router can be configured to, upon receiving an interrupt acknowledgement code, or the when the ISR timer expires, automatically clear the RTR.PIP bits that were set when the distributed interrupt code was generated.

6.2.16 Auxiliary time-code / distributed interrupt code interface

There is an auxiliary time-code / distributed interrupt code interface that consists of the two dedicated pins AUXTICKIN and AUXTICKOUT, as well as the internal signals *auxtimein[7:0]*, *auxtimeout[7:0]*, and *auxtimeinen*, which can be mapped to GPIO[16:0] pins. The dedicated pins allow both a simple time-tick generator, as well as a simple time-tick observer, without the need to use any of the GPIO[16:0] pins. However, for the full functionality of the time-code / distributed interrupt code interface, the GPIOSEL pin has to be low (see section 5).

The AUXTICKOUT pin is set high for one CLK cycle when a valid time-code or distributed interrupt code is received on any of the ports, and the value of the time-code / distributed interrupt code is presented on the *auxtimeout[7:0]* signals. The rules that determine whether or not a received time-code / distributed interrupt code will be sent out on the auxiliary interface are the same as for all the other ports.

The AUXTICKIN pin allows for transmission of time-codes and distributed interrupt codes from the auxiliary interface to the other ports. The AUXTICKIN pin can be either synchronous or asynchronous to CLK. In the synchronous mode, a tick is detected each clock cycle that AUXTICKIN is high. In the asynchronous mode, a tick is detected each time a rising edge is seen on AUXTICKIN.

If GPIOSEL is high or *auxtimeinen* is not asserted when a tick i detected, the router's internal time-count value (RTR.TC.TC) is used to send a new time-code out onto the ports. If GPIOSEL is low and *auxtimeinen* is asserted when a tick is detected, the router checks the value of the *auxtimein[7:0]* signals to determine if a time-code or distributed interrupt code should be sent. Note that, in the asynchronous mode, *auxtimein[7:0]* must hold their value for at least two CLK cycles after the rising edge on AUXTICKIN.

The rules that determine whether a distributed interrupt code should be forwarded to the ports are the same as when a distributed interrupt code is received from any other port. However, for time-codes, the value on *auxtimein[7:0]* is always forwarded, and the router's internal time-count value is updated. Note that time-codes / distributed interrupt codes received through the auxiliary interface is not forwarded back out onto the interface itself.

Just as for the router ports, the auxiliary interface has enable / disable bits for time-codes (RTR.RTRCFG.AT) and distributed interrupt codes (RTR.RTRCFG.AI), which need to be set high in order for respective code to be transmitted / received.

6.2.17 SpaceWire-D support

6.2.17.1 Time-code / distributed interrupt code truncation

The router supports truncation of packets when it receives a valid time-code / distributed interrupt code (time-code or distributed interrupt code). A time-code is considered valid when the value equals the internal time count plus one (modulo 64). An distributed interrupt code is considered valid if the corresponding ISR bit is flipped due to reception of the code. The feature can be enabled individually for each port by setting the corresponding RTR.PCTRL.TS bit to 1. A filter, allowing only certain

time-codes / distributed interrupt codes to spill packets, can also be configured individually for each port (see RTR.PCTRL2 register).

If a packet transfer is ongoing when a valid time-code / distributed interrupt code is received, and the code matches the filter in RTR.PCTRL2, the rest of the packet is spilled, and an EEP is written to the FIFO of the output port(s).

Time-code / distributed interrupt code truncation can also be enabled for port 0. In that case, it is the RMAP / SpaceWire Plug-and-Play reply packet that is spilled.

6.2.18 Character and packet counters

Each port, except port 0, has counters for incoming and outgoing characters and packets. For the character counters (RTR.ICHARCNT / RTR.OCHARCNT registers), only SpaceWire data characters are counted (not EOP/EEP). Characters deleted due to header deletion are counted on the incoming port but not on the outgoing port. For the packet counters (RTR.IPKTCNT / RTR.OPKTCNT registers), each EOP/EEP that is preceded by at least one data character is counted as a packet. The counters wrap around, and signal an overflow, when they reach the maximum value. The counters are accessed through the configuration port. See section 6.5.3 for details.

6.2.19 Error detection and reporting

The router can detect and report the following errors:

- SpaceWire link errors: parity error, disconnect error, escape error, credit error (see ECSS-E-ST-50-12C [SPW] for definition of these errors).
- Packet spill due to: timeout (see section 6.2.12), packet length truncation (see section 6.2.13), time-code / distributed interrupt code truncation (see section 6.2.17.1), and spill-if-not-ready feature (see section 6.2.7)
- Invalid address error (see section 6.2.9)
- RMAP errors (see section 6.5.1)
- SpaceWire Plug-and-Play errors (see section 6.5.4)

Each error type has corresponding status bits in respective RTR.PSTS register (RTR.PSTSCFG for the configuration port). Common for all the status bits is that they are set when the error is detected, and stay set until they are cleared manually.

The router also indicates an error in a port by setting the corresponding internal *porterr[19:0]* signal to 1. These signals can be mapped to the GPIO pins, as described in section 5. The *porterr[19:0]* signals stay asserted until all the error bits in the corresponding port's status register are cleared.

Another way the router can report an error is to asserted the IRQ pin. Whether or not the IRQ pin is set when one of the above mentioned errors occur is controlled by the two mask registers, RTR.IMASK and RTR.IPMASK. When the error occur and both the port's corresponding bit in RTR.IPMASK as well as the error type's corresponding bit in RTR.IMASK, are set, then the port's corresponding bit in the Port interrupt pending register (RTR.PIP) is set. The IRQ pin is high as long as any bit in the RTR.PIP register is set.

The router can also be configured to generate a distributed interrupt code when the IRQ pin gets set. See section 6.2.15 for more details.

6.2.20 Setting link-rate for the SpaceWire ports

The initialization divisor register (RTR.IDIV) determines the link-rate during initialization (all states up to and including the connecting-state) for all SpaceWire ports. The register is also used to calculate the link interface FSM timeouts for all SpaceWire ports (6.4 us and 12.8 us, as defined in the SpaceWire standard). The RTR.IDIV register should always be set so that a 10 Mbit/s link-rate is achieved

during initialization. In that case the timeout values will also be calculated correctly. The reset value of the RTR.IDIV.ID field is taken from the SPWCLKDIV[5:0] pins, see Table 36 for details.

To achieve a 10 Mbit/s link-rate, the RTR.IDIV register should be set according to the following formula:

RTR.IDIV = (<frequency in MHz of internal SpaceWire clock> / 10) - 1

The link-rate in run-state can be controlled individually per SpaceWire port with the run-state divisor located in each port's control register (RTR.PCTRL.RD field). The link-rate in run-state is calculated according to the following formula:

<link-rate in Mbits/s> = <frequency in MHz of internal SpaceWire clock> / (RTR.PCTRL.RD+1)

The value in RTR.PCTRL.RD only affects the link-rate in run-state, and does not affect the 6.4 us or 12.8 us timeouts values. The reset value of the RTR.PCTRL.RD field is taken from the SPWCLKDIV[5:0] pins, see Table 27 for details.

6.2.21 SpaceWire transceiver enable signals

There are two dedicated enable pins (SPWEN[18:17]) and sixteen internal enable signals (*spwen[16:1]*), one for each SpaceWire port, intended for controlling external transceivers. Each index corresponds to the port with the same number. SPWEN[18:17] are dedicated pins since the corresponding ports use LVTTL signalling, and the enable signals are usually used for enabling / disabling external LVDS transceivers. The enable signals for the other ports are multiplexed on the GPIO[16:1] pins (see section 5), and are typically used with external LVDS repeaters.

For the LVDS ports, an enable signal is active as long as either the corresponding internal LVDS driver or corresponding internal LVDS receiver is powered on. For the LVTTL ports, an enable signal is active as long as the link interface for the corresponding port is not disabled.

6.2.22 Power saving features

The router supports two power saving features: internal LVDS driver / receiver power down, and clock gating of the ports.

When a SpaceWire port is inactive, the internal LVDS driver is placed in power-down mode. When the internal LVDS driver is in power-down mode, the corresponding SPW_TXDp/n and SPW_TXSp/n pins are tri-stated. A SpaceWire port is considered inactive if one of the following conditions are true:

1. Link interface is disabled, either because RTR.PCTRL.LD is set to 1, or because of the automatic-disconnect feature (see section 6.2.11).
2. Link interface is not disabled, but the transmitter has not been enabled since the last time the link interface was disabled.
3. After reset, until the LVDS driver becomes enabled. The LVDS driver is enabled in the started-state, connecting-state, and run-state.

The LVDS receiver is only set in power-down mode when the link interface is disabled.

To clock gate a SpaceWire port, both the link disable bit (RTR.PCTRL.LD) and the port disable bit (RTR.PCTRL.DI) should be set. The port stays clock gated until either of the two bits are cleared.

To clock gate the SIST port, the SpaceWire In-System Test block must be disabled from the System Level Test Configuration registers (see section 12).

6.3 SpaceWire ports

6.3.1 Overview

The SpaceWire ports have port numbers 1-18. Each SpaceWire port consist of a SpaceWire codec that implements an encoder-decoder compliant to ECSS-E-ST-50-12C [SPW]. The interface to the router's switch matrix consists of a transmit FIFO, receive FIFO, and control and status signals. The transmit and receive FIFOs can both contain 64 N-chars. All the configuration parameters and status information for the ports are accessible through the router's configuration port, either through RMAP or AMBA AHB (see section 6.5.3).

6.3.2 Link-interface FSM

The link-interface FSM controls the link interface (a more detailed description is found in ECSS-E-ST-50-12C [SPW]). The low-level protocol handling (the signal and character level) is handled by the transmitter and receiver while the FSM handles the exchange level.

The link-interface FSM is controlled through the control signals provided in the RTR.PCTRL registers. The link can be disabled through the RTR.PCTRL.LD bit, 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 RTR.PCTRL.LS bit is set, or the link-start-on-request feature described in section 6.2.10 is trying to start the port, or when a NULL character has been received and the RTR.PCTRL.AS bit is set.

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.

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 receive 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 transmit FIFO, and there are credits available. NULLs are sent when no other character transmission is requested, or when the FSM is in a state where no other transmission is allowed.

The credit counter (incoming credits) is automatically increased when FCTs are received, and decreased when N-Chars are transmitted. The credit counter for a SpaceWire port can be read in the corresponding RTR.CREDCNT register.

6.3.3 Transmitter

The state of the FSM, credit counters, possible request to send a time-code / distributed interrupt code, and requests from the transmit 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 / distributed interrupt codes) to be transmitted are presented to the low-level transmitter, which run on the internal SpaceWire clock. For information on how to change the transmission rate, please see section 6.2.20.

6.3.4 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.

Received L-Chars are the handled automatically by link-interface, while all N-Chars are stored in the receive FIFO.

6.4 SIST port

The SIST port has port number 19. It is internally connected to the SpaceWire In-System Test interface (described in section 7) through a FIFO interface. The interface towards the router's switch matrix consists of a transmit FIFO, receive FIFO, and control and status signals. The transmit and receive FIFOs can both contain 64 N-chars. All the configuration parameters and status information for the port is accessible through the router's configuration port, either through RMAP or AMBA AHB (see section 6.5.3).

6.5 Configuration port

The configuration port has port number 0. It consists of an RMAP target, AMBA AHB slave interface, SpaceWire Plug-and-Play interface, and a set of configuration and status registers.

6.5.1 RMAP target

6.5.1.1 Overview

The configuration port's RMAP target implements the RMAP protocol, as defined in the RMAP standard [RMAP]. Verified writes and reads up to 128 B, and read-modify-writes of 4 B (8 B if the mask field is included in the count) are supported.

Replies from the configuration port are always sent to the port they arrived on, regardless of the values of the RMAP command's Initiator Logical Address field, and Reply Address field. This feature is implemented as per the RMAP standard's "implicit partial return address feature" as specified in section 5.1.6.i [RMAP], due to such implementation there is no need to specify in the "Reply SpW Address" field of RMAP commands a byte to route a reply from the configuration port RMAP target (port 0) to the very input port the RMAP command arrived.

The address space of the configuration port is specified in section 6.5.3.

Additional requirements on the RMAP commands imposed by the configuration port's RMAP target are:

- The Target Logical Address field must be 0xFE.
- The Address fields must contain a 4 B aligned address.
- The Extended Address field must be 0x00.
- Key field must be 0x00.
- For write and read commands the Data Length fields must contain a value that is a multiple of 4, ranging from 0 to 128 B.
- For read-modify-write commands the Data Length fields must contain a value of 0 or 8.
- For write commands the Verify Data Before Write bit in the Instruction field must be set to 1.

How the RMAP target handles commands that does not meet the above requirement is detailed in sections 6.5.1.2 and 6.5.1.4.

When an RMAP write command larger than 4 bytes is processed (i.e. more than one register written by the same command), the registers will not change value simultaneously. The command is buffered locally, since only verified write commands are allowed, and then read out from the buffer and written to the registers, with one CLK cycle delay between each register write. This needs to be considered when for example updating large parts of the routing table. Data traffic that arrives while the RMAP command is being processed may or may not be routed according to the new values depending on the address of the packet and how much of the RMAP command that has been processed. Note that the RMAP target is blocked while the RMAP write command is being processed, which means that data returned in an RMAP read command is always either the value before or after the complete RMAP write, never in between.

6.5.1.2 RMAP command support

Table 21 lists all possible RMAP commands and shows how the configuration port's RMAP target handles them. An RMAP command will always have bits 7:6 of the command's Instruction field set to "01", and those bits are therefore left out of the table. Bits 1:0 of the command's Instruction field determines the length of the command's Reply Address Field, and does not affect the action taken, so they have been left out of the table as well. The action taken assumes that no errors were detected in the RMAP packet. For handling of RMAP packet error, see section 6.5.1.4.

Table 21. RMAP command decoding and handling.

Bit 5	Bit 4	Bit 3	Bit 2		
Write / Read	Verify Data Before Write	Reply	Increment Addr	Function	Action taken
0	0	0	0	Invalid	No operation performed. Error code 0x02 is saved in the RTR.PCTRLCFG.EC field. No reply is sent.
0	0	0	1	Invalid	No operation performed. Error code 0x02 is saved in the RTR.PCTRLCFG.EC field. No reply is sent.
0	0	1	0	Read single address	Read operation performed, if the requirements in section 6.5.1.1 are met.
0	0	1	1	Read incrementing address	Read operation performed, if the requirements in section 6.5.1.1 are met.
0	1	0	0	Invalid	No operation performed. Error code 0x02 is saved in the RTR.PCTRLCFG.EC field. No reply is sent.
0	1	0	1	Invalid	No operation performed. Error code 0x02 is saved in the RTR.PCTRLCFG.EC field. No reply is sent.
0	1	1	0	Invalid	No operation performed. Reply is sent with error code 0x02. Error code is also saved in the RTR.PCTRLCFG.EC field.
0	1	1	1	Read-modify-write incrementing address	Read-modify-write operation performed if the requirements in section 6.5.1.1 are met.
1	0	0	0	Write, single address, don't verify before writing, no reply	No operation performed. Error code 0x0A is saved in the RTR.PCTRLCFG.EC field. No reply sent.
1	0	0	1	Write, incrementing address, don't verify before writing, no reply	No operation performed. Error code 0x0A is saved in the RTR.PCTRLCFG.EC field. No reply sent.
1	0	1	0	Write, single address, don't verify before write, send reply	No operation performed. Reply is sent with error code 0x0A. Error code is also saved in the RTR.PCTRLCFG.EC field.
1	0	1	1	Write, incrementing address, don't verify before write, send reply	No operation performed. Reply is sent with error code 0x0A. Error code is also saved in the RTR.PCTRLCFG.EC field.
1	1	0	0	Write, single address, verify before writing, no reply	Write operation performed if the requirements in section 6.5.1.1 are met.

Table 21. RMAP command decoding and handling.

Bit 5	Bit 4	Bit 3	Bit 2		
Write / Read	Verify Data Before Write	Reply	Incre-ment Addr	Function	Action taken
1	1	0	1	Write, incre-menting address, verify before writing, no reply	Write operation performed if the requirements in section 6.5.1.1 are met.
1	1	1	0	Write, single address, verify before writing, send reply	Write operation performed if the requirements in section 6.5.1.1 are met.
1	1	1	1	Write, incre-menting address, verify before writing, send reply	Write operation performed if the requirements in section 6.5.1.1 are met.

6.5.1.3 Access control

After reset / power-up the configuration port's address space can be accessed from all the ports. Configuration port accesses can be individually disabled per port by clearing the corresponding RTR.PCTRL.CE bit. Write commands, and read-modify-write commands to the configuration area can be globally disabled by writing a 0 to the RTR.CFGWE.WE bit.

There is also a CFGLOCK pin which can be used to disable configuration accesses from all ports except port 1 and 2. This signal overrides the setting of the RTR.PCTRL.CE for all ports, enabling configuration port accesses for port 1 and 2, and disabling them for all other ports. The RTR.CFGWE register still affects ports 1 and 2 in this case.

When a correct RMAP command is received but not allowed due to one or more of the access control features being enabled, a reply with Status field set to 0x0A (Authorization failure) is sent (if requested), and the RTR.PSTSCFG.EC field is updated to reflect the error. If a reply is not requested, the RTR.PSTSCFG.EC field is still set. In both cases, the operation is not performed.

6.5.1.4 RMAP Error handling

Table 22 shows the order in which errors in an RMAP command are detected. As soon as an error is detected, the command is discarded. If a reply should be sent, to a command that included an error, the reply is sent as soon as possible after the error is detected. This means that the reply might be sent out before the complete incoming RMAP command has been received. Note that since the complete RMAP command is buffered before it is executed, a command that contains an error is never executed.

Table 22. RMAP target error detection order

Detection Order	**Error type**	**RMAP error code**	**Action taken**
1	Wrong Protocol Identifier	N/A	The RTR.PSTSCFG.PT bit is set in order to indicate that the error occurred. No reply is sent.
2	EOP / EEP before com-pleted header	N/A	The RTR.PSTSCFG.EO / RTR.PSTSCFG.EE bit is set in order to indi-cate that the error occurred. No reply is sent.

Table 22. RMAP target error detection order

Detection Order	Error type	RMAP error code	Action taken
3	Header CRC error	N/A	The RTR.PSTSCFG.HC bit is set in order to indicate that the error occurred. No reply is sent.
4	Unused RMAP packet type	N/A	If the packet type (bit 7:6 of the packet's Instruction field) is "10" or "11" then the bit RTR.PSTSCFG.PT is set. For the value "00" (indicating a reply), no bit in RTR.PSTSCFG is set, since the RMAP standard [RMAP] does not specify that such an event should be recorded.
5	EEP immediately after header	N/A	The RTR.PSTSCFG.EE bit is set in order to indicate that the error occurred. No reply is sent.
6	Unused RMAP command code	0x02	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.
7	Invalid Target Logical Address	0x0C	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.
8	Invalid Key	0x03	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.
9	Verify buffer overrun	0x09	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.
10	RMW data length error	0x0B	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.
11	RMAP command not implemented or not authorized.	0x0A	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.
12	Early EOP / early EEP (not immediately after header)	0x05 / 0x07	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.
13	Invalid Data CRC	0x04	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.
14	EEP	0x07	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.
15	Too much data	0x06	RMAP error code is saved in the RTR.PCTRLCFG.EC field. Reply is sent if the Reply bit in the command's Instruction field was set to 1.

Most of the errors listed in table 22 are errors that only occur in one specific way, and they are also explained in the RMAP standard [RMAP]. Authorization failure (error code 0x0A) is however an exception. All the cases that lead to an authorization failure are listed below:

- A read command's Data Length field exceed 128 B.
- A command's (read, write, or read-modify-write) Address field does not contain a 4 B aligned address.
- The access control features described in section 6.5.1.3 prevented the port from accessing the RMAP target.
- The Address field of a command (read, write, or read-modify-write) contains an address that is outside of the configuration port's memory space.
- The Address field of the command (read, write, or read-modify-write) combined with the Length field would generate an access outside of the configuration port's memory space.
- The Length field of a command (read, write, or rea-modify write) is not a multiple of 4.
- A non-verified write command was received.

6.5.2 AMBA AHB slave interface

The configuration port provides an AMBA AHB slave interface, which makes the whole configuration port's address space accessible from the AHB bus.

The routing table is shared between the ports, RMAP target and AHB slave, so accesses from the AHB slave might be stalled because of accesses from the other sources. The priority order when accessing the routing table, starting from the highest, is: router ports, AHB slave, RMAP target. Note that since the AHB slave has higher priority than the RMAP target, it is possible to read and write to the configuration port's registers in the middle of an RMAP write command. This needs to be considered in order to avoid a mismatch between the expected written value and actual written value.

None of the access control mechanisms mentioned in section 6.5.1.3 have any effect on the AHB slave interface.

6.5.3 Registers

The configuration port's registers listed in this section can be accessed either through the RMAP target, or the AMBA AHB slave interface. Both the RMAP addresses and AMBA addresses are specified in table 23. Registers that exist in several identical copies, corresponding to different addresses or ports, for example the RTR.RTPMAP registers, are only described once. The register layout used is explained in section 2.4.

Table 23. GRSPWROUTER registers

RMAP address	AMBA address	Register name	Acronym
0x00000000	0xFFF20000	RESERVED *	
0x00000004 - 0x0000004C	0xFFF20004 - 0xFFF2004C	Routing table port mapping, physical addresses 1-19	RTR.RTPMAP
0x00000050 - 0x0000007C	0xFFF20050 - 0xFFF2007C	RESERVED *	
0x00000080 - 0x000003FC	0xFFF20080 - 0xFFF203FC	Routing table port mapping, logical addresses 32-255	RTR.RTPMAP
0x00000400	0xFFF20400	RESERVED *	
0x00000404 - 0x0000044C	0xFFF20404 - 0xFFF2044C	Routing table address control, physical addresses 1-19	RTR.RTACTRL
0x00000450 - 0x0000047C	0xFFF20450 - 0xFFF2047C	RESERVED *	
0x00000480 - 0x000007FC	0xFFF20480 - 0xFFF207FC	Routing table address control, logical addresses 32-255	RTR.RTACTRL
0x00000800	0xFFF20800	Port control, port 0 (configuration port)	RTR.PCTRLCFG
0x00000804 - 0x0000084C	0xFFF20804 - 0xFFF2084C	Port control, port 1-19 (SpaceWire ports and SIST port)	RTR.PCTRL
0x00000850 - 0x0000087C	0xFFF20850 - 0xFFF2087C	RESERVED	
0x00000880	0xFFF20880	Port status, port 0 (configuration port)	RTR.PSTSCFG
0x00000884 - 0x000008CC	0xFFF20884 - 0xFFF208CC	Port status, ports 1-19 (SpaceWire ports and SIST port)	RTR.PSTS
0x000008D0 - 0x000008FC	0xFFF208D0 - 0xFFF208FC	RESERVED	
0x00000900 - 0x0000094C	0xFFF20900 - 0xFFF2094C	Port timer reload, ports 0-19	RTR.PTIMER
0x00000950 - 0x0000097C	0xFFF20950 - 0xFFF2097C	RESERVED	
0x00000980	0xFFF20980	Port control 2, ports 0 (configuration port)	RTR.PCTRL2CFG

Table 23. GRSPWROUTER registers

RMAP address	AMBA address	Register name	Acronym
0x00000984 - 0x000009CC	0xFFF20984 - 0xFFF209CC	Port control 2, ports 1-19 (SpaceWire ports and SIST port)	RTR.PCTRL2
0x000009D0 - 0x000009FC	0xFFF209D0 - 0xFFF209FC	RESERVED	
0x00000A00	0xFFF20A00	Router configuration / status	RTR.RTRCFG
0x00000A04	0xFFF20A04	Time-code	RTR.TC
0x00000A08	0xFFF20A08	Version / instance ID	RTR.VER
0x00000A0C	0xFFF20A0C	Initialization divisor	RTR.IDIV
0x00000A10	0xFFF20A10	Configuration write enable	RTR.CFGWE
0x00000A14	0xFFF20A14	Timer prescaler reload	RTR.PRESCALER
0x00000A18	0xFFF20A18	Interrupt mask	RTR.IMASK
0x00000A1C	0xFFF20A1C	Interrupt port mask	RTR.IPMASK
0x00000A20	0xFFF20A20	Port interrupt pending	RTR.PIP
0x00000A24	0xFFF20A24	Interrupt code generation	RTR.ICODEGEN
0x00000A28	0xFFF20A28	Interrupt code distribution ISR, interrupt 0-31	RTR.ISR0
0x00000A2C	0xFFF20A2C	Interrupt code distribution ISR, interrupt 32-63	RTR.ISR1
0x00000A30	0xFFF20A30	Interrupt code distribution ISR timer reload	RTR.ISRTIMER
0x00000A34	0xFFF20A34	Interrupt code distribution ACK-to-INT timer reload	RTR.AITIMER
0x00000A38	0xFFF20A38	Interrupt code distribution ISR change timer reload	RTR.ISRCTIMER
0x00000A3C	0xFFF20A3C	RESERVED	
0x00000A40	0xFFF20A40	SpaceWire link running status	RTR.LRUNSTS
0x00000A44	0xFFF20A44	Capability	RTR.CAP
0x00000A48 - 0x00000A4C	0xFFF20A48 - 0xFFF20A4C	RESERVED	
0x00000A50	0xFFF20A50	SpaceWire Plug-and-Play - Device Vendor and Product ID	RTR.PNPVEND
0x00000A54	0xFFF20A54	SpaceWire Plug-and-Play - Unit Vendor and Product ID	RTR.PNPUVEND
0x00000A58	0xFFF20A58	SpaceWire Plug-and-Play - Unit Serial Number	RTR.PNPUSN
0x00000A5C - 0x00000C0C	0xFFF20A5C - 0xFFF20C0C	RESERVED	
0x00000C10, 0x00000C20 ... 0x00000D30	0xFFF20C10, 0xFFF20C20 ... 0xFFF20D30	Outgoing character counter, ports [1, 2 ... 19]	RTR.OCHARCNT
0x00000C14, 0x00000C24 ... 0x00000D34	0xFFF20C14, 0xFFF20C24 ... 0xFFF20D34	Incoming character counter, ports [1, 2 ... 19]	RTR.ICHARCNT
0x00000C18, 0x00000C28 ... 0x00000D38	0xFFF20C18, 0xFFF20C28 ... 0xFFF20D38	Outgoing packet counter, ports [1, 2 ... 19]	RTR.OPKTCNT
0x00000C1C, 0x00000C2C ... 0x00000D3C	0xFFF20C1C, 0xFFF20C2C ... 0xFFF20D3C	Incoming packet counter, ports [1, 2 ... 19]	RTR.IPKTCNT
0x00000D40 - 0x00000DFC	0xFFF20D40 - 0xFFF20DFC	RESERVED	
0x00000E00 - 0x00000E4C	0xFFF20E00 - 0xFFF20E4C	Maximum packet length, ports 0-19	RTR.MAXPLEN
0x00000E50 - 0x00000E7C	0xFFF20E50 - 0xFFF20E7C	RESERVED	
0x00000E84 - 0x00000EC8	0xFFF20E84 - 0xFFF20EC8	Credit counter, ports 1-18	RTR.CREDCNT

Table 23. GRSPWROUTER registers

RMAP address	AMBA address	Register name	Acronym
0x00000ECC - 0x00000EFC	0xFFF20ECC - 0xFFF20EFC	RESERVED	
0x00000F00	0xFFF20F00	General purpose out, bits 0-31	RTR.GPOA
0x00000F04	0xFFF20F04	General purpose out, bits 32-48	RTR.GPOB
0x00000F08 - 0x00000F0C	0xFFF20F08 - 0xFFF20F0C	RESERVED	
0x00000F10	0xFFF20F10	General purpose in, bits 0-1	RTR.GPIA
0x00000F14 - 0x00000FFC	0xFFF20F14 - 0xFFF20FFC	RESERVED	
0x00001000	0xFFF21000	RESERVED**	
0x00001004 - 0x0000104C	0xFFF21004 - 0xFFF2104C	Routing table, combined port mapping and address control, addresses 1-19	RTR.RTCOMB
0x00001050 - 0x0000107C	0xFFF21050 - 0xFFF2107C	RESERVED**	
0x00001080 - 0x000013FC	0xFFF21080 - 0xFFF213FC	Routing table, combined port mapping and address control, addresses 32-255	RTR.RTCOMB
0x00001400 - 0x00001FFC	0xFFF21400 - 0xFFF21FFC	RESERVED	
0x00002000 - 0x00002FFC	0xFFF22000 - 0xFFF22FFC	APB address area	RTR.APBAREA

* Physical address 0 (configuration port), and physical addresses 20-31 (non existing ports) does not have an RTR.RTP-MAP or RTR.RTACTRL register, and are therefore RESERVED.

** Physical address 0 (configuration port), and physical addresses 20-31 (non existing ports) does not have an RTR.RTCOMB register, and are therefore RESERVED.

Table 24. 0x00000004-0x0000004C, 0x00000080-0x000003FC - RTR.RTPMAP - Routing table port mapping, addresses 1-19 and 32-255

31 20	19 1	0
RESERVED	PE	PD
0x000	*	0
r	rw*	rw

31: 20 RESERVED

19: 1 Port enable bits (PE) - When set to 1, each bit enables packets with the physical / logical address corresponding to this RTR.RTPMAP register to be sent on the port with the same number as the bit index. For physical addresses, the bit index corresponding to the port with the same number as the physical address itself is always 1 (not possible to write to 0). For logical addresses, at least one bit must be set to 1 in order for a packet with the corresponding address to be routed; otherwise the packet is spilled and an invalid address error generated. Reset value for physical addresses is all zeroes (except for the bit index corresponding to the port with the same number as the address). Reset value for logical addresses is zero.

0 Packet distribution (PD) - When set to 1, packet distribution is used for the physical / logical address corresponding to this RTR.RTPMAP register. When set to 0, group adaptive routing is used. See section 6.2.3 and 6.2.4 for more information.

Table 25. 0x00000404-0x0000044C, 0x00000480-0x000007FC - RTR.RTACTRL - Routing table address control, addresses 1-19 and 32-255

31 4	3	2	1	0
RESERVED	SR	EN	PR	HD
0x0000000	*	*	*	*
r	rw	rw	rw	rw

31: 4 RESERVED

3 Spill-if-not-ready (SR) - When set to 1, an incoming packet with the corresponding physical / logical address is immediately spilled if the selected output port's link interface is not in run-state. If packet distribution is used for the incoming packet, and this bit is set, the packet is spilled unless all output ports' link interfaces are in run state. For physical addresses, this bit is double mapped in the RTR.PCTRL.SR field. Reset value for physical addresses are taken from the SPILLIFNOTREADY pin. Reset value for logical addresses is N/R.

2 Enable (EN) - Enables the routing table address control entry. Address control entries for physical addresses are always enabled, and this field is constant 1. For logical addresses, this bit must be set to 1 in order for packets with the corresponding logical address to be routed. Reset value for logical addresses is 0.

1 Priority (PR) - Sets the arbitration priority of this physical / logical address. 0 = low priority, 1 = high priority. Used when more than one packet is competing for the same output port. For physical addresses, this bit is double mapped in the RTR.PCTRL.PR field. Reset value for physical addresses is 0. Reset value for logical addresses is N/R.

0 Header deletion (HD) - Enables / disabled header deletion for the corresponding logical address. For physical addresses, header deletion is always enabled, and this bit is constant 1. Reset value for logical addresses is N/R.

Table 26. 0x00000800 - RTR.PCTRLCFG - Port control, port 0 (configuration port)

31 18	17	16	15 10	9	8 0
RESERVED	PL	TS	RESERVED	TR	RESERVED
0x0000	0	0	0x00	*	0x000
r	rw	rw	r	rw	r

31: 18 RESERVED

17 Packet length truncation (PL) - When set to 1, an RMAP / SpaceWire Plug-and-Play reply is spilled, and an EEP written to the transmit FIFO of the output port, if the total length of the reply packet exceeds the maximum length specified in the RTR.MAXPLEN register for port 0. See section 6.2.13 for more information on packet length truncation.

16 Time-code / distributed interrupt code truncation (TS) - When set to 1, an RMAP / SpaceWire Plug-and-Play reply is spilled, and an EEP written to the transmit FIFO of the output port, if a valid time-code / distributed interrupt code is received and if the code matches the codes selected by the RTR.PCTRL2CFG.SV and RTR.PCTRL2CFG.SM fields. See section 6.2.17 for more information.

15: 10 RESERVED

9 Timer enable (TR) - Enable data character timer for port 0. See section 6.2.12 for details. Reset value set from GPIO[23:0] pins, as described in section 5.

8: 0 RESERVED

Table 27. 0x00000804-0x0000084C - RTR.PCTRL - Port control, ports 1-19 (SpaceWire ports and SIST port)

31 30 29 24	23 22	21	20	19	18	17	16	15	14 11	10	9	8	7	6	5	4	3	2	1	0
RD	RES	ST	SR	AD	LR	PL	TS	IC	ET	RESERVED	DI	TR	PR	TF	RS	TE	R	CE	AS	LS	LD
0x0 *	0x0	0	*	*	*	0	0	*	0	0x0	*	*	0	0	0	1	0	*	1	0	0
rw	r	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw

31: 24 Run-state clock divisor (RD) - Clock divisor value used for the corresponding port's link interface when in run-state. Field is only available for the SpaceWire ports. Bits 31:30 have reset value 0x0, while bits 29:24 get their reset value from the SPWCLKDIV[5:0] pins. For more information about setting the link-rate for SpaceWire ports during run-state see section 6.2.20.

23: 22 RESERVED

21 Static routing enable (ST) - When set to 1, incoming packets on this port are routed based on the physical address specified in the corresponding RTR.PCTRL2.SD field, and the setting of the corresponding RTR.PCTRL2.SC bit, instead of the packet's first byte. Header deletion is not used when static routing is enabled, which means that the first byte of the packet is always sent as well. This bit can only be set to 1 if the RTR.RTRCFG.SR bit is set to 1. Note that when this bit is set to 1 it is not possible to access the configuration port from this port.

20 Spill-if-not-ready (SR) - This bit is double mapping of the RTR.RTACTRL.SR bit. See table 25.

19 Auto-disconnect (AD) - When set to 1, the auto-disconnect feature described in section 6.2.11 is enabled. Reset value taken from the AUTODCONNECT pin. This bit is only available for the SpaceWire ports.

18 Link-start-on-request (LR) - When set to 1, the link-start-on-request feature described in section 6.2.10 is enabled. Reset value taken from the LINKSTARTONREQ pin. This bit is only available for the SpaceWire ports.

17 Packet length truncation (PL) - When set to 1, packets for which this port is the input port will be spilled, and an EEP written to the transmit FIFO of the output port(s), if the packets exceed the maximum length specified in the corresponding RTR.MAXPLEN register. See section 6.2.13 for more information on packet length truncation.

16 Time-code / distributed interrupt code truncation (TS) - When set to 1, packets for which this port is the input port will be spilled, and an EEP written to the transmit FIFO of the output port(s), if a valid time-code / distributed interrupt code is received, and if the code also matches the codes selected by the RTR.PCTRL2.SV and RTR.PCTRL2.SM fields. See section 6.2.17 for more information.

Table 27. 0x00000804-0x0000084C - RTR.PCTRL - Port control, ports 1-19 (SpaceWire ports and SIST port)

15	Distributed interrupt code enable (IC) - When set to 0, all incoming distributed interrupt codes on this port are discarded, and no distributed interrupt codes are sent out on the port. When set to 1, the four bits RTR.PCTRL2.IR, RTR.PCTRL2.IT, RTR.PCTRL2.AR, RTR.PCTRL2.AT are used to enable / disable distributed interrupt code transmit and receive. Note that the global distributed interrupt code enable bit, RTRTCFG.IE, also must be set to 1 for distributed interrupt codes to be sent / received. See section 6.2.15 for a description of distributed interrupts. Reset value set from GPIO[23:0] pins, as described in section 5.
14	Enable external time (ET) - When a time-code is received on the port and this bit is set to 0, the router discards the received time-code value and instead increments its internal time-counter value (RTR.TC.TC), and forwards a time-code with the new value to the other ports. If this bit is set to 1 when the time-code is received, the time-code is processed according to the rules described in section 6.2.14. This bit is only available for the SIST port.
13: 11	RESERVED
10	Disable port (DI) - When set to 1, data transfers to and from this port are disabled. See section 6.2.5 for details. Reset value is 0 for SpaceWire ports, and 1 for SIST port. NOTE: If this bit is set in combination with the RTR.PCTRL.LD bit, the link interface for this port will be clock gated, and the LVDS drivers for this port will be powered-down.
9	Packet timer enable (TR) - Enable the data character timer for incoming packets. See section 6.2.12 for details. Reset value set from GPIO[23:0] pins, as described in sections 5.
8	Priority (PR) - This bit is double mapping of the RTR.RTACTRL.PR bit. See table 25.
7	Transmit FIFO reset (TF) - Resets the transmit FIFO on this port. This means that the FIFO is emptied (counters and pointers set to 0), and an EEP is written to the FIFO to ensure that any incomplete packet is detected by the receiver. If a packet transmission is active (another port is using this port as output port) when this bit is set, the remainder of that packet will be spilled before the EEP is inserted. This bit is self-clearing, and should not be written with 0 while it is 1, since that could abort the ongoing transmit FIFO reset.
6	Receive FIFO spill (RS) - Spills the receive FIFO for this port, meaning that the packet currently being received is spilled. The output port(s) used for the packet will have an EEP written to the transmit FIFO to indicate that the packet was ended prematurely. If no packet is received, setting this bit has no effect. This bit is self-clearing, and should not be written with 0 while it is 1, since that could abort the ongoing receive FIFO spill.
5	Time-code enable (TE) - Enables time-codes to be received and transmitted on this port. When set to 1, received time-codes are processed according to the rules described in section 6.2.14. If this bit is set to 0, all received time-codes on this port are ignored.
4	RESERVED
3	Configuration port access enable (CE) - Enable accesses to the configuration port from this port. If set to 0, incoming packets with physical address 0 will be spilled. Reset value is 1 for SpaceWire ports, and 0 for SIST port.
2	Autostart (AS) - Enable the link interface FSM's autostart feature, as defined in ECSS-E-ST-50-12C [SPW]. This bit is only available for the SpaceWire ports.
1	Link start (LS) - Start the link interface FSM. This bit is only available for the SpaceWire ports.
0	Link disabled (LD) - Disable the link interface FSM. This bit is only available for the SpaceWire ports. NOTE: When this bit is set to 1, the LVDS driver for this SpaceWire port will be powered down. If this bit is set in combination with the RTR.PCTR.DI bit, the link interface will be clock gated.

Table 28. 0x00000880 - RTR.PSTSCFG - Port status, port 0 (configuration port)

31	30	29	28	27	26	25	24	23 20	19	18	17 12	11 7	6 5	4	3 0
EO	EE	PL	TT	PT	HC	PI	CE	EC	R	TS	RESERVED	IP	RES	CP	PC
0	0	0	0	0	0	0	0	0x0	0	0	0x00	0x0	0x0		0x0
wc	wc	wc	wc	wc	wc	wc	rw*	r	r	wc	r	r	r	rw*	r

31 Early EOP (EO) - Set to 1 when an RMAP / SpaceWire Plug-and-Play command with an early EOP was received by the configuration port. See section 6.5.1.4 for error detection order.

30 Early EEP (EE) - Set to one when an RMAP / SpaceWire Plug-and-Play command with an early EEP was received by the configuration port. See section 6.5.1.4 for error detection order.

29 Packet length truncation (PL) - Set to 1 when an RMAP / SpaceWire Plug-and-Play reply packet has been spilled due to a maximum length violation. See section 6.2.13 for details.

28 Time-code / distributed interrupt code tick truncation (TT) - Set to one when an RMAP / SpaceWire Plug-and-Play reply packet has been spilled due to a time-code / distributed interrupt code. See section 6.2.17 for details.

27 Packet type error (PT) - Set to one if an RMAP / SpaceWire Plug-and-Play packet with correct header CRC, but with the packet type bits set to the reserved values “10” or “11”, was received by the configuration port. See section 6.5.1.4 for error detection order.

26 Header CRC Error (HC) - Set to one if a Header CRC error is detected in an RMAP / SpaceWire Plug-and-Play command received by the configuration port. See section 6.5.1.4 for error detection order.

25 Protocol ID Error (PI) - Set to one if a packet received by the configuration port had the wrong protocol ID. Supported protocol ID:s are 0x01 (RMAP), and 0x03 (SpaceWire Plug-and-Play). See section 6.5.1.4 for error detection order.

24 Clear error code (CE) - Write with a 1 to clear the RTR.PCTRLCFG.EC field. This bit is self clearing and always reads 0. Writing 0 has no effect.

23: 20 Error code (EC) - Shows the four least significant bits of the latest non-zero RMAP status code. If zero, no error has occurred.

19 RESERVED

18 Timeout spill (TS) - Set to one when an RMAP reply was spilled due to a packet timeout. See section 6.2.12 for details.

17: 12 RESERVED

11: 7 Input port (IP) - The number of the last port from which a packet was routed to the configuration port. This field is updated even if an operation is not performed, for example due to an incorrect RMAP packet.

6: 5 RESERVED

4 Clear SpaceWire Plug-and-Play error code (CP) - Write with a 1 to clear the RTR.PCTRLCFG.PC field. This bit is self clearing and always reads 0. Writing 0 has no effect.

3: 0 SpaceWire Plug-and-Play Error code (PC) - Shows the four least significant bits of the latest non-zero SpaceWire Plug-and-Play status code. If zero, no error has occurred.

Table 29. 0x00000884-0x000008CC - RTR.PSTS - Port status, ports 1-19 (SpaceWire ports and SIST port)

31 30	29	28	27	26	25 23	22	21	20	19	18	17	16	15	14 12	11 7	6	5	4	3	2	1	0
PT	PL	TT	RS	SR	RESERVED	LR	SP	AC	R	TS	R	TF	RE	LS	IP	PR	PB	IA	CE	ER	DE	PE
*	0	0	0	0	0x0	0	0	0	0	0	0	0	1	000	00000	0	0	0	0	0	0	0
r	wc	wc	wc	wc	r	r	r	r	r	wc	r	r	r	r	r	r	r	wc	wc	wc	wc	wc

31: 30 Port type (PT) - The type of this port. Constant value of “00” for the SpaceWire ports, and constant value of “11” for the SIST port.

29 Packet length truncation (PL) - Set to 1 when a packet for which this port was the input port has been spilled due to the packet length truncation feature. See section 6.2.13 for details.

28 Time-code / distributed interrupt code tick truncation (TT) - Set to 1 when a packet for which this port was the input port has been spilled due to the time-code / distributed interrupt code truncation feature. See section 6.2.17 for details.

Table 29. 0x00000884-0x000008CC - RTR.PSTS - Port status, ports 1-19 (SpaceWire ports and SIST port)

27	RMAP / SpaceWire Plug-and-Play spill (RS) - Set to 1 when an RMAP / SpaceWire Plug-and-Play command received on this port was spilled by the configuration port.
26	Spill-if-not-ready spill (SR) - Set to 1 when a packet received on this port was spilled due to the spill-if-not-ready feature. See section 6.2.7.
25: 23	RESERVED
22	Link-start-on-request status (LR) - Set to 1 when this port either was started, or currently is trying to start, due to the link-start-on-request feature, described in section 6.2.10. This bit is only available for the SpaceWire ports.
21	Spill status (SP) - This bit is 1 when a packet that is incoming on this port currently is being spilled. Otherwise, this bit is 0.
20	Active status (AC) - Set to 1 when a packet arrives at this port and the port has been given access to the routing table. Cleared when the packet has been transmitted or spilled.
19	RESERVED
18	Timeout spill (TS) - Set to 1 when a packet for which this port was the input port was spilled due to a packet timeout. See section 6.2.12 for details.
17	RESERVED
16	Transmit FIFO full (TF) - Set to 1 when the transmit FIFO on this port is full.
15	Receive FIFO empty (RE) - Set to 1 when the receive FIFO on this port is empty.
14: 12	Link state (LS) - Current link state. 000 = Error reset. 001 = Error wait, 010 = Ready, 011 = Started, 100 = Connecting, 101 = Run state. This field is only available for the SpaceWire ports.
11: 7	Input port (IP) - This field shows the number of the input port for either the currently ongoing packet transfer on this port (if RTR.PSTS.PB = 1), or for the last packet transfer on this port (if RTR.PSTS.PB = 0).
6	Port receive busy (PR) - Set to 1 when this port is the input port of an ongoing packet transfer.
5	Port transmit busy (PB) - Set to 1 when this port is the output port of an ongoing packet transfer.
4	Invalid address (IA) - Set to 1 when an invalid address error occurred on this port. See section 6.2.9 for details.
3	Credit error (CE) - Set to 1 when a credit error has occurred. This bit is only available for the SpaceWire ports.
2	Escape error (ER) - Set to 1 when an escape error has occurred. This bit is only available for the SpaceWire ports.
1	Disconnect error (DE) - Set to 1 when a disconnect error has occurred. This bit is only available for the SpaceWire ports.
0	Parity error (PE) - Set to 1 when a parity error has occurred on. This bit is only available for the SpaceWire ports.

Table 30. 0x00000900-0x0000094C - RTR.PTIMER - Port timer reload, ports 0-19

31 10	9 0
RESERVED	RL
0x000000	*
r	rw*

31: 10	RESERVED
9: 0	Timer reload (RL) - Port timer reload value, counted in prescaler ticks. This value is used to reload the corresponding port timer used for packet transfer timeouts, and auto-disconnect. The minimum value of this field is 1. Trying to write 0 will result in 1 being written. Reset value set from GPIO[23:0] pins, as described in sections 5.

Table 31. 0x00000980 - RTR.PCTRL2CFG - Port control 2, port 0 (configuration port)

31 24	23 16	15	14 13 12 11 10 9 8 6 5 1	0
SM	SV	OR	RESERVED	
0xC0	0x00	1	0x0000	
rw	rw	rw	r	

31: 24 Time-code / distributed interrupt code truncation mask (SM) - Defines which bits of a time-code / distributed interrupt code that must match the value specified in RTR.PCTRL2CFG.SV in order for an RMAP / SpaceWire Plug-and-Play reply packet to be spilled. If a bit in this field is set to 1, the corresponding bit in RTR.PCTRL2.SV must match the time-code / distributed interrupt code. If a bit in this field is set to 0, the corresponding bit in RTR.PCTRL2.SV does not have to match the time-code / distributed interrupt code.

23: 16 Time-code / distributed interrupt code truncation value (SV) - Defines the value to use together with the RTR.PCTRL2CFG.SM field when checking if a received time-code / distributed interrupt code should spill an ongoing RMAP / SpaceWire Plug-and-Play reply.

15 Overrun timeout enable (OR) - Enables spilling due to overrun timeouts for RMAP / SpaceWire Plug-and-Play replies. See section 6.2.12 for details.

14: 0 RESERVED

Table 32. 0x00000984-0x000009CC - RTR.PCTRL2 - Port control 2, ports 1-19 (SpaceWire ports and SIST port)

31 24	23 16	15	14	13	12	11	10	9	8 6	5 1	0
SM	SV	OR	UR	R	AT	AR	IT	IR	RESERVED	SD	SC
0xC0	0x00	1	1	0	1	1	1	1	0x0	0x00	0
rw	rw	rw	rw	r	rw	rw	rw	rw	r	rw	rw

31: 24 Time-code / distributed interrupt code truncation mask (SM) - Defines which bits of a time-code / distributed interrupt code that must match the value specified in RTR.PCTRL2.SV in order for a packet, for which this port is the input port, to be spilled. If a bit in this field is set to 1, the corresponding bit in RTR.PCTRL2.SV must match the time-code / distributed interrupt code. If a bit in this field is set to 0, the corresponding bit in RTR.PCTRL2.SV does not have to match the time-code / distributed interrupt code.

23: 16 Time-code / distributed interrupt code truncation value (SV) - Defines the value to use together with the RTR.PCTRL2.SM field when checking if a time-code / distributed interrupt code should spill a packet for which this port is the input port.

15 Overrun timeout enable (OR) - Enables spilling due to overrun timeouts for packets for which this port is the input port. See section 6.2.12 for details.

14 Underrun timeout enable (UR) - Enables spilling due to unerrun timeouts for packets for which this port is the input port. See section 6.2.12 for details.

13 RESERVED

12 Interrupt acknowledgement code / extended interrupt code transmit enable (AT) - Enables the transmission of interrupt acknowledgement codes / extended interrupt codes on this port. If set to 0, no interrupt acknowledgement codes / extended interrupt codes will be forwarded to this port.

11 Interrupt acknowledgement code / extended interrupt code receive enable (AR) - Enabled the reception of interrupt acknowledgement codes / extended interrupt codes on this port. If set to 0, all received interrupt acknowledgement codes / extended interrupt codes on this port will be silently discarded.

10 Interrupt code transmit enable (IT) - Enables the transmission of interrupt codes on this port. If set to 0, no interrupt codes will be forwarded to this port.

9 Interrupt code receive enable (IR) - Enabled the reception of interrupt codes on this port. If set to 0, all received interrupt codes on this port will be silently discarded.

8: 6 RESERVED

5: 1 Static route destination (SD) - When RTR.PCTRL.ST is set to 1, incoming packets on this port will be routed based on the value of this field, and the setting of RTR.PCTRL2.SC, instead of the packet's first byte.

0 Static route configuration (SC) - When this bit is set to 1, the RTR.RTPMAP register corresponding to the physical address specified by the RTR.PCTRL2.SD field will be used when routing packets, if RTR.PCTRL.ST is set to 1.

Table 33. 0x00000A00 - RTR.RTRCFG - Router configuration / status

31 27	26 22	21 17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SP	RESERVED	FP	R	SR	PE	IC	IS	IP	AI	AT	IE	RE	EE	AA	SA	TF	R	TA	PP
0x12	0x00	0x01	0	*	*	0	0	0	*	1	*	0	*	*	1	*	0	1	1
r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	r	r	r

31: 27 SpaceWire ports (SP) - Set to the number of SpaceWire ports in the router. Constant value of 0x12.

26: 22 RESERVED

21: 17 FIFO ports (FP) - Set to the number of FIFO ports + SIST ports in the router. Constant value of 0x01.

16 RESERVED

15 Static routing enable (SR) - This read-only bit specifies if the router's static routing feature is enabled (1) or disabled (0). See section 6.2.6 for details. The value is set from the STATICROUTEEN pin at reset.

14 SpaceWire Plug-and-Play enable (PE) - This read-only bit specifies if the router's SpaceWire Plug-and-Play features are enabled (1) or disabled (0). See section 6.5.4 for details. The value is set from the PNPEN pin at reset.

13 ISR change timer enable (IC) - If set to 1, the router will wait for the time period specified by the RTR.IRCTIMER register after an ISR bit change value, before it allows an incoming distributed interrupt code to change the value of the same ISR bit. If set to 0, the ISR change timers are not used, and an ISR bit is allowed to change value again as soon as the previous distributed interrupt code has been distributed.

12 Distributed interrupt code selection routine (IS) - If set to 0, the router uses round-robin on the interrupt numbers when deciding which distributed interrupt code to distribute next. If set to 1, the router gives priority to lower interrupt numbers when deciding which distributed interrupt code to distribute. See section 6.2.15.1.

11 Distributed interrupt code priority (IP) - When set to 0, all interrupt codes have priority over all interrupt acknowledgement codes / extended interrupt codes, and will be distributed first. When set to 1, all interrupt acknowledgement codes have priority over all interrupt codes. See section 6.2.15.1.

10 Auxiliary distributed interrupt codes enable (AI) - If set to 1, distributed interrupt codes can be sent and received on the auxiliary time-code / distributed interrupt code interface. If set to 0, all distributed interrupt codes received on the auxiliary interface are silently discarded, and no distributed interrupt codes will be transmitted on the interface. Reset value set from GPIO[23:0] pins, as described in section 5.

9 Auxiliary time-code enable (AT) - If set to 1, time-codes can be sent and received on the auxiliary time-code / distributed interrupt code interface. If set to 0, all time-codes received on the auxiliary interface are silently discarded, and no time-codes will be transmitted on the interface.

8 Distributed interrupt codes enable (IE) - Global enable/disable for distributed interrupt codes. If set to 0, all received distributed interrupt codes will either be silently discarded (if RTRCFG.TF = 1), or handled as time-codes (if RTRCFG.TF = 0). When set to 1, whether or not distributed interrupt codes are received or transmitted on a port depends on the setting of the register bits RTR.PCTRL.IC, RTR.PCTRL2.IR, RTR.PCTRL2.IT, RTR.PCTRL2.AR, and RTR.PCTRL2.AT. Reset value taken from GPIO[23:0] pins, as described in section 5.

7 Reset (RE) - Resets the complete router when written with a 1. When this bit is written through RMAP, an RMAP reply will not be sent, even if the reply bit in the RMAP commands Instruction field is set to 1. This bit is self-clearing.

6 Enable extended distributed interrupts (EE) - If set to 0, all distributed interrupt codes with bit 5 set to 1 are handled as interrupt acknowledgement codes. If set to 1, all distributed interrupt codes with bit 5 set to 1 are handled as extended interrupt code. Reset value taken from GPIO[23:0] pins, as described in section 5. See section 6.2.15.

5 Asynchronous auxiliary interface (AA) - This read-only bit specifies whether the inputs of the auxiliary time-code / distributed interrupt code interface, described in section 6.2.16, are handled as synchronous or asynchronous to CLK. A value of 0 means that the inputs are handles as synchronous to CLK, while 1 means asynchronous. Reset value is set through GPIO[23:0] pins, as specified in section 5.

4 Self addressing enable (SA) - If set to 1, ports are allowed to send packets to themselves. If set to 0, packets with the same input port as output port are spilled, and an invalid address error is asserted for that port.

3 Time-code control flag mode (TF) - When set to 0, all received time-codes / distributed interrupt codes are handled as time-codes, no matter the value of the control flags (bits 7:6 of the code). When set to 1, the time-code control flags must have value "00" to be considered valid time-codes. Note that the RTRCFG.IE bit has priority over this bit, which means that if RTRCFG.IE is 1, then setting this bit to 0 has no impact. Reset value taken from GPIO[23:0] pins, as specified in section 5.

2 RESERVED

Table 33. 0x00000A00 - RTR.RTRCFG - Router configuration / status

1 Timers available (TA) - Constant value 1. Indicates that the router has support for timers, as described in section 6.2.12.

0 SpaceWire Plug and Play available (PP) - Constant value 1. Indicates that the router support SpaceWire Plug and Play, as described in section 6.5.4.

Table 34. 0x00000A04 - RTR.TC - Time-code

31 10	9	8	7 6	5 0
RESERVED	RE	EN	CF	TC
0x000000	0	*	0x0	0x00
r	rw*	rw	r	r

31: 10 RESERVED

9 Reset time-code (RE) - When this field is written to 1, the RTR.TC.CF and RTR.TC.TC fields are reset. This bit is self-clearing, and always reads 0. Writing 0 has no effect.

8 Enable time-codes (EN) - When set to 1, received time-codes are handled by the router according to the rules described in 6.2.14. When set to 0, all received time-codes are silently discarded. Reset value set through GPIO[23:0] pins, as specified in section 5.

7: 6 Time-control flags (CF) - The current value of the router's time-code control flags (bits 7:6 of the latest valid time-code received).

5: 0 Time-counter (TC) - Current value of the router's time counter.

Table 35. 0x00000A08 - RTR.VER - Version / instance ID

31 24	23 16	15 8	7 4	3 0
MA	MI	PA	ID	
0x01	0x03	0x00	0x0	*
r	r	r	rw	

31: 24 Major version (MA) - Holds the major version number of the router. Constant value 0x01.

23: 16 Minor version (MI) - Holds the minor version number of the router. Constant value 0x03.

15: 8 Patch (PA) - Holds the patch number of the router. Constant value 0x00.

7: 0 Instance ID (ID) - Holds the instance ID number of the router. Reset value for bits 3:0 is set through GPIO[23:0] pins, as specified in section 5.

Table 36. 0x00000A0C - RTR.IDIV - Initialization divisor

31 8	7 0
RESERVED	ID
0x000000	*
r	rw

31: 8 RESERVED

7: 0 Initialization clock divisor (ID) - Clock divisor value used by all the SpaceWire links to generate the 10 Mbit/s rate during initialization. Bits 7:6 have reset value 0x0, while bits 5:0 get their reset value from the SPWCLKDIV[5:0] pins. For more information about setting the link-rate for SpaceWire ports during initialization see section 6.2.20.

Table 37. 0x00000A10 - RTR.CFGWE - Configuration port write enable

31 … 1	0
RESERVED	WE
0x00000000	1
r	rw

31: 1 RESERVED

0 Configuration port write enable (WE) - When set to 1, write accesses to the configuration port area are allowed. When set to 0, write accesses are only allowed to this register. RMAP write and RMAP read-modify-write commands will be replied to with the Status field set to 0x0A (authorization failure), if a reply was requested. The value of this bit has no effect for SpaceWire Plug-and-Play commands.

Table 38. 0x00000A14 - RTR.PRESCALER - Timer prescaler reload

31 … 16	15 … 0
RESERVED	RL
0x0000	*
r	rw*

31: 16 RESERVED

15: 0 Timer prescaler reload (RL) - Global prescaler reload value used for generating a common tick for the data character timers, auto-disconnect timers, and distributed interrupt code timers. The prescaler runs on the system clock, and a tick is generated every RTR.PRESCALER.RL+1 CLK cycle. The minimum value of this field is 49. Trying to write a value less than that will result in 49 being written. Reset value is set through GPIO[23:0] pins, as described in section 5.

Table 39. 0x00000A18 - RTR.IMASK - Interrupt mask

31 … 11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PE	SR	RS	TT	PL	TS	AC	RE	IA	LE	R
0x00000	0	0	0	0	0	0	0	0	0	0	0
r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r

31: 11 RESERVED

10 SpaceWire Plug-and-Play error (PE) - Generate an interrupt when a SpaceWire Plug and Play error has been detected in the configuration port. The different errors are described in 6.5.4.

9 Spill-if-not-ready (SR) - Generate an interrupt when a packet has been spilled because of the spill-if-not-ready feature described in section 6.2.7.

8 Run-state entry (RS) - Generate an interrupt when a SpaceWire link enters run-state.

7 Time-code / distributed interruptcode tick truncation (TT) - Generate an interrupt when a packet has been spilled because of the time-code / distributed interrupt code truncation feature described in section 6.2.17.1.

6 Packet length truncation (PL) - Generate an interrupt when a packet has been spilled due to the packet length truncation feature described in section 6.2.13.

5 Timeout spill (TS) - Generate an interrupt when a packet has been spilled due to the timeout mechanism.

4 Auxiliary configuration port error (AC) - Generate an interrupt when either a header CRC error, protocol ID error, packet type error, early EOP, or early EEP has been detected in the configuration port.

3 RMAP error (RE) - Generate an interrupt when an error has been detected in the configuration port for an RMAP command such that the PSTS.EC field is set to a non-zero value.

2 Invalid address (IA)- Generate an interrupt when an invalid address error has occurred on a port. See RTR.PSTS:IA bit and section 6.2.9 for a definition of invalid address.

1 Link error (LE) - Generate an interrupt when a link error has been detected on a SpaceWire port.

0 RESERVED

Table 40. 0x00000A1C - RTR.IPMASK - Interrupt port mask

31 20	19 0
RESERVED	IE
0x000	0x00000
r	rw

31: 20 RESERVED

19: 0 Port interrupt enable (IE) - Set a bit to 1 to enable interrupts to be generated for an error detected in the port with the same number as the bit index. An interrupt is signaled through the IRQ pin, and optionally through a distributed interrupt code.

Table 41. 0x00000A20 - RTR.PIP - Port interrupt pending

31 20	19 0
RESERVED	IP
0x000	0x00000
r	wc

31: 20 RESERVED

19: 0 Interrupt pending (IP) - When a bit is set to 1, the port with the same number as the bit index was the source of an interrupt. A bit in this field will only be set to 1 for a generated interrupt if the port's corresponding bit in RTR.IPMASK is set, as well as the error types corresponding bit in RTR.IMASK, are set.

Table 42. 0x00000A24 - RTR.ICODEGEN - Interrupt code generation

31 21	20	19	18	17	16	15 6	5 0
RESERVED	UA	AH	IT	TE	EN	RESERVED	IN
0x000	0	0	0	1	0	0x000	0x00
r	rw	rw	rw	rw	rw	r	rw

0

31: 21 RESERVED

20 Interrupt code generation un-acknowledge mode (UA) - If this bit is set to 1, an ISR timeout for a distributed interrupt that was generated by the router will clear the bits in the RTR.PIP register that were set when the interrupt was generated. If this bit is set to 0, no extra handling is done on an ISR timeout event, and the bits in RTR.PIP will stay set. See section 6.2.15.

19 Interrupt acknowledgement code handling (AH) - When set to 1, and the router has generated an interrupt code, a received interrupt acknowledgement code with the interrupt number matching the RTR.ICODEGEN.IN field will clear the bits in the RTR.PIP register that were set when the interrupt code was generated. If set to 0, no extra handling of a received interrupt acknowledgement code is done and the bits in RTR.PIP will stay set. This bit is unused when the distributed interrupts are operating in the extended interrupt mode. See section 6.2.15.

18 Interrupt type (IT) - 0 = Level. 1 = Edge. When set to 0, a new interrupt code is distributed as long as RTR.PIP register is non zero. When set to 1, a new interrupt code is distributed only when a bit in RTR.PIP toggles from 0 to 1. See section 6.2.15.

Table 42. 0x00000A24 - RTR.ICODEGEN - Interrupt code generation

17	Interrupt acknowledgement code to interrupt code timer enable (TE) - If set to 1, the router will wait for the time period specified by the RTR.AITIMER register after the reception of an interrupt acknowledgement code (for which the router generated the corresponding interrupt code) until a new interrupt code is allowed to be generated. If set to 0, the timer is not used, and a new interrupt code is allowed to be generated as soon as the interrupt acknowledgement code has been distributed. This bit is unused when the distributed interrupts are operating in the extended interrupt mode.
16	Interrupt code generation enable (EN) - When 1, distributed interrupt code generation is enabled, and an interrupt code / extended interrupt code can be generated when an internal error event occurs. See section 6.2.15.
15: 6	RESERVED
5: 0	Interrupt number (IN) - Sets the interrupt number of the distributed interrupt code that will be generated when the interrupt code generation feature is enabled (RTR.ICODEGEN.EN = 1). Note that when the distributed interrupts are operating in interrupt with acknowledgements mode, this field must not be set to a value larger than 31, since that would specify an interrupt with acknowledgement code. See section 6.2.15.

Table 43. 0x00000A28 - RTR.ISR0 - Interrupt code distribution ISR register, interrupt 0-31

31	0
IB	
0x00000000	
wc	

31: 0	Distributed interrupt code ISR bits (IB) - The current value of the distributed interrupt code ISR register for interrupt numbers 0 to 31. Each bit index corresponds to the ISR bit value for the corresponding interrupt number. A bit value of 1 indicates that an interrupt code with the corresponding interrupt number has been received, but not yet acknowledged. A bit value of 0 indicates either that no interrupt code with the corresponding interrupt number has been received, or that the previous interrupt code was either acknowledged or timed out. This register should be normally only be used for diagnostics and / or FDIR.

Table 44. 0x00000A2C - RTR.ISR1 - Interrupt code distribution ISR register, interrupt 32-63

31	0
IB	
0x00000000	
wc	

31: 0	Distributed interrupt code ISR bits (IB) - The current value of the distributed interrupt code ISR register for interrupt numbers 32 to 63. Each bit index + 32 corresponds to the ISR bit value for the corresponding interrupt number. A bit value of 1 indicates that an extended interrupt code with the corresponding interrupt number has been received. A bit value of 0 indicates either that no extended interrupt code with the corresponding interrupt number has been received, or that the previous interrupt cod has timed out. Note that if the distributed interrupts are operating in interrupt with acknowledgements mode, this register is unused. This register should be normally only be used for diagnostics and / or FDIR.

Table 45. 0x00000A30 - RTR.ISRTIMER - Interrupt code distribution ISR timer reload

31 10	9 0
RESERVED	RL
0x00000	*
r	rw

31: 10 RESERVED

9: 0 Interrupt code distribution ISR timer reload (RL) - Interrupt code distribution ISR timer reload value, counted in prescaler ticks. Each ISR bit has its own timer, which is started and reloaded with the value of this field when an interrupt code / extended interrupt code with the corresponding interrupt number is received (or generated by the router). Reset value is set through GPIO[23:0] pins, as specified in section 5. See section 6.2.15 for details on interrupt code distribution.

Table 46. 0x00000A34 - RTR.AITIMER - Interrupt code distribution ACK-to-INT timer reload

31 10	9 0
RESERVED	RL
0x00000	*
r	rw

31: 10 RESERVED

9: 0 Interrupt acknowledgement code to interrupt code timer reload (RL) - Interrupt acknowledgement code to interrupt code timer reload value, counted in prescaler ticks. When an interrupt acknowledgement code is received - for which the router generated the corresponding interrupt code - the interrupt acknowledgement code to interrupt code timer is started and reloaded with the value of this field. Reset value is set through GPIO[23:0] pins, as specified in section 5. This register is unused when the distributed interrupts are operating in the extended interrupt mode. See section 6.2.15 for details on interrupt code distribution.

Table 47. 0x00000A38 - RTR.ISRCTIMER - Interrupt code distribution ISR change timer reload

31 10	9 0
RESERVED	RL
0x00000	0
r	rw

31: 5 RESERVED

9: 0 Interrupt code distribution ISR change timer reload (RL) - Interrupt code distribution ISR change timer reload value, counted in prescaler ticks. Each time an ISR bit change value, the corresponding ISR change timer is started and reloaded with the value of this field. See section 6.2.15 for details on interrupt code distribution.

Table 48. 0x00000A40 - RTR.LRUNSTAT - Link running status

31 19	18 1	0
RESERVED	LR	R
0x0000	0x00000	0
r	r	r

31: 19 RESERVED

18: 1 Link running status (LR)- Each bit is set to 1 when the link interface for the SpaceWire port with the same number as the bit index is in run-state. If the link interface is not in run-state, the bit is set to 0.

0 RESERVED

Table 49. 0x00000A44 - RTR.CAP - Capability

31 26	25 24 23	22 20	19	18 16	15	14	13	12	11	10	9 5	4 0
RESERVED		PF	R	RM	R	AA	AX	R	ID	SD	PC	CC
0x000		0x2	0	0x5	0	1	1	0	1	1	0x1F	0x1F
r		r	r	r	r	r	r	r	r	r	r	r

31: 23 RESERVED

22: 20 Port N-char FIFO size (PF) - The number of entries in the port FIFOs can be determined by the value of this field, according to the formula: Entries = $2^{(RTR.CAP.PF+4)}$. Constant value of 0x2 = 64 entries.

19 RESERVED

18: 16 RMAP maximum data length (RM) - This field specifies the maximum data length in an RMAP read / write command that the configuration port can handle. The length can be determined according to the formula: Length = $2^{(RTR.CAP.RM+2)}$. Constant value of 0x5 = 128 bytes.

15 RESERVED

14 Asynchronous axiliary time-code / distributed interrupt code support (AA) - Specifies that the router has support for the auxiliary time-code / distributed interrupt code interface inputs to be asynchronous to CLK. See section 6.2.14. Constant value of 1.

13 Auxiliary time-code / distributed interrupt code support (AX) - Specifies that the router has support for the auxiliary time-code / distributed interrupt code feature described in 6.2.14. Constant value of 1.

12 RESERVED

11 Distributed interrupt code support (ID) - Specifies that the router has support for the interrupt distribution scheme, described in 6.2.15. Constant value of 1.

10 SpaceWire-D support (SD) - Specifies that the router has support for the SpaceWire-D, described in section 6.2.17. Constant value of 1.

9: 5 Port packet counter bits (PC) - Specifies the number of bits in the port's incoming / outgoing packet counters. Constant value of 0x1F = 31 bits

4: 0 Port character counter bits (CC) - Specifies the number of bits in the port's incoming / outgoing character counters. Constant value of 0x1F = 31 bits

Table 50. 0x00000A50 - RTR.PNPVEND - SpaceWire Plug-and-Play - Device Vendor and Product ID

31 16	15 0
VI	PI
0x0003	0x0718
r	r

31: 16 SpaceWire Plug-and-Play Vendor ID (VI) - Double mapping of the VEND bits from the SpaceWire Plug-and-Play Device Vendor and Product ID field. See table 67.

25: 0 SpaceWire Plug-and-Play Product ID (PI) - Double mapping of the PROD bits from the SpaceWire Plug-and-Play Device Vendor and Product ID field. See table 67.

Table 51. 0x00000A54 - RTR.PNPUVEND - SpaceWire Plug-and-Play - Unit Vendor and Product ID

31 16	15 0
VI	PI
0x0000	0x0000
rw	rw

31: 16 SpaceWire Plug-and-Play Unit vendor ID (VI) - Double mapping of the VEND bits from the SpaceWire Plug-and-Play Unit Vendor and Product ID field (see table 76).

25: 0 SpaceWire Plug-and-Play Unit product ID (PI) - Double mapping of the PROD bits from the SpaceWire Plug-and-Play Unit Vendor and Product ID field (see table 76).

Table 52. 0x00000A58 - RTR.PNPUSN - SpaceWire Plug-and-Play - Unit Serial Number

31	4	3	0
SN			
0x000000		*	
rw			

31: 0 SpaceWire Plug-and-Play Unit serial number (SN) - Double mapping of the SpaceWire Plug-and-Play Unit Serial Number field (see table 77). Reset value for bits 3:0 is set through GPIO[23:0] pins, as specified in section 5.

Table 53. 0x00000C10,0x00000C20...0x00000D30 - RTR.OCHARCNT - Outgoing character counter, ports 1-19

31	30	0
OR	CC	
0	0x00000000	
wc	rw*	

31 Counter overrun (OR) - This bit is set to 1 when the character counter (RTR.OCHARCNT.CC) overflows. A write with a 1 to this field will clear the whole character counter (including this bit)

30: 0 Character counter (CC) - Number of data characters (EOP, EEP, time-codes, distributed interrupt codes are not included) that have been transmitted on the corresponding port. When the counter reaches its maximum value, it sets the RTR.OCHARCNT.OR bit to 1 and continue counting from zero. A write to this field where bit 30 is set to 1 will reset the RTR.OCHARCNT.OR bit. A write to this field where bit 30 is set to 0 has no effect.

Table 54. 0x00000C14,0x00000C24...0x00000D34 - RTR.ICHARCNT - Incoming character counter, ports 1-19

31	30	0
OR	CC	
0	0x00000000	
wc	rw*	

31 Counter overrun (OR) - This bit is set to 1 when the character counter (RTR.ICHARCNT.CC) overflows. A write with a 1 to this field will the whole character counter (including this bit)

30: 0 Character counter (CC) - Number of data characters (EOP, EEP, time-codes, distributed interrupt codes are not included) that have been received on the corresponding port. When the counter reaches its maximum value, it sets the RTR.ICHARCNT.OR bit to 1 and continue counting from zero. A write to this field where bit 30 is set to 1 will reset the RTR.ICHARCNT.OR bit. A write to this field where bit 30 is set to 0 has no effect.

Table 55. 0x00000C18,0x00000C28...0x00000D38 - RTR.OPKTCNT - Outgoing packet counter, ports 1-19

31	30	29	0
OR	PC		
0	0x00000000		
wc	rw*		

31 Counter overrun (OR) - This bit is set to 1 when the packet counter (RTR.OPKTCNT.PC) overflows. A write with a 1 to this field will reset the whole character counter (including this bit).

30: 0 Packet counter (PC) - Number of packets that have been transmitted on the corresponding port. When the counter reaches its maximum value, it sets the RTR.OPKTCNT.OR bit to 1 and continue counting from zero. A write to this field where bit 30 is set to 1 will reset the RTR.OPKTCNT.OR bit. A write to this field where bit 30 is set to 0 has no effect.

Table 56. 0x00000C1C,0x00000C2C...0x00000D3C - RTR.IPKTCNT - Incoming packet counter, ports 1-19

31	30 29	0
OR	PC	
0	0x00000000	
wc	rw*	

31 Counter overrun (OR) - This bit is set to 1 when the packet counter (RTR.IPKTCNT.PC) overflows. A write with a 1 to this field will reset the whole character counter (including this bit).

30: 0 Packet counter (PC) - Number of packets that have been received on the corresponding port. When the counter reaches its maximum value, it sets the RTR.IPKTCNT.OR bit to 1 and continue counting from zero. A write to this field where bit 30 is set to 1 will reset the RTR.IPKTCNT.OR bit. A write to this field where bit 30 is set to 0 has no effect.

Table 57. 0x00000E00-0x00000E4C - RTR.MAXPLEN - Maximum packet length, ports 0-19

31 25	24 0
RESERVED	ML
0x00	0x000000
r	rw

31: 25 RESERVED

24: 0 Maximum packet length (ML) - Maximum length of packets for which the corresponding port is the input port. This field is only used when the RTR.PCTRL.PL bit (RTR.PCTRLCFG.PL for port 0) is set to 1. See section 6.2.17 for details.

Table 58. 0x00000E84-0x00000EC8 - RTR.CREDCNT - Credit counter, ports 1-18

31 12	11 6	5 0
RESERVED	OC	IC
0x00000	0	0
r	r	r

31: 12 RESERVED

11: 6 Out credit counter (OC) - Number of outgoing credits. For each credit, the other end of the link is allowed to send one N-Char.

5: 0 In credit counter (IC) - Number of incoming credits. For each credit, the port is allowed to transmit one N-Char.

Table 59. 0x00000F00 - RTR.GPOA - General purpose out, bits 0-31

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
GPIO15		GPIO14		GPIO13		GPIO12		GPIO11		GPIO10		GPIO9		GPIO8		GPIO7		GPIO6		GPIO5		GPIO4		GPIO3		GPIO2		GPIO1		GPIO0	
0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0	
rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw		rw	

31: 0 GPIO pin config (GPIOxx) - Each two bit field controls which function that is mapped to the corresponding GPIO pin. See section 5 for information on how the setting of this register affect the GPIO[23:0] pins.

Table 60. 0x00000F04 - RTR.GPOB - General purpose out, bits 32-48

31 17	16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
RESERVED	CLL	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
0x0000	0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
r	wc	rw	rw	rw	rw	rw	rw	rw	rw

31: 17 RESERVED

16 Clear lost lock (CLL) - If this bit is written to 1, the RTR.GPIA.LL bit is cleared. This bit is self clearing, and always reads 0.

15: 0 GPIO pin config (GPIOxx) - Each two bit field controls which function that is mapped to the corresponding GPIO pin. See section 5 for information on how the setting of this register affects the GPIO[23:0] pins.

Table 61. 0x00000F10 - RTR.GPIA - General purpose in, bits 0-1

31 17	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2	1	0
RESERVED		LL	CL
0x00000000		0	N/R
r		r	r

31: 2 RESERVED

1 Lost lock (LL) - This bit is a sticky bit that indicates if the lock bit from the SpaceWire clock PLL has gone low. This bit can be cleared by writing a 1 to the RTR.GPOB.CLL bit.

0 SpaceWire clock lock (CL) - Shows the current value of the SpaceWire clock PLL lock output.

Table 62. 0x00001004-0x000013FC - RTR.RTCOMB - Routing table, combined port mapping and address control, addresses 1-255

31	30	29	28	27 20	19 1	0
SR	EN	PR	HD	RESERVED	PE	PD
N/R	0	N/R	N/R	0x00	N/R	N/R
rw	rw	rw	rw	r	rw	rw

31 Spill-if-not-ready (SR) - This bit is a double mapping of the RTR.RTACTRL.SR bit. See table 25.

30 Enable (EN) - This bit is a double mapping of the RTR.RTACTRL.EN bit. See table 25.

29 Priority (PR) - This bit is a double mapping of the RTR.RTACTRL.PR bit. See table 25.

28 Header deletion (HD) - This bit is a double mapping of the RTR.RTACTRL.HD bit. See table 25.

27: 20 RESERVED

19: 1 Port enable bits (PE) - This field is a double mapping of the RTR.RTPMAP.PE field. See table 24.

0 Packet distribution (PD) - This field is a double mapping of the RTR.RTPMAP.PD field. See table 24.

NOTE: See note for RTR.RTPMAP (table 24).

Table 63. 0x00002000-0x00002FFC - RTR.APBAREA - APB address area

31 0
APB
N/A
rw

31: 0 The registers for the SPI controller and General purpose I/O interface are accessible through this register area. See table 8 for details of how the registers are mapped.

6.5.4 SpaceWire Plug-and-Play interface

The configuration port supports parts of the SpaceWire Plug-and-Play protocol described in [SPW-PNP]. The supported fields are listed in table 66, and explained in more detail in tables 67 through 81. The layout used is explained in section 2.4.

During reset, the PNPEN pin works as a global enable / disable pin for the SpaceWire Plug-and-Play support. Support is enabled if the pin is high during reset, and disabled if the pin is low during reset. The RTR.RTRCFG.PE bit shows if SpaceWire Plug-and-Play is supported or not.

The SpaceWire Plug-and-Play protocol uses standard RMAP commands and replies with the same requirements as presented in section 6.5.1, but with the following differences:

- Protocol Identifier field of a command shall be set to 0x03.
- A command's address fields shall contain a word address. The SpaceWire Plug-and-Play addresses are encoded as shown in table 64.
- The increment bit in the command's instruction field shall be set to 1, otherwise a reply with Status field set to 0x0A (authorization failure) is sent.
- RMAP Read-modify-write command is replaced by a compare-and-swap operation. The command's data fields shall contain the new data to be written, while the mask fields shall contain the value that the current data must match in order for the new data to be written. If there is a mismatch, a reply with Status field set to 0x0A (authorization failure) is sent.
- The reply packet's Status field can contain the additional status codes described in table 65.

Table 64. SpaceWire Plug-and-Play address encoding

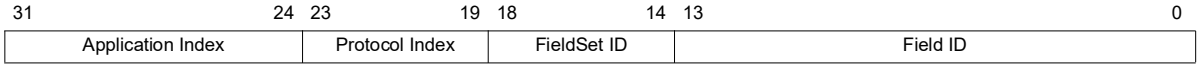


Table 65. SpaceWire Plug-and-Play status codes

Value	Description
0xF0	Unauthorized access - A write, or compare-and-swap command arrived either when the router was not configured (Device ID field = 0), or the command did not match the owner information saved in the Link Information field and Owner Address fields.
0xF1	Reserved field set - A read, write, or compare-and-swap command's address field points to a non existing field set.
0xF2	Read-only field - A write, or compare-and-swap command's address points to a read-only field.
0xF3	Compare-and-swap-only-field - A write command's address points to a compare-and-swap-only field.

Note that it is not possible to access the SpaceWire Plug-and-Play fields through the AHB slave interface, except for the fields that are double mapped into the configuration port's address space (see section 6.5.3).

An access (read, write, or compare-and-swap) made either to a field outside the Device Information service, or to a field in an undefined field set within the Device Information service, will generate a reply with the Status field set to 0xF1. An access (read, write, or compare-and-swap) to an undefined or unsupported field in one of the defined field sets, within the Device Information service, is not treated as an error, and the Status field of the reply will be 0x00. Possible write-data for such an access is discarded, and possible read-data returned is always 0.

Please reference the [SPWPNP] for additional details to what is presented in this section.

Table 66. SpaceWire Plug-and-Play support

SpW PnP Address	Register name	Acronym	Service - Field set - Field
0x00000000	SpaceWire Plug-and-Play - Device Vendor and Product ID	RTR.PNPVEND	Device Information - Device Identification - Device Vendor and Product ID
0x00000001	SpaceWire Plug-and-Play - Version	RTR.PNPVER	Device Information - Device Identification - Version
0x00000002	SpaceWire Plug-and-Play - Device Status	RTR.PNPDEVSTS	Device Information - Device Identification - Device Status
0x00000003	SpaceWire Plug-and-Play - Active Links	RTR.PNPACTLNK	Device Information - Device Identification - Active Links
0x00000004	SpaceWire Plug-and-Play - Link Information	RTR.PNPLNKINFO	Device Information - Device Identification - Link Information
0x00000005	SpaceWire Plug-and-Play - Owner Address 0	RTR.PNPOA0	Device Information - Device Identification - Owner Address 0
0x00000006	SpaceWire Plug-and-Play - Owner Address 1	RTR.PNPOA1	Device Information - Device Identification - Owner Address 1
0x00000007	SpaceWire Plug-and-Play - Owner Address 2	RTR.PNPOA2	Device Information - Device Identification - Owner Address 2
0x00000008	SpaceWire Plug-and-Play - Device ID	RTR.PNPDEVID	Device Information - Device Identification - Device ID
0x00000009	SpaceWire Plug-and-Play - Unit Vendor and Product ID	RTR.PNPUVEND	Device Information - Device Identification - Unit Vendor and Product ID
0x0000000A	SpaceWire Plug-and-Play - Unit Serial Number	RTR.PNPUSN	Device Information - Device Identification - Unit Serial Number
0x00004000	SpaceWire Plug-and-Play - Vendor String Length	RTR.PNPVSTRL	Device Information - Vendor / Product String - Vendor String Length
0x00006000	SpaceWire Plug-and-Play - Product String Length	RTR.PNPPSTRL	Device Information - Vendor / Product String - Product String Length
0x00008000	SpaceWire Plug-and-Play - Protocol Count	RTR.PNPPCNT	Device Information - Protocol Support - Protocol Count
0x0000C000	SpaceWire Plug-and-Play - Application Count	RTR.PNPACNT	Device Information - Application Support- Application Count

Table 67. 0x00000000 - RTR.PNPVEND - SpaceWire Plug-and-Play - Device Vendor and Product ID

31 16	15 0
VEND	PROD
0x0003	0x0718
r	r

31: 16 Vendor ID (VEND) - SpaceWire vendor ID assigned to Frontgrade Gaisler. Constant value of 0x0003.

15: 0 Product ID (PROD) - Product ID assigned to the Frontgrade Gaisler 18x SpaceWire Router. Constant value of 0x0718

Table 68. 0x00000001 - RTR.PNPVER - SpaceWire Plug-and-Play - Version

31 24	23 16	15 8	7 0
MAJOR	MINOR	PATCH	RESERVED
0x01	0x03	0x00	0x00
r	r	r	r

31: 24 Major version number (MAJOR) - Constant value of 0x01.

23: 16 Minor version number (MINOR) - Constant value of 0x03.

15: 8 Patch / Build number (PATCH) - Constant value of 0x00.

7: 0 RESERVED

Table 69. 0x00000002 - RTR.PNPDEVSTS - SpaceWire Plug-and-Play - Device Status

31 8	7 0
RESERVED	STATUS
0x000000	0x00
r	r

31: 8 RESERVED

7: 0 Device status (STATUS) - Constant value of 0x00.

Table 70. 0x00000003 - RTR.PNPACTLNK - SpaceWire Plug-and-Play - Active Links

31 20	19 1	0
RESERVED	ACTIVE	R
0x000	0x00000	0
r	r	r

31: 20 RESERVED

19: 1 Link active (ACTIVE) - If set to 1, the port with the same number as the bit index is running. If set to 0, the port is not running. For the SpaceWire ports (ports 1-18), the corresponding bit will be set to 1 if the link interface is in run-state and the port is not disabled through the Port Control register (RTR.PCTRL.DI = 0). For the SIST port (port 19), the bit is set to 1 if RTR.PCTRL.DI = 0.

0 RESERVED

Table 71. 0x00000004 - RTR.PNPLNKINFO -SpaceWire Plug-and-Play - Link Information

31 24	23 22	21	20 16	15 13	12 8	7	6	5	4 0
OLA	OAL	R	OL	RES	RL	T	U	R	LC
0x00	0x0	0	0x0	0x0	0x0	1	0	0	0x13
r	r	r	r	r	r	r	r	r	r

31: 24 Owner logical address (OLA) - Shows the value of the Initiator Logical Address field from the last successful compare-and-swap command that set the Device ID field.

23: 22 Owner address length (OAL) - Shows how many of the three Owner Address fields that contain valid data.

21 RESERVED

20: 16 Owner link (OL) - Shows the number of the port which was used for the last successful operation to set the value of the Device ID field.

15: 13 RESERVED

Table 71. 0x00000004 - RTR.PNPLNKINFO -SpaceWire Plug-and-Play - Link Information

12: 8	Return link (RL) - Shows the number of the port through which the reply to the current read command will be transmitted.
7	Device type (T) - Constant value of 1, indicating that this device is a router.
6	Unit information (U) - Indicates if the unit identification information (Unit Vendor and Product ID field, and Unit Serial Number field) are valid. 0 = invalid, 1 = valid. This bit will be 0 after reset / power-up. Once the Unit Vendor and Product ID field has been written with a non-zero value, this bit will be set to 1.
5	RESERVED
4: 0	Link count (LC) - Shows the number of router ports. Constant value of 0x13.

Table 72. 0x00000005 - RTR.PNPOA0 - SpaceWire Plug-and-Play - Owner Address 0

31	0
RA	
0x00000000	
r	

31: 0	Reply address (RA) - Shows byte 0-3 of the Reply Address from the last successful compare-and-swap command that set to the Device ID field. If there was no Reply Address, then this field is zero.

Table 73. 0x00000006 - RTR.PNPOA1 - SpaceWire Plug-and-Play - Owner Address 1

31	0
RA	
0x00000000	
r	

31: 0	Reply address (RA) - Shows byte 4-7 of the Reply Address from the last successful compare-and-swap command that set to the Device ID field. If the Reply Address was four bytes or less, then this field is zero.

Table 74. 0x00000007 - RTR.PNPOA2 - SpaceWire Plug-and-Play - Owner Address 2

31	0
RA	
0x00000000	
r	

31: 0 Reply address (RA) - Shows byte 8-11 of the Reply Address from the last successful compare-and-swap command that set to the Device ID field. If the Reply Address was eight bytes or less, then this field is zero.

Table 75. 0x00000008 - RTR.PNPDEVID - SpaceWire Plug-and-Play - Device ID

31	0
DID	
0x00000000	
cas	

31: 0 Device ID (DID) - Shows the device identifier. After reset / power-up, or when this field is written to zero, the router is not considered to have an owner. The same applies to the case when the port indicated by the OL bits in the Link Information field is either disconnected, or disabled by setting the RTR.PCTRL.DI bit to 1. This field is only writable through a compare-and-swap operation.

Table 76. 0x00000009 - RTR.PNPUVEND - SpaceWire Plug-and-Play - Unit Vendor and Product ID

31	16	15	0
VEND		PROD	
0x0000		0x0000	
r		r	

31: 16 Unit vendor ID (VEND) - Shows the unit vendor identifier. This field is read-only through the SpaceWire Plug-and-Play protocol, however it is writable through RMAP and AHB (see section6.5.3). When this field, ot the PROD field, is written with a non-zero value, the U bit in the Link Information field is set to 1.

15: 0 Unit product ID (VEND) - Shows the unit product identifier. This field is read-only through the SpaceWire Plug-and-Play protocol, however it is writable through RMAP and AHB (see section 6.5.3). When this field, or the VEND field, is written with a non-zero value, the U bit in the Link Information field is set to 1.

Table 77. 0x0000000A - RTR.PNPUSN - SpaceWire Plug-and-Play - Unit Serial Number

31	0
USN	
0x00000000	
r	

31: 0 Unit serial number (USN) - Shows the unit serial number. This field is read-only through the SpaceWire Plug-and-Play protocol, however it is writable through RMAP and AHB (see section 6.5.3).

Table 78. 0x00004000 - RTR.PNPVSTRL - SpaceWire Plug-and-Play - Vendor String Length

31 15	14 0
RESERVED	LEN
0x00000	0x0000
r	r

31: 15 RESERVED

14: 0 Vendor string length (LEN) - Constant value of 0, indicating that no vendor string is present.

Table 79. 0x00006000 - RTR.PNPPSTRL - SpaceWire Plug-and-Play - Product String Length

31 15	14 0
RESERVED	LEN
0x00000	0x0000
r	r

31: 15 RESERVED

14: 0 Product string length (LEN) - Constant value of 0, indicating that no product string is present.

Table 80. 0x00008000 - RTR.PNPPCNT - SpaceWire Plug-and-Play - Protocol Count

31 5	4 0
RESERVED	PC
0x0000000	0x00
r	r

31: 5 RESERVED

4: 0 Protocol count (PC) - Constant value of 0, indicating that no protocols can be managed by using SpaceWire Plug-and-Play.

Table 81. 0x0000C000 - RTR.PNPACNT - SpaceWire Plug-and-Play - Application Count

31 8	7 0
RESERVED	AC
0x000000	0x00
r	r

31: 8 RESERVED

7: 0 Application count (AC) - Constant value of 0, indicating that no applications can be managed by using SpaceWire Plug-and-Play.

7 SpaceWire In-System Test

7.1 Overview

The SpaceWire In-System Test (SIST) implements a test generator and a test checker for the verification of SpaceWire links, nodes and routers. It can be used as a built-in self test in a SpaceWire router.

The test generator is provided for the verification of the SpaceWire router and codec functionality. The SpaceWire In-System Test (SIST) protocol provides a means for verifying larger part of the designs' functionality, without the need to generate high speed test patterns and observe results at high frequencies. The SIST module is connected to SpaceWire router via an internal FIFO port. The other side of the SIST module is connected to the AMBA APB bus, and can be controlled through the UART and JTAG interfaces. The SIST module can generate and send SpaceWire packets via the SpaceWire router FIFO port. It can also receive SpaceWire packets via the FIFO port and check there contents. The SpaceWire packets are generated deterministically and can therefore also be easily checked on reception.

The SIST allows direct access to the switch matrix of the router. Characters can directly be written to and read from the FIFO interface connected to the switch matrix, by means of register accesses. This can only be done while a test sequence is not active. Note that a test sequence can be individually enabled for the transmitter and the receiver part. This allows for example automatic test sequence generation, but user controlled test sequence verification.

The SIST module provides the following external and internal interfaces:

- SpaceWire router FIFO port interface
- AMBA APB slave interface

The operation of the module is highly programmable by means of control registers.

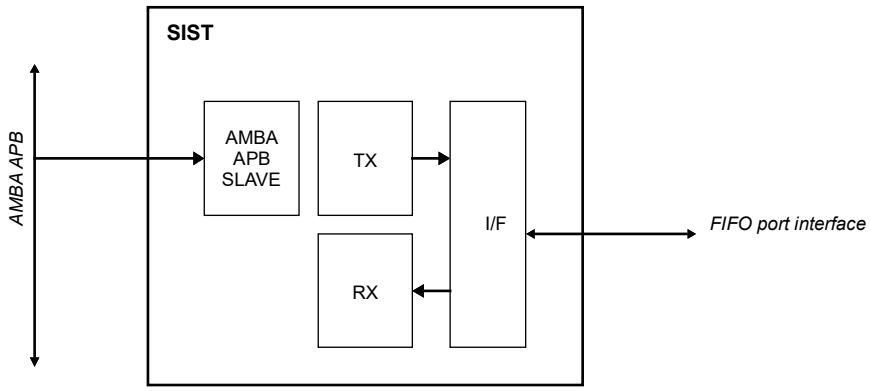


Figure 6. Block diagram

7.2 Operation

7.2.1 Packet format

The packet format is similar to the commands defined for the RMAP protocol [RMAP]:

- SpW Address (0 to 31 bytes), programmable through the SIST.ADDRx registers.
- Logical Address (1 byte), programmable through the SIST.ADDR0 register.
- Protocol ID (1 byte), programmable through the SIST.PID field
- Transaction Identifier (2 bytes) (i.e. seed), programmable through the SIST.SEED register
- Data Length (3 bytes), programmable through the SIST.LEN register

- Header CRC (1 byte as per [RMAP], covering header from Logical Address, inclusive)
- Data (0 to 16 MiB-1) (data is a pseudo-random generated bit string based on the seed)
- Data CRC (1 byte as per [RMAP], covering all Data bytes)
- End-Of-Packet

Packets of up to 2**24 bytes can be generated and checked. Sequences of up to 2**16 packets can be generated, or auto repeat can be enabled. The data is generated by means of a 16-bit wide LFSR, with a programmable polynomial. The state of the LFSR (a.k.a. seed) at the beginning of the data in the packet is transmitted as part of the packet header, allowing each packet to be checked independently. The seed can also be used to detect dropped packets. The length of the packet data field is sent in the packet header. The only managed parameter is the polynomial; everything else can be derived from the packet header.

Packets are automatically generated in an initiator, the contents of a packet is deterministic. Packets are automatically checked in a target when received, providing statistics. The initiator and target are normally the same end-point in a SpaceWire network, but may be different.

The interval between packets is the same as between data characters. The interval width for the two is programmable through the SIST.CTRL.RATE field.

The packet follows the SpaceWire protocol identification [SPWID] format. The SpW Address bytes can be used for path addressing or regional local addressing in a SpaceWire network.

7.2.2 Usage

It is possible to combine the SIST functionality with the internal loop-back function (see section 12), or with external cables looping back the SpaceWire signals per port or between pair of ports.

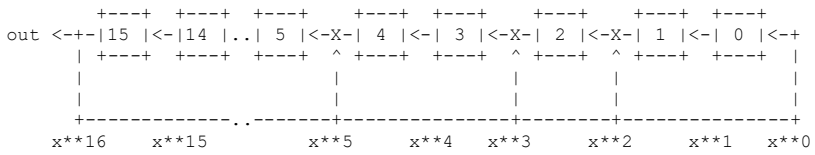
The SIST module also allows direct data read and write to the FIFO port, as well as sending and receiving signaling codes (time-codes and distributed interrupt codes).

The SIST functionality is protected by means of a protected general on/off register (protection done by expected fixed pattern in data).

7.2.3 Data generation

The data field contents are generated from a Linear Feedback Shift Register (LFSR) that is shifted 8 times for each data byte. The implementation uses a Galois version of LFSR (reverse). See figure 7 for a graphical representation. The generator polynomial is: $g(x) = x^{16} + x^5 + x^3 + x^2 + x^0$

Figure 7. Linear Feedback Shift Register (LFSR)



The SIST.PID.POLY field defines which of the feedback points are to be connected in the LFSR. The leftmost SIST.PID.POLY bit (i.e. MSb) corresponds to term x**15 which is represented as delay element with index 15 in figure 7, and the rightmost SIST.PID.POLY bit (i.e. LSb) corresponds to term x**0 which is represented as delay element with index 0 in figure 7. The term x**16 is always implemented. The LFSR is pre-initialized from the SIST.SEED.TXSEED field for the transmitter, and the SIST.SEED.RXSEED field for the receiver, using the same bit association as for SIST.PID.POLY above.

7.3 Registers

The SIST is programmed through registers mapped into AMBA address space. The register layout used is explained in section 2.4.

Table 82. SIST registers

AMBA address	Register	Acronym
0xFFE00200 - 0xFFE0021C	SpaceWire Address Register 0-7	SIST.ADDR0 - SIST.ADDR7
0xFFE00220	Protocol ID and Polynomial Register	SIST.PID
0xFFE00224	Seed Register	SIST.SEED
0xFFE00228	Packet Length Register	SIST.LEN
0xFFE0022C	Control Register	SIST.CTRL
0xFFE00230	Error Register 0	SIST.ERROR0
0xFFE00234	Error Register 1	SIST.ERROR1
0xFFE00238	Error Register 2	SIST.ERROR2
0xFFE0023C	Packet Counter Register	SIST.PKTCNTR
0xFFE00240	Timer Register 0	SIST.TIMER0
0xFFE00244	Timer Register 1	SIST.TIMER1
0xFFE00248	Status Register	SIST.STAT
0xFFE0024C	State Register	SIST.STATE
0xFFE00250	Transmitter Byte Count Register	SIST.TXBYTECNTR
0xFFE00254	Receiver Byte Count Register	SIST.RXBYTECNTR
0xFFE00258	Time-Code Register	SIST.TIME
0xFFE0025C	Protection Register	SIST.PROT
0xFFE00280 - 0xFFE0029C (multiple mapping)	Data Input Registers 0-7	SIST.DIN
0xFFE002A0 - 0xFFE002BC (multiple mapping)	Data Output Registers 0-7	SIST.DOUT

Table 83. 0xFFE00200 - SIST.ADDR0 - SpaceWire Address Register 0

31 24	23 16	15 8	7 0
LA	SPWADDR1	SPWADDR2	SPWADDR3
0xFE	0x00	0x00	0x00
rw	rw	rw	rw

31: 24 Logical Address (LA) - First address byte

23: 16 SpaceWire address (SPWADDR1) - Second address byte

15: 8 SpaceWire address (SPWADDR2) - Third address byte

7: 0 SpaceWire address (SPWADDR3) - Fourth address byte

Table 84. 0xFFE00204 - SIST.ADDR1 - SpaceWire Address Register 1

31 24	23 16	15 8	7 0
SPWADDR4	SPWADDR5	SPWADDR6	SPWADDR7
0x00	0x00	0x00	0x00
rw	rw	rw	rw

31: 24 SpaceWire address (SPWADDR4) - Fifth address byte
23: 16 SpaceWire address (SPWADDR5) - Sixth address byte
15: 8 SpaceWire address (SPWADDR6) - Seventh address byte
7: 0 SpaceWire address (SPWADDR7) - Eighth address byte

Table 85. 0xFFE00208 - SIST.ADDR2 - SpaceWire Address Register 2

31 24	23 16	15 8	7 0
SPWADDR8	SPWADDR9	SPWADDR10	SPWADDR11
0x00	0x00	0x00	0x00
rw	rw	rw	rw

31: 24 SpaceWire address (SPWADDR8) - Ninth address byte
23: 16 SpaceWire address (SPWADDR9) - Tenth address byte
15: 8 SpaceWire address (SPWADDR10) - Eleventh address byte
7: 0 SpaceWire address (SPWADDR11) - Twelfth address byte

Table 86. 0xFFE0020C - SIST.ADDR3 - SpaceWire Address Register 3

31 24	23 16	15 8	7 0
SPWADDR12	SPWADDR13	SPWADDR14	SPWADDR15
0x00	0x00	0x00	0x00
rw	rw	rw	rw

31: 24 SpaceWire address (SPWADDR12) - Thirteenth address byte
23: 16 SpaceWire address (SPWADDR13) - Fourteenth address byte
15: 8 SpaceWire address (SPWADDR14) - Fifteenth address byte
7: 0 SpaceWire address (SPWADDR15) - Sixteenth address byte

Table 87. 0xFFE00210 - SIST.ADDR4 - SpaceWire Address Register 4

31 24	23 16	15 8	7 0
SPWADDR16	SPWADDR17	SPWADDR18	SPWADDR19
0x00	0x00	0x00	0x00
rw	rw	rw	rw

31: 24 SpaceWire address (SPWADDR16) - Seventeenth address byte
23: 16 SpaceWire address (SPWADDR17) - Eigtheenth address byte
15: 8 SpaceWire address (SPWADDR18) - Nineteenth address byte
7: 0 SpaceWire address (SPWADDR19) - Twentieth address byte

Table 88. 0xFFE00214 - SIST.ADDR5 - SpaceWire Address Register 5

31 24	23 16	15 8	7 0
SPWADDR20	SPWADDR21	SPWADDR22	SPWADDR23
0x00	0x00	0x00	0x00
rw	rw	rw	rw

31: 24 SpaceWire address (SPWADDR20) - Twenty-first address byte
23: 16 SpaceWire address (SPWADDR21) - Twenty-second address byte
15: 8 SpaceWire address (SPWADDR22) - Twenty-third address byte
7: 0 SpaceWire address (SPWADDR23) - Twenty-fourth address byte

Table 89. 0xFFE00218 - SIST.ADDR6 - SpaceWire Address Register 6

31 24	23 16	15 8	7 0
SPWADDR24	SPWADDR25	SPWADDR26	SPWADDR27
0x00	0x00	0x00	0x00
rw	rw	rw	rw

31: 24 SpaceWire address (SPWADDR24) - Twenty-fifth address byte
23: 16 SpaceWire address (SPWADDR25) - Twenty-sixth address byte
15: 8 SpaceWire address (SPWADDR26) - Twenty-seventh address byte
7: 0 SpaceWire address (SPWADDR27) - Twenty-eighth address byte

Table 90. 0xFFE0021C - SIST.ADDR7 - SpaceWire Address Register 7

31 24	23 16	15 8	7 0
SPWADDR28	SPWADDR29	SPWADDR30	SPWADDR31
0x00	0x00	0x00	0x00
rw	rw	rw	rw

31: 24 SpaceWire address (SPWADDR28) - Twenty-ninth address byte
23: 16 SpaceWire address (SPWADDR29) - Thirtieth address byte
15: 8 SpaceWire address (SPWADDR30) - Thirty-first address byte
7: 0 SpaceWire address (SPWADDR31) - Thirty-second address byte

Table 91. 0xFFE00220 - SIST.PID - Protocol ID and Polynomial Register

31 16	15 8	7 0
POLY	RESERVED	PID
0x002D	0x00	0xF0
rw	r	rw

31: 16 Polynomial (POLY) - Reset value corresponds to g(x) = x**16 + x**5 + x**3 + x**2 + x**0
15: 8 RESERVED
7: 0 Protocol Identifier (PID)

Table 92. 0xFFE00224 - SIST.SEED - Seed Register

31 16	15 0
TXSEED	RXSEED
0xFFFF	0xFFFF
rw	rw

31: 16 Transmitter seed (TXSEED) - Should be manually set before transmission is stared via the SIST.CTRL.TXSTART bit.

15: 0 Receiver seed (RXSEED) - Should be manually set before reception is stared via the SIST.CTRL.RXSTART bit.

Table 93. 0xFFE00228 - SIST.LEN - Packet Length Register

31 24	23 0
RESERVED	LEN
0x00	0x000000
r	rw

31: 24 RESERVED

23: 0 Packet Length (LEN) - Generated packet length for transmitter, expected packet length for receiver

Table 94. 0xFFE0022C - SIST.CTRL - Control Register

31	30	29	28 24	23 16	15 0
TXSTART	RXSTART	AR	ADDRLEN	RATE	SEQLEN
0	0	0	00000	0x00	0X0000
rw	rw	rw	rw	rw	rw

31: Transmitter start (TXSTART) - Starts transmitter when written to 1. Transmission ongoing while bit is read as 1.

30: Receiver start (RXSTART) - Starts receiver when written to 1. Reception ongoing while bit is read as 1.

29: Auto repeat (AR) - Enable auto repeat at end of sequence (transmitter and receiver). 0 = disabled, 1 = enabled.

28: 24 SpaceWire address length (ADDRLEN) - SpaceWire address length (note that the Logical Address byte is not counted)(transmitter)

23: 16 Data rate (RATE) - Determines the transmitter delay, counted in CLK cycles, between each byte in a packet, and between each packet in a sequence. The delay will be RATE+1 CLK cycles. This is the delay at which characters are fed into the FIFO port, the actual transmission rate is determined by the traffic in the router, the link speed and the network.

15: 0 Sequence length (SEQLEN) - Used as reload value in SIST.PKTCNTR register at end of sequences when SIST.CTRL.AR bit is set.

Table 95. 0xFFE00230- SIST.ERROR0 - Error Register 0

31 24	23 16	15 8	7 0
ADDRPID	SEED	LEN	HDRCRC
0x00	0x00	0x00	0x00
rw	rw	rw	rw

31: 24 Logical address and protocol ID error counter (ADDRPID) - Indicates number of received packets with either erroneous logical address or erroneous protocol ID.

23: 16 Transaction ID error counter (SEED) - Indicates number of received packets with erroneous transaction ID field (i.e seed) (only if no header CRC error in received packet). Received seed will be used for the received packet.

15: 8 Length error counter (LEN) - Indicates number of received packets with erroneous length field (only if no header CRC error in received packet). Received length field will be used for the received packet.

7: 0 Header CRC error counter (HDRCRC) - Indicates number of received packets with erroneous header CRC.

Table 96. 0xFFE00234 - SIST.ERROR1 - Error Register 1

31 16	15 0
ONEBIT	MOREBITS
0x0000	0x0000
rw	rw

31: 16 Single bit error counter (ONEBIT) - Indicates number of single bit error detected in data byte.

15: 0 Multiple bit error counter (MOREBIT) - Indicates number of multiple bit errors detected in data byte.

Table 97. 0xFFE00238 - SIST.ERROR2 - Error Register 2

31 16	15 8	7 0
DATACRC	EOP	EEP
0x0000	0x00	0x00
rw	rw	rw

31: 16 Data CRC error counter (DATACRC) - Indicates number of received packets with erroneous data CRC in received packet.

15: 8 EOP error counter (EOP) - Indicates number of unexpected EOP received.

7: 0 EEP error counter (EEP) - Indicates number of unexpected EEP received.

Table 98. 0xFFE0023C - SIST.PKTCNTR - Packet Counter Register

31 16	15 0
TXCNTR	RXCNTR
0x0000	0x0000
rw	rw

31: 16 Transmit packet counter (TXCNTR) - Packets remaining to be transmitted. Should be manually set before transmission is started via the SIST.CTRL.TXSTART bit. Field should not be written to while packet transmission is enabled via the SIST.CTRL.TXSTART bit.

This field is also automatically initialized from the SIST.CTRL.SEQLEN field when a sequence completes and automatic repeat is enabled via the SIST.CTRL.AR bit.

15: 0 Receive packet counter (RXCNTR) - Packets remaining to be received. Should be manually set before reception is started via the SIST.CTRL.RXSTART bit. Field should not be written to while packet reception is enabled via the SIST.CTRL.RXSTART bit.

This field is also automatically initialized from the SIST.CTRL.SEQLEN field when a sequence completes and automatic repeat is enabled via the SIST.CTRL.AR bit.

Table 99. 0xFFE00240 - SIST.TIMER0 - Timer Register 0

31 16	15 0
RESERVED	TIME_MS
0x0000	0x0000
r	rw

31: 16 RESERVED

15: 0 Time counter (TIME_MS) - Most significant part of time counter. Measures transmission time. Must be manually cleared before transmission is stared via the SIST.CTRL.TXSTART bit.

Table 100. 0xFFE00244 - SIST.TIMER1 - Timer Register 1

31 0
TIME_LS
0x00000000
rw

31: 0 Time counter (TIME_LS) - Least significant part of time counter. Measures transmission time. Must be manually cleared before transmission is stared via the SIST.CTRL.TXSTART bit.

Table 101. 0xFFE00248 - SIST.STAT - Status Register

31 5	4	3	2	1	0
RESERVED	TCO	TXAF	TXF	RXAE	RXAV
0x0000000	0	0	0	1	0
r	r	r	r	r	r

31: 5 RESERVED

4: Time-Code output tick (TCO) - Reads 1 when a new Time-Code was received, sticky that is cleared on a read access. Arrival of new Time-Codes are ignored till read.

3: Almost full (TXAF) - Transmitter almost full.

2: Full (TXF) - Transmitter full.

1: Almost empty (RXAE) - Receiver almost empty.

0: Character available (RXAV) - Character available in receiver.

Table 102. 0xFFE0024C - SIST.STATE - State Register

31 8	7 4	3 0
RESERVED	TXSTATE	RXSTATE
0x0000000	0x0	0x2
r	r	r

31: 8 RESERVED

7: 4 Transmitter state (TXSTATE) - State of transmitter - see definition below.

3: 0 Receiver state (RXSTATE) - State of receiver - see definition below.

0000	0x0	cReset	Reset state
0001	0x1	sSpWAddr	SpaceWire Address state
0010	0x2	sLogAddr	Logical Address state
0011	0x3	sPID	Protocol Identifier state
0100	0x4	sTID0	Transaction Identifier state - MSB
0101	0x5	sTID1	Transaction Identifier state - LSB
0110	0x6	sLen0	Length state - MSB
0111	0x7	sLen1	Length state
1000	0x8	sLen2	Length state - LSB
1001	0x9	sHCRC	Header CRC state
1010	0xA	sData	Data Field state
1011	0xB	sDCRC	Data CRC state
1100	0xC	sEOP	End-of-Packet state
1101	0xD	sSpill	Spill state
1110	0xE	-	{unused}
1111	0xF	-	{unused}

Table 103. 0xFFE00250 - SIST.TXBYTECNTR - Transmitter Byte Count Register

31 24	23 0
RESERVED	CNTR
0x00	0x000000
r	r

31: 24 RESERVED

23: 0 Byte counter (CNTR) - Remaining number of bytes in data field of the packet being transmitted, down count.

Table 104. 0xFFE00254 - SIST.RXBYTECNTR - Receiver Byte Count Register

31 24	23 0
RESERVED	CNTR
0x00	0x000000
r	r

31: 24 RESERVED

23: 0 Byte counter (CNTR) - Remaining number of bytes in data field of the packet being received, down count.

Table 105. 0xFFE00258 - SIST.TIME - Time-Code Register

31 16	15 8	7 0
RESERVED	TIMEOUT	TIMEIN
0x0000	0x00	0x00
r	r	rw

31: 16 RESERVED

15: 8 Time-Code receive (TIMEOUT) - Time-Code received from FIFO interface (sticky from last detection). Arrival of new Time-Codes are ignored till read SIST.STAT has been read.

0: 0 Time-Code transmit (TIMEIN) - Time-Code sent to FIFO interface when written to, when SIST.PORT.EN is set.

Table 106. 0xFFE0025C - SIST.PROT- Protection Register

31 16	15 8	7	6	5	4	3	2	1	0
PROT	GIRQADDR	UC	RST	GIRQ	TCIE	IE	MASK	STOP	EN
0x0000	0x1F	0	0	0	0	0	0	0	0
r	rw	rw	rw	rw	rw	rw	rw	rw	rw

31: 16 Protection field (PROT) - Write access to register is only accepted when this field is written to with 0x55AA

15: 8 General interrupt port address (GIRQADDR) - Port number of to which a dummy access is made to generate a general interrupt in the router.

7: Unclear (UC) - When set, the state of the transmitter or receiver is not cleared when disabled through SIST.CTRL.TXSTART and SIST.CTRL.RXSTART, respectively.

6: Reset (RST) - Resets SIST when written with a logical 1. Write only.

5: Generate general interrupt (GIRQ) - Generate a General Interrupt in the router when set, at end of a non-auto-repeat transmit sequence (when SIST.CTRL.AR is cleared) or on the detection of a first error by the receiver (when SIST.PROT.STOP is set), by accessing an unimplemented SpaceWire router port (specified by SIST.PROT.GIRQADDR).

4: Time-Code Interrupt Enable (TCIE) - Enables AMBA APB interrupt when set. Interrupts are generated on the reception of Time-Codes (masked while the SIST.STAT.TCO bit is set).

3: Interrupt Enable (IE) - Enables AMBA APB interrupt when set. Interrupts are generated at end of any sequence (transmit or receive), and on the detection of a first error by the receiver (when SIST.PROT.STOP is set).

2: Mask error (MASK) - Mask errors until a correct header has been received (used when SIST.PROT.STOP is set).

1: Stop (STOP) - Stop transmitter and receiver on first error detected and then end the transmit and receive sequences.

0: Enable (EN) - Enable SIST when set.

Table 107. 0xFFE00280-0xFFE0029C - SIST.DIN - Data Input Registers 0-7

31 11	10	9	8	7 0
RESERVED	RXAE	RXAV	CTRL	DATA
0x00000	1	0	0	0x00
r	r	r	r	r

31: 11 RESERVED

10: Almost empty (RXAE) - Receiver almost empty (status before character is read out).

9: Character available (RXAV) - Character available in receiver (status before character is read out).

8: Character control part (CTRL) - Control part of character, same conditions for SIST.DIN.DATA field.

7: 0 Character data part (DATA) - Data part of character, read from FIFO when a character is available (SIST.DIN.RXAV is set), interface is enabled (SIST.PROT.EN is set), and receiver is disabled (SIST.CTRL.RXSTART is clear).

Note: This register is mapped at eight address location to allow a burst of read accesses when SIST.STAT.RXAE is clear and SIST.STAT.RXAV is set. Only one character per word is supported.

Table 108. 0xFFE002A0-0xFFE002BC - SIST.DOUT - Data Output Registers 0-7

31 9	8	7 0
RESERVED	CTRL	DATA
0x000000	0	0x00
r	w	w

31: 9 RESERVED

8: Character control part (CTRL) - Control part of character, same conditions for SIST.DOUT.DATA field.

7: 0 Character data part (DATA) - Data part of character, sent to FIFO when not full (SIST.STAT.TXF is clear), interface is enabled (SIST.PROT.EN is set), transmitter is disabled (SIST.CTRL.TXSTART is clear).

Note 1: This register is mapped at eight address location to allow a burst of write accesses when SIST.STAT.TXAF is clear and SIST.STAT.TXF is clear. Only one character per word is supported.

Note 2: Note: Since the SIST.DOUT registers are not readable, 0x00000000 will always be returned if a read is attempted.

8 UART Interface

8.1 Overview

The interface consists of an UART connected to the AMBA AHB bus as a master. A simple communication protocol, described in section 8.2.1, is supported. The protocol allows a read or write transfer to be generated to any address on the AMBA AHB bus.

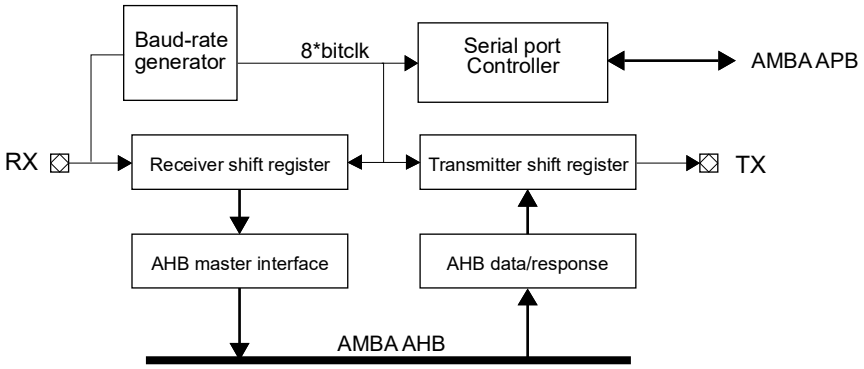


Figure 8. Block diagram

8.2 Operation

8.2.1 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.

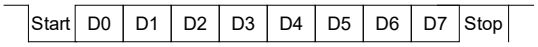


Figure 9. Data frame

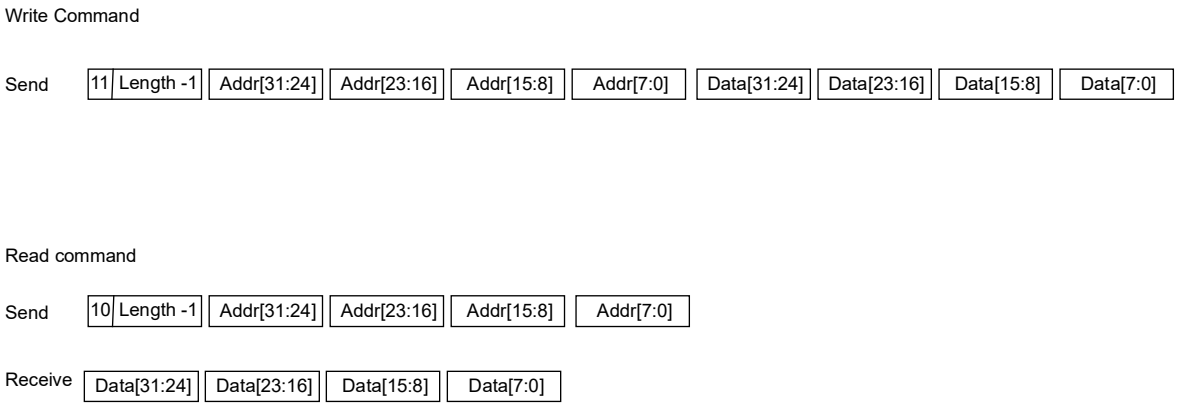


Figure 10. Commands

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.

8.2.2 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 AHB UART scaler reload register (UART.SCALE) 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 have been found, the corresponding scaler reload value is latched into the UART.SCALE register, and the BL bit is set in the AHB UART control register (UART.CTRL). If the UART.CTRL.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
```

where system_clk is the frequency in Hz of the CLK pin, and baudrate is the desired baudrate in bits per second.

8.3 Registers

The UART interface is programmed through registers mapped into AMBA address space. The register layout used is explained in section 2.4.

Table 109. AHB UART registers

AMBA address	Register	Acronym
0xFFE00004	AHB UART status register	UART.STS
0xFFE00008	AHB UART control register	UART.CTRL
0xFFE0000C	AHB UART scaler register	UART.SCALE

Table 110. 0xFFE00004 - UART.STS - AHB UART status register

31 ... 7	6	5	4	3	2	1	0
RESERVED	FE	RE	OV	BR	TH	TS	DR
0x000000	0	NA	0	0	1	1	0
r	rw	r	rw	rw	r	r	r

31: 7 RESERVED

6 Framing error (FE) - indicates that a framing error was detected.

5 RESERVED

4 Overrun (OV) - indicates that one or more character have been lost due to overrun.

3 Break (BR) - indicates that a BREAKE has been received.

2 Transmitter hold register empty (TH) - indicates that the transmitter hold register is empty.

1 Transmitter shift register empty (TS) - indicates that the transmitter shift register is empty.

0 Data ready (DR) - indicates that new data has been received by the AMBA AHB master interface.

Table 111. 0xFFE00008 - UART.CTRL - AHB UART control register

31	2	1	0
RESERVED		BL	EN
0x00000000		0	0
r		rw	rw

31: 2 RESERVED

1 Baud rate locked (BL) - is automatically set when the baud rate is locked.

0 Receiver enable (EN) - if set, enables both the transmitter and receiver.

Table 112. 0xFFE0000C - UART.SCALE - AHB UART scaler reload register

31	18	17	0
RESERVED		RELOAD	
0x0000		0x3FFFF	
r		rw	

31: 18 RESERVED

17: 0 Baudrate scaler reload value (RELOAD) - Baudrate scaler reload value = (((system_clk *10)/(baudrate*8))-5)/10, where system_clk is the frequency in Hz of the CLK pin, and baudrate is the desired baudrate in bits per second.

9 JTAG Interface

9.1 Overview

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

The JTAG interface uses a 6-bit instruction register. The IDCODE (0b001001) and BYPASS (0b111111) instructions are implemented which allows the JTAG interface to be daisy-chained. The GR718B IDCODE is 0x00718649. Boundary scan is not implemented by the GR718B.

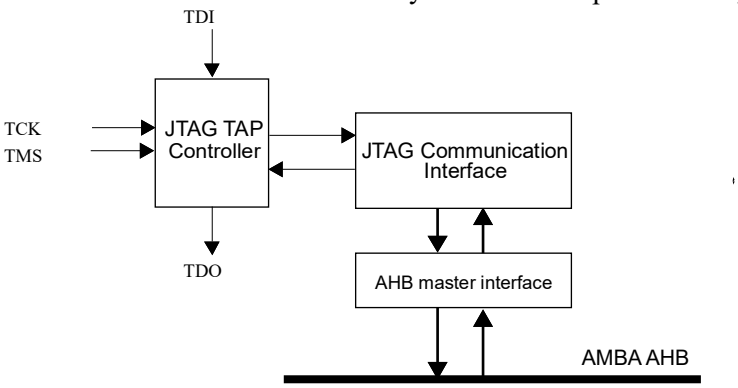


Figure 11. JTAG Debug link block diagram

9.2 Operation

9.2.1 Transmission protocol

The JTAG interface 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/address 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 the 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 KiB boundary. Sequential transfers are always word based.

Table 113. JTAG Data register, selected by instruction code 0b000011

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: 0	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 is placed in MSB bits.

Table 114. JTAG Command/Address register, selected by instruction code 0b000010

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: 0	AHB address

10 SPI Controller

10.1 Overview

The SPI controller provides a link from the router configuration port and the Serial Peripheral Interface (SPI) bus. The SPI bus parameters are highly configurable via registers. Core features also include configurable word length, bit ordering, clock gap insertion, and automatic slave select. The SPI controller only supports SPI master mode. Using the device on SPI buses with other masters could result in collisions and should be avoided.

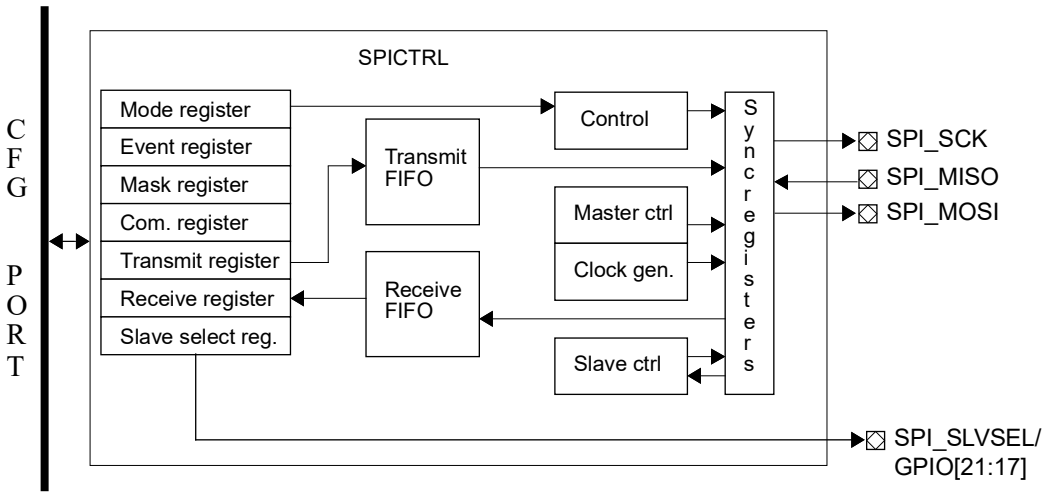


Figure 12. Block diagram

10.2 Operation

10.2.1 SPI transmission protocol

The SPI bus is a full-duplex synchronous serial bus. Transmission starts when a slave is selected through its slave select signal (SPI_SLVSEL/GPIO[21:17] pins), and the clock (SPI_SCK pin) transitions from its idle state. Data is transferred from the master through the Master-Output-Slave-Input signal (SPI_MOSI pin), and from the slave through the Master-Input-Slave-Output signal (SPI_MISO pin). In a system with only one master and one slave, the slave select input of the slave may be always active and the master does not need to have a slave select output.

During a transmission on the SPI bus, data is either changed or read at a transition of SPI_SCK. If data has been read at edge n, data is changed at edge n+1. If data is read at the first transition of SPI_SCK, the bus is said to have clock phase 0, and if data is changed at the first transition of SPI_SCK, the bus has clock phase 1. The idle state of SPI_SCK may be either high or low. If the idle state of SPI_SCK is low, the bus has clock polarity 0, and if the idle state is high, the clock polarity is 1. The combined values of clock polarity (SPI.MODE.PO) and clock phase (SPI.MODE.PH) determine the mode of the SPI bus. Figure 13 shows one byte (0x55) being transferred MSb first over the SPI bus under the four different modes. Note that the idle state of the SPI_MOSI line is ‘1’ and that SPI.MODE.PH = 0 means that the devices must have data ready before the first transition of SPI_SCK. The figure does not include the SPI_MISO signal, since the behavior of this line is the same as for the SPI_MOSI signal.

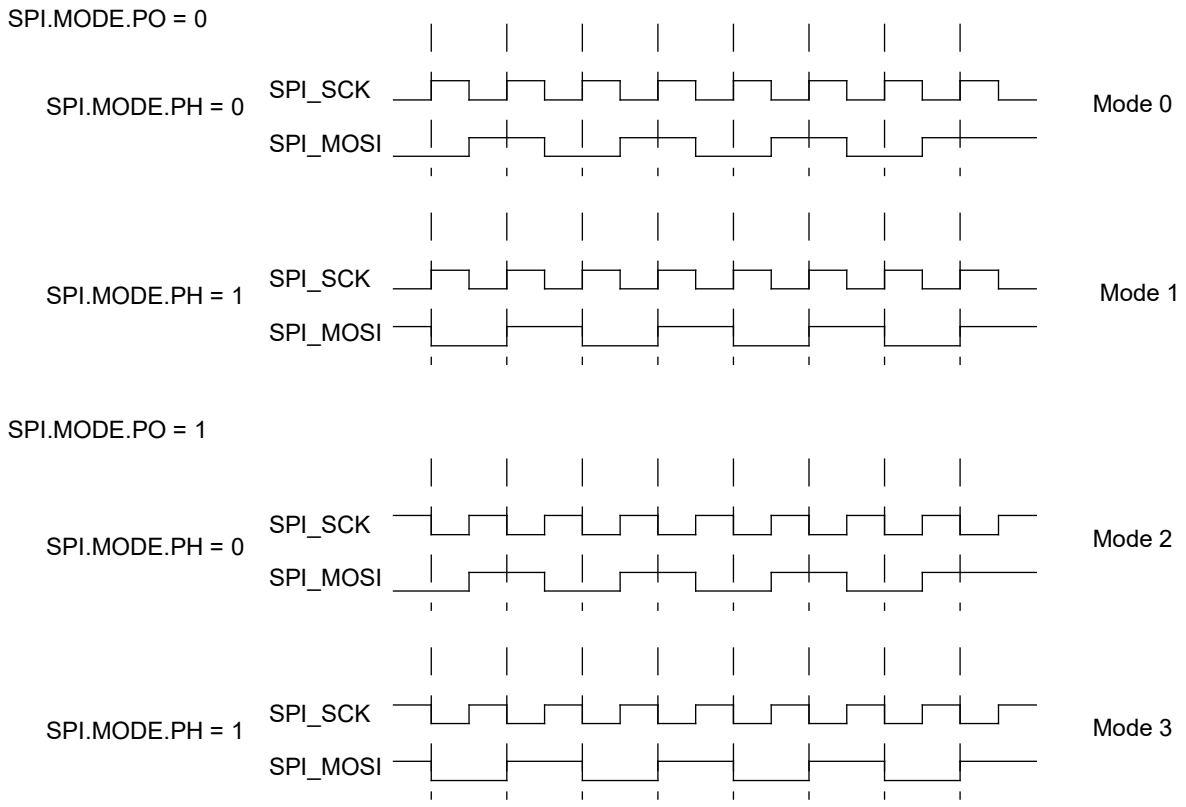


Figure 13. SPI transfer of byte 0x55 in all modes

10.2.2 Receive and transmit queues

The transmit queue consists of the Transmit register (SPI.TX) and the transmit FIFO. The receive queue consists of the Receive register (SPI.RX) and the receive FIFO. The total number of words that can exist in each queue is thus the FIFO depth plus one. When one or more free slots exists in the transmit queue, the SPI.EVENT.NF bit is asserted. Software may only write to the SPI.TX register when this bit is asserted. When a word, as defined by SPI.MODE.LEN, has been received, the data is placed in the receive queue. When the receive queue has one or more elements stored, the SPI.EVENT.NE bit will be asserted. The SPI.RX register will only contain valid data if the SPI.EVENT.NE bit is asserted, and software should not access the SPI.RX register unless this bit is set. If the receive queue is full and a new word is received, an overrun condition will occur. The received data will be discarded and the SPI.EVENT.OV bit will be set. Since only master mode is supported, an underrun condition will never occur. When the master has an empty transmit queue the bus will go into an idle state.

10.2.3 Clock generation

The SPI controller always generates the clock. The generated frequency depends on the system clock frequency and the fields SPI.MODE.DV, SPI.MODE.FC, and SPI.MODE.PM. With SPI.MODE.DV set to 0, the frequency of SPI_SCK is derived through:

$$SCKfrequency = \frac{CLKfrequency}{(4-(2 \cdot FC)) \cdot (PM+1)}$$

With SPI.MODE.DV set to 1, the frequency of SPI_SCK is derived through:

$$SCKfrequency = \frac{CLKfrequency}{16 \cdot (4 - (2 \cdot FC)) \cdot (PM + 1)}$$

Note that the fields of the SPI.MODE register should not be changed when the SPI controller is enabled (SPI.MODE.EN = 1). If the SPI.MODE.FC field is set to 1, the SPI register interface is compatible with the register interface found in MPC83xx SoCs. If the SPI.MODE.FC field is set to 1, a higher frequency SPI_SCK can be generated.

10.2.4 Master operation

The SPI controller will transmit a word when there is data available in the transmit queue. When the transmit queue is empty, SPI_SCK is driven to its idle state. Multiple masters are not supported, so the Multiple-master error bit (SPI.EVENT.MM) will never be asserted. This also means that writing the SPI.MODE.IG bit has no effect.

10.3 Registers

The SPI registers listed in this section can be accessed through the SpaceWire router's configuration port, either through the RMAP target or the AMBA AHB slave interface. Both the RMAP address and AHB address are specified in table 115. The register layout used is explained in section 2.4.

*Table 115.*SPI controller registers

RMAP address	AMBA address	Register	Acronym
0x2000	0xFFF22000	Capability register	SPI.CAP
0x2004 - 0x201C	0xFFF22004 - 0xFFF22024	RESERVED	
0x2020	0xFFF22020	Mode register	SPI.MODE
0x2024	0xFFF22024	Event register	SPI.EVENT
0x2028	0xFFF22028	Mask register	SPI.MASK
0x202C	0xFFF2202C	Command register	SPI.CMD
0x2030	0xFFF22030	Transmit register	SPI.TX
0x2034	0xFFF22034	Receive register	SPI.RX
0x2038	0xFFF22038	Slave Select register	SPI.SLVSEL
0x203C	0xFFF2203C	Automatic slave select register	SPI.ASLVSEL

Table 116. 0x2000 / 0xFFF22000 - SPI.CAP - Capability register

31 24	23 20	19 18	17	16	15 8	7 6 5	4 0
SSSZ	MAXWLEN	RES	AS	SS	FDEPTH	RESERVED	REV
0x6	0x0	0x0	1	1	0x10	0x0	0x5
r	r	r	r	r	r	r	r

31: 24 Slave Select register size (SSSZ) - Indicates supported number of slave select signals. Constant value of 0x6.

23: 20 Maximum word Length (MAXWLEN) - Indicates the maximum word length supported by the core: Constant value of 0x0, indicating support for word lengths of 4-16 bits, as well as 32-bit words.

19: 18 RESERVED

17 Automatic slave select available (AS) - Indicates support for setting slave select signals automatically. Constant value of 1.

16 Slave Select Enable (SS) - Indicates support for slave select signals. Constant value of 1.

15: 8 FIFO depth (FDEPTH) - This field indicates the depth of the internal FIFOs. Constant value of 0x10. (The number of words that can be stored in the transmit and receive queues is 0x11, since the SPIT.TX and SPI.RX registers can contain one word each as well.)

7: 5 RESERVED

4: 0 Core revision (REV) - Constant value of 0x5.

Table 117. 0x2020 / 0xFFF22020 - SPI.MODE - Mode register

31	30	29	28	27	26	25	24	23 20	19 16	15	14	13	12	11 7	6 5	4	3	2	1	0
R	LO	PO	PH	DV	RV	MS	EN	LEN	PM	R	AS	FC	R	CG	ASDEL	TA	R	IG	CI	R
0	0	0	0	0	0	0	0	0x0	0x0	0	0	0	0	0x0	0x0	0	0	0	0	0
r	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	r	rw	rw	rw	r	rw*	rw	r

31 RESERVED

30 Loop mode (LO) - When this bit is set, the SPI controller's transmitter and receiver are interconnected, and the controller will operate in loopback mode.

29 Clock polarity (PO) - Determines the polarity (idle state) of the SPI_SCK clock.

28 Clock phase (PH) - When this bit is '0', data will be read on the first transition of SPI_SCK. When this bit is '1' data will be read on the second transition of SPI_SCK.

27 Divide by 16 (DV) - Divide system clock by 16, see description of SPI.MODE.PM field below, and see section 10.2.3 on clock generation.

26 Reverse data (RV) - When this bit is '0', data is transmitted LSb first, when this bit is '1' data is transmitted MSb first. This bit affects the layout of the SPI.TX and SPI.RX registers.

25 Master/Slave (MS) - Only master mode is supported so this bit must be set to 1 for correct operation.

24 Enable core (EN) - When this bit is set to '1' the core is enabled. No fields in the SPI.MODE register should be changed while this bit is set to '1'.

23: 20 Word length (LEN) - The value of this field determines the length in bits of a transfer on the SPI bus. Values are interpreted as:

0b0000 - 32-bit word length

0b0001-0b0010 - Illegal values

0b0011-0b1111 - Word length is SPI.MODE.LEN+1, allows words of length 4-16 bits.

The value of this field must never specify a word length that is greater than the maximum allowed word length specified by the SPI.CAP.MAXWLEN field.

Table 117. 0x2020 / 0xFFF22020 - SPI.MODE - Mode register

19: 16 Prescale modulus (PM) - This value is used to divide the system clock and generate the SPI clock (SPI_SCK pin). The value in this field depends on the value of the SPI.MODE.FC bit.

If SPI.MODE.FC is '0':
The system clock (CLK pin) is divided by 4*(SPI.MODE.PM+1) if the SPI.MODE.DV field is '0', and 16*4*(SPI.MODE.PM+1) if the SPI.MODE.DV field is '1'. The highest SPI_SCK frequency is attained when SPI.MODE.PM is set to 0b0000 and SPI.MODE.DV to '0'. This configuration will give a SPI_SCK frequency that is CLK/4. When SPI.MODE.FC is '0', the register interface is compatible with that found in MPC83xx SoCs.

If SPI.MODE.FC is '1':
The system clock (CLK pin) is divided by 2*(SPI.MODE.PM+1) if the SPI.MODE.DV field is '0', and 16*2*(SPI.MODE.PM+1) if the SPI.MODE.DV field is set to '1'. The highest SPI_SCK frequency is attained when SPI.MODE.PM is set to 0b0000 and SPI.MODE.DV to '0'. This configuration will give a SPI_SCK frequency that is CLK/2.

15 RESERVED

14 Automatic slave select (AS) - If this bit is set to '1' the controller will swap the contents in the SPI.SLVSEL register with the contents of the SPI.ASLVSEL register automatically when a transfer is started. When the transmit queue is empty, the SPI.SLVSEL register will be swapped back. Note that if the controller is disabled (by writing the SPI.MODE.EN bit to '0') when a transfer is in progress, the registers may still be swapped when the interface goes idle. Also see the SPI.MODE.ASDEL field, which can be used to insert a delay between the slave select register swap and the start of a transfer.

13 PM factor (FC) - If this bit is 1, the controller's register interface is no longer compatible with the MPC83xx register interface. The value of this bit affects how the SPI.MODE.PM field is utilized to scale the SPI clock. See the description of the SPI.MODE.PM field.

12 RESERVED

11: 7 Clock gap (CG) - The value of this field determines how many SPI_SCK clock cycles that is inserted between each consecutive word. This only applies when the transmit queue is kept non-empty. After the last word of the transmit queue has been sent, the controller goes into idle state, and will continue to transmit data as soon as a new word is written to the SPI.TX register, regardless of the value of this field. A value of 0b00000 in this field enables back-to-back transfers.

6: 5 Automatic Slave Select Delay (ASDEL) - If the controller is configured to use automatic slave select (SPI.MODE.AS field set to '1'), it will insert a delay corresponding to SPI.MODE.ASDEL*(SPI_SCK cycle time)/2 between the swap of the slave select registers and the first toggle of SPI_SCK. As an example, if this field is set to "10" the core will insert a delay corresponding to one SPI_SCK cycle between assigning the SPI.ASLVSEL register to the SPI.SLVSEL register and toggling SPI_SCK for the first time in the transfer.

4 Toggle Automatic slave select during Clock Gap (TA) - If this bit is set, and the SPI.MODE.AS field is set, the core will perform the swap of the slave select registers at the start and end of each clock gap. The clock gap is defined by the SPI.MODE.CG field and must be set to a value >= 2 if this field is set.

3 RESERVED

2 Ignore SPISEL input (IG) - This bit is writable, but setting it has no effect (since only master mode is supported).

1 Require Clock Idle for Transfer End (CI) - If this bit is '0', the controller will regard the transfer of a word as completed when the last bit has been sampled. If this bit is set to '1', the controller will wait until it has set SPI_SCK to its idle level before regarding a transfer as completed. This setting only affects the behavior of the SPI.EVENT.TIP bit, and automatic slave select toggling at the end of a transfer, when the SPI.MODE.PH bit is '0'.

0 RESERVED

Table 118. 0x2024 / 0xFFF22024 - SPI.EVENT - Event register

31	30	15	14	13	12	11	10	9	8	7	0
TIP	RESERVED		LT	R	OV	RES		NE	NF	RESERVED	
0	0x0000		0	0	0	0x0		0	0	0x00	
r	r		wc	r	wc	r		r	r	r	

31 Transfer in progress (TIP) - This bit is '1' when a transfer is in progress, and set to '0' when the transfer is considered to be finished. Behavior affected by setting of SPI.MODE.CI field.

Table 118. 0x2024 / 0xFFF22024 - SPI.EVENT - Event register

30: 15 RESERVED

14 Last character (LT) - This bit is set when a transfer completes, if the transmit queue is empty and the SPI.CMD.LT bit is set.

13 RESERVED

12 Overrun (OV) - This bit gets set when the receive queue is full and new data is received. The controller continues communicating over the SPI bus but discards the new data.

11: 10 RESERVED

9 Not empty (NE) - This bit is set when the receive queue contains one or more elements. It is cleared automatically by the controller when the receive queue is empty.

8 Not full (NF) - This bit is set when the transmit queue has room for one or more words. It is cleared automatically by the core when the transmit queue is full.

7:0 RESERVED

Table 119. 0x2028 / 0xFFF22028 - SPI.MASK - Mask register

31	30 15	14	13	12	11	10	9	8	7 0
TIP	RESERVED	LT	R	OV	UN	MM	NE	NF	RESERVED
0	0x0000	0	0	0	0	0	0	0	0x00
rw*	r	rw*	r	rw*	rw*	rw*	rw*	rw*	r

31 Transfer in progress enable (TIP) - Writable, but has no effect (used in designs where SPI controller is allowed to generate interrupt).

30: 15 RESERVED

14 Last character enable (LT) - Writable, but has no effect (used in designs where SPI controller is allowed to generate interrupt).

13 RESERVED

12 Overrun enable (OV) - Writable, but has no effect (used in designs where SPI controller is allowed to generate interrupt).

11 Underrun enable (UN) - Writable, but has no effect (used in designs where SPI controller is allowed to generate interrupt).

10 Multiple-master error enable (MM) - Writable, but has no effect (used in designs where SPI controller is allowed to generate interrupt).

9 Not empty enable (NE) - Writable, but has no effect (used in designs where SPI controller is allowed to generate interrupt).

8 Not full enable (NF) - Writable, but has no effect (used in designs where SPI controller is allowed to generate interrupt).

7:0 RESERVED

Table 120. 0x202C / 0xFFF2202C - SPI.CMD - Command register

31 23	22	21 0
RESERVED	LT	RESERVED
0x00	LT	0x000000
r	w	r

31: 23 RESERVED

22 Last (LT) - When this bit is written to ‘1’, the controller sets an internal register to ‘1’. When the internal register is ‘1’, and the last character in the transmit queue is sent, the SPI.EVENT.LT bit is set. The internal register is automatically cleared when the SPI.EVENT.LT bit is set.

21: 0 RESERVED

Note: Since this register is not readable, 0x00000000 will always be returned if a read is attempted.

Table 121. 0x2030 / 0xFFF22030 - SPI.TX - Transmit register

31	0
TDATA	
N/R	
rw	

31: 0 Transmit data (TDATA) - Writing a word into this register places the word in the transmit queue. This register will only react to writes if the SPI.EVENT.NF is set. The layout of this register depends on the value of the SPI.MODE.RV field:

SPI.MODE.RV = '0': The word to transmit should be written with its least significant bit at bit 0.

SPI.MODE.RV = '1': The word to transmit should be written with its most significant bit at bit 31.

Table 122. 0x2034 / 0xFFF22034 - SPI.RX - Receive register

31	0
RDATA	
N/R	
r	

31: 0 Receive data (RDATA) - This register contains valid receive data when the SPI.EVENT.NE bit is set. The placement of the received word depends on the fields SPI.MODE.LEN and SPI.MODE.RV:

For SPI.MODE.LEN = 0b0000 - The data is placed with its MSb in bit 31 and its LSb in bit 0.

For other lengths and SPI.MODE.RV = '0' - The data is placed with its MSb in bit 15.

For other lengths and SPI.MODE.RV = '1' - The data is placed with its LSb in bit 16.

To illustrate this, a transfer of a word with eight bits (SPI.MODE.LEN = 7) that are all set to one will have the following placement:

SPI.MODE.RV = '0' - 0x0000FF00

SPI.MODE.RV = '1' - 0x00FF0000

Table 123. 0x2038 / 0xFFF22038 - SPI.SLVSEL - Slave select register

31	6	5	0
RESERVED		SLVSEL	
0x000000		0x3F	
r		rw	

31: 6 RESERVED

5: 0 Slave select (SLVSEL) - Sets the slave select signals.

Table 124. 0x203C / 0xFFF2203C - SPI.ASLVSEL - Automatic slave select register

31	6	5	0
RESERVED		ASLVSEL	
0x000000		N/R	
r		rw	

31: 6 RESERVED

5: 0 Automatic Slave select (ASLVSEL) - The slave select signals are assigned from this register when the controller is about to perform a transfer, and the SPI.MODE.AS field is set to '1'. After a transfer has been completed, the slave select signals are assigned the original value from the SPI.SLVSEL register.

11 General Purpose I/O Interface

11.1 Operation

The input from the GPIO pins are synchronized by two flip-flops in series to remove potential metastability, and the values can be read from the GPIO.DATA register. The values in the GPIO.OUT register are mapped to the internal signals *gpio[23:0]*. Whether or not the *gpio[23:0]* signals are mapped to the GPIO pins is controlled by the GPIO multiplexing, as described in section 5. When the *gpio[23:0]* signals are mapped to the GPIO pins, the GPIO.DIR register determines the direction of the GPIO pins.

11.2 Registers

The general purpose I/O registers listed in this section can be accessed through the router's configuration port, either through the RMAP target, or the AMBA AHB slave interface. Both the RMAP address and AMBA address are specified in table 125. The register layout used is explained in section 2.4.

Table 125. General Purpose I/O Port registers

RMAP address	AMBA address	Register	Acronym
0x2100	0xFFF22100	I/O port data register	GPIO.DATA
0x2104	0xFFF22104	I/O port output register	GPIO.OUT
0x2108	0xFFF22108	I/O port direction register	GPIO.DIR
0x210C - 0x2118	0xFFF2210C - 0xFFF22118	RESERVED	
0x211C	0xFFF2211C	Capability register	GPIO.CAP

Table 126. 0x2100 / 0xFFF22100 - GPIO.DATA - I/O port data register

31 24	23 0
RESERVED	INVAL
0x00	*
r	r

31: 24 RESERVED

23: 0 I/O pin input value (INVAL) - Each bit shows the input value of the corresponding GPIO pin. Reset value depends on values on GPIO pins.

Table 127. 0x2104 / 0xFFF22104 - GPIO.OUT - I/O port output register

31 24	23 0
RESERVED	OUTVAL
0x00	0x000000
r	rw

31: 24 RESERVED

23: 0 I/O pin output value (OUTVAL) - Each bit determines the value driven on the corresponding GPIO pin when the matching bit in the GPIO:DIR.DIR field is set to 1.

Table 128. 0x2108 / 0xFFF22108 - GPIO.DIR - I/O port direction register

31 24	23 0
RESERVED	DIR
0x00	0x000000
r	rw

31: 24 RESERVED

23: 0 I/O pin direction (DIR) - Each bit determines the direction of the corresponding GPIO pin. 0 = output disabled, 1 = output enabled.

Table 129. 0x211C / 0xFFF2211C - GPIO.CAP - Capability register

31 5	4 0
RESERVED	NPINS
0x0000000	0x17
r	r

31: 5 RESERVED

4: 0 Number of GPIO pins (NPINS) - Shows the number of GPIO pins minus one. Constant value of 0x17.

12 System Level Test Configuration

Each SpaceWire port can be configured to operate in loop-back mode, either internal loop-back, external loop-back, or both. This feature is intended for system level testing.

Internal loop-back means that the ports internal data and strobe signals are not mapped to the corresponding external SpaceWire I/O pins. They are instead routed back to the port internally (transmit data to receive data, transmit strobe to receive strobe). External loop-back means that the external SpaceWire I/O pins are not routed to the corresponding port. Instead they are routed back out on the external pins (SPW_RXDp/n to SPW_TXDp/n and SPW_RXSp/n to SPW_TXSp/n). Loop back is controlled from the registers described in 12.1.

12.1 Registers

The registers are mapped into AMBA address space. The register layout used is explained in section 2.4.

*Table 130.*System level test configuration registers

AMBA address	Register	Acronym
0xFFE00100	System level test configuration 1	SYSTEST.CFG1
0xFFE00104	System level test configuration 2	SYSTEST.CFG2

Table 131. 0xFFE00100 - SYSTEST.CFG1 - System level test configuration 1

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
LB16	LB15	LB14	LB13	LB12	LB11	LB10	LB9	LB8	LB7	LB6	LB5	LB4	LB3	LB2	LB1
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

31: 0 Loop-back control (LBxx) - Each two-bit field controls the loop-back connection for one SpaceWire port. LB1 controls SpaceWire port 1, LB2 controls SpaceWire port 2 etc. The LSb of each LBxx field (i.e. bit 0, bit 2 etc) controls the internal loop-back, while the MSb of each LBxx field (i.e. bit 1, bit 3 etc) controls the external loop-back.

Table 132. 0xFFE00104 - SYSTEST.CFG2 - System level test configuration 2

31 5	4	3 2	1 0
RESERVED	EN	LB18	LB17
0x0000000	0	0x0	0x0
r	rw	rw	rw

4 SIST enable (EN) - When this bit is set to 1 the SpaceWire In-System Test block (described in section 7) is enabled and clocked. When this bit is cleared the SpaceWire In-System Test block is disabled and clock-gated.

3: 0 See description of LBxx fields in table 131.

13 AMBA AHB controller with plug & play support

13.1 Overview

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

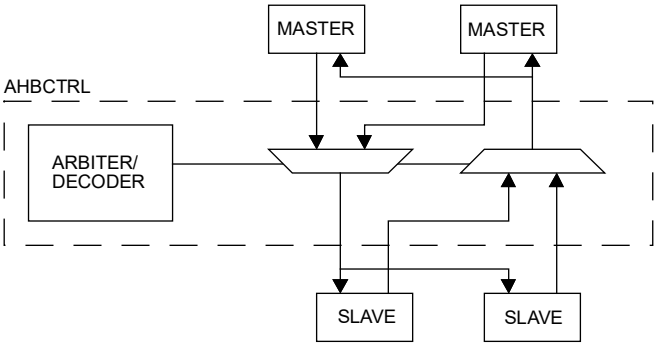


Figure 14. AHB controller block diagram

13.2 Operation

13.2.1 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).

13.2.2 Decoding

Decoding of AHB slaves is done using the plug & play, detailed in section 2.3.1. A slave can occupy any binary aligned address space with a size of 1 - 4096 MiB. Access to unused addresses will cause an AHB error response.

14 AMBA AHB/APB bridge with plug & play support

14.1 Overview

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

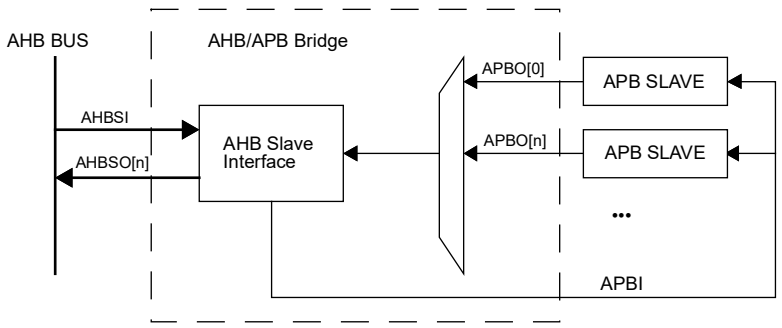


Figure 15. AHB/APB bridge block diagram

14.2 Operation

14.2.1 Decoding

Decoding of APB slaves is done using the plug & play, detailed in section 2.3.1.

15 Electrical description

15.1 Absolute maximum ratings

Table 133. Absolute maximum ratings [1) 2)]

Symbol	Parameter	Rating		Units
		Min.	Max.	
V_{DDIO}	DC Supply Voltage for I/O	-0.3	4.0	V
V_{DD}	DC Supply Voltage for Core	-0.3	2.2	V
V_{DDA}	Analog Supply Voltage for PLL	-0.3	2.2	V
V_{IN_LVTTL}	Input Voltage for LVTTL I/O	-0.3	V_{DDIO} + 0.3 [3)]	V
V_{IN_LVDS} [4)]	Input Voltage for LVDS I/O	-0.3	V_{DDIO} + 0.3 [3)]	V
$V_{IN\_VREFEXT}$ [4)]	Input Voltage for VREFEXT	-0.3	V_{DDIO} + 0.3 [3)]	V
V_{OUT} [4)]	Output Voltage	-0.3	V_{DDIO} + 0.3 [3)]	V
I_{IN} [4)]	Input pin current for VREFEXT, LVTTL and LVDS I/O	-10	10	mA
T_{store}	Storage Temperature	-65	+150	°C
T_{solder}	Lead Temperature (Soldering 10 sec.)		+250	°C
T_j	Junction Temperature		+150	°C
Θ_{JC} (ceramic)	Thermal Resistance, Junction to Case		4	°C/W
P_D	Power Dissipation		6.0	W
V_{ESD}	Human Body Model ESD level		1000	V

Note 1: Extended operation at the maximum levels may degrade the performance and affect the reliability of the device. Exceeding any maximum rating level may permanently damage the device.

Note 2: All GND pins must be connected to the same GND plane on PCB. Any externally applied voltage difference between GND pins can create harmful circulating ground currents inside the package

Note 3: Voltage shall not exceed 4.0V.

Note 4: All on-chip LVDS transceivers and all LVTTL are non-cold-spare ports. All inputs and outputs have ESD protection via on-chip diodes to IO ground and IO supply. Forward bias voltage for on-chip ESD protection diodes should not exceed 0.3 V and DC current should not exceed 10mA. For more information see section 15.6 for equivalent LVDS ESD and IO buffer schematics.

15.2 Recommended operating conditions

Table 134. Recommended operating conditions

Symbol	Parameter	Rating			Units
		Min.	Typ.	Max.	
V_{DDIO}	DC Supply Voltage for I/O	3.0 [1) 3) 4)]	3.3	3.6 [1)]	V
V_{DD}	DC Supply Voltage for Core	1.65 [3) 4)]	1.8	1.95	V
V_{DDA} [5)]	Analog Supply Voltage for PLL	1.65 [3) 4)]	1.8	1.95	V
V_{IN_LVTTL}	Input Voltage for LVTTL I/O	0		V_{DDIO}	V
V_{IN_LVDS}	Input Voltage for LVDS I/O	0		V_{DDIO}	V
$V_{IN\_VREFEXT}$	Input voltage for VREFEXT	1.2	1.25	1.3	V
$\|V_{ID}\|$	Magnitude of LVDS differential input voltage	0.1 [2)]		0.6	V
T_{case}	Operating Case Temperature	-55		+125	°C
SL_{IN_LVTTL}	Slew rate of all LVTTL inputs	1			V/ns
SL_{IN_LVDS}	Slew rate of all LVDS inputs	0.1			V/ns

Note 1: Applies only with external LVDS voltage reference. Use of internal LVDS voltage reference will reduce the recommended range for V_{DDIO} to 3.3V±5%

Note 2: Failure to comply with the differential input thresholds will make the receiver logic condition unknown. For noisy environment or protection from various failures such as open inputs, floating inputs or shorted inputs an external fail-safe function should be considered.

Note 3: All on-chip LVDS transceivers and CMOS ports are non-cold-spare ports (below minimum recommended supply voltage). All ports have ESD protection via on-chip diodes to IO ground and IO supply.

Note 4: All outputs are non-defined, i.e. spurious spikes might occur on the outputs, when the supply voltages are below minimum recommended limits.

Note 5: The analog supply voltage for PLL (VDDA) may draw increased supply current that may impact long-term reliability if VDDA is supplied while VDD is not supplied. Short-term conditions, such as power-up and power-down, do not impact long-term reliability

15.3 Power supplies

Table 135. DC characteristics (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Symbol	Parameter	Condition	Rating			Units
			Min.	Typ.	Max.	
I_{DD}	Core Supply Current	F_{CLK} = 50Mhz and internal SpaceWire clock at 200 MHz [1)]		0.75	1.0	A
I_{DDS}	Core Standby Current	No clocks		2	50	mA
I_{DDA}	PLL Supply Current	F_{CLK} = 50Mhz and internal SpaceWire clock at 200 MHz		5	15	mA
I_{DDIO}	I/O Supply Current	1) 4)		450	600 [5)]	mA
$I_{DDIOS\_LVDS\_E}$	I/O Standby Current	2) 4)		350	450	mA
$I_{DDIOS\_LVDS\_D}$	I/O Standby Current	3) 4)		1	10	mA

Note 1: All SpaceWire links active at full data rate

Note 2: IDDIO with all LVDS receivers and transmitters enabled and no clock signals.

Note 3: IDDIO with all LVDS receivers and transmitters disabled and no clock signals.

Note 4: All LVDS output terminals are terminated with 100Ω and no external load on the LVTTL outputs and $|V_{ID}| > 100$ mV

Note 5: Valid for any recommended operating condition. The I/O supply current may exceed 600mA by a maximum of 200mA if the input levels on the LVDS inputs are not within their recommended operation conditions (e.g with |VID| < 100mV).

Note: There is no direct ESD protection between V_{DD} and V_{DDIO}, so they can be powered up independently. However, it is recommended to power up V_{DDIO} before V_{DD} in order to minimize toggling, and hence reduce the rush-in currents and power dissipation.

15.4 Input voltages, leakage currents and capacitances

Table 136. DC characteristics for LVTTL inputs (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Symbol	Parameter	Condition	Rating			Units
			Min.	Typ.	Max.	
V_{IH1} 3)	High-level input voltage		2.0			V
V_{IH2} 4)	High-level input voltage		2.1			V
V_{IL}	Low-level input voltage				0.8	V
I_{ILEAK_1} 1)	Input Leakage Current	$V_{IN} = V_{DDIO}$			10	uA
		V_{IN} = 0V	-10			uA
I_{ILEAK_2} 2)	Input Leakage Current	$V_{IN} = V_{DDIO}$			160	uA
		V_{IN} = 0 V	-10			uA
C_{I_LVTTL} 5)	LVTTL Input capacitance			5 5)		pF

Note 1: TESTEN[0:1] and SELVREF input signals are not included.VREFEXT analog signal is included.

Note 2: TESTEN[0:1] and SELVREF input signals only.

Note 3: RESETn input signal is not included.

Note 4: RESETn input signal only. This is a Schmitt trigger input.

Note 5: IBIS model including data per pin is available upon request

Table 137. DC characteristics for LVDS inputs (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Symbol	Parameter	Condition	Rating			Units
			Min.	Typ.	Max.	
V_{CM} 4)	Common mode input voltage	$\|V_{ID}\|$ = 100mV	0.05		2.35	V
		$\|V_{ID}\|$ = 600mV	0.3		2.1	V
$\|V_{ID}\|$	Magnitude of differential input voltage	V_{CM} = +0.3V to +2.1V	0.1 3)		0.6	V
V_{TH}	Differential input high threshold		100			mV
V_{TL}	Differential input low threshold				-100	mV
I_I 4)	Input current		-20		20	uA
I_{IB}	Input balance current		-6		6	uA
C_{I_LVDS}	LVDS Input capacitance			5 5)		pF

Note 1: LVDS receivers fully compliant to ANSI TIA/EIA-644 “Low Voltage Differential Signaling”

Note 2: LVDS receivers are designed with no differential input voltage hysteresis

Note 3: Failure to comply with the differential input thresholds will make the receiver logic condition unknown. For noisy environment or protection from various failures such as open inputs, floating inputs or shorted inputs a external fail-safe function should be considered.

Note 4: All on-chip LVDS receivers are non-cold-spare ports. The LVDS input pin voltage must be between GND V_{DDIO}, otherwise the input current limits for I_I are not valid due to the on-chip ESD protection diodes will conduct current.

Note 5: IBIS model including data per pin is available upon request

15.5 Output voltages, leakage currents and capacitances

Table 138. DC characteristics for LVTTL outputs (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Symbol	Parameter	Condition	Rating			Units
			Min.	Typ.	Max.	
V_{OH}	Output High Voltage	I_{OH} = -2 mA[1)]	2.4			V
		I_{OH} = -6 mA[2)]	2.4			V
V_{OL}	Output Low Voltage	I_{OL} = 2 mA[1)]			0.4	V
		I_{OL} = 6 mA[2)]			0.4	V
I_{OLEAK}	Output Leakage Current	Outputs at tri-state. V_{OUT} = V_{DDIO} and V_{OUT}=0V	-10		10	uA
C_{O_LVTTL}	LVTTL Output capacitance			5 [3)]		pF

Note 1: All outputs defined with 2mA drive capability in table 151 and GPIO[0:23]

Note 2: All outputs defined with 6mA drive capability in table 151

Note 3: IBIS model including data per pin is available upon request

Table 139. DC characteristics for LVDS outputs (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Symbol	Parameter	Condition	Rating			Units
			Min.	Typ.	Max.	
V_{OS}	Offset voltage	1)	1.125	1.250	1.375	V
ΔV_{OS}	Change in magnitude of V_{OS} for complementary output states	1)	-50		50	mV
$\|V_{OD}\|$	Absolute differential Output voltage	1)	250	350	450	mV
$\Delta\|V_{OD}\|$	Change in magnitude of $\|V_{OD}\|$ for complementary output states	1)	-50		50	mV
I_{OZ} 5)	Output leakage current when disabled	4)	-2		2	uA
I_{OS}	Short-circuit Output current				24	mA
I_{ODS}	Differential short-circuit Output current				12	mA
C_{O_LVDS}	LVDS Output capacitance			5 6)		pF

Note 1: LVDS outputs are terminated with 100 Ohm.

Note 2: LVDS drivers fully compliant to ANSI TIA/EIA-644 “Low Voltage Differential Signaling”

Note 3: V_{DDIO} = 3.3 V +/- 0.3 V only applies with external LVDS voltage reference. See table 134 under recommended operating conditions for V_{DDIO} restrictions when using internal LVDS voltage reference.

Note 4: Each LVDS pair is internally connected with a resistor (>1kΩ). The given output leakage current values only apply with the opposite LVDS output terminal floating.

Note 5: All on-chip LVDS transmitters are non-cold-spare ports. The LVDS output pin voltage must be between GND V_{DDIO}, otherwise the output current limits for I_{OZ} are not valid due to the on-chip ESD protection diodes will conduct current.

Note 6: IBIS model including data per pin is available upon request

15.6 Simplified IO buffer schematics

Simplified input and output buffer schematics presented in this chapter is applicable within absolute maximum rating conditions, see chapter 15.1

15.6.1 Simplified LVDS input buffer schematic

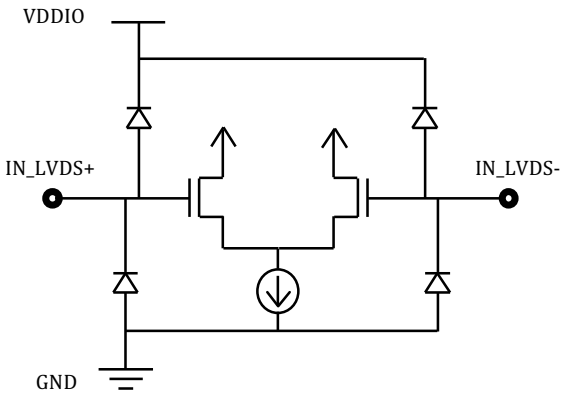


Figure 16. Simplified LVDS input buffer schematic

15.6.2 Simplified LVDS output buffer schematic

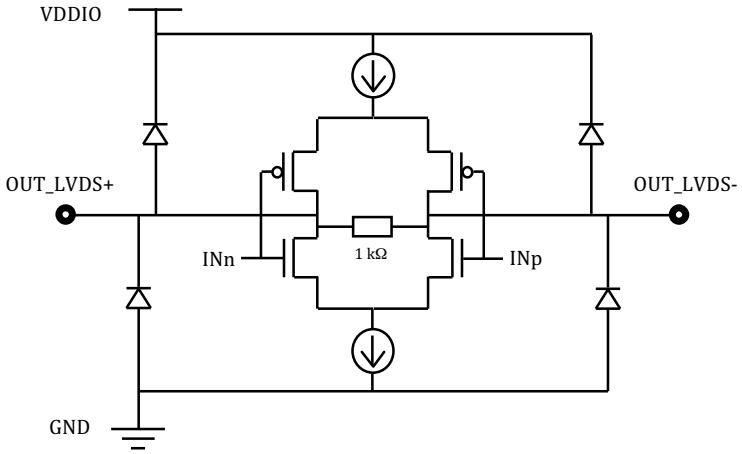


Figure 17. Simplified LVDS output buffer schematic

15.7 AC characteristics

All measured AC parameters have been tested with 10 pF to 20 pF capacitive load on the outputs. Timing measurements have been performed using a voltage level equivalent to $V_{DDIO}/2$.

15.7.1 System clock timing

The timing waveforms are shown in figure 18, and the timing parameters are defined in table 140.

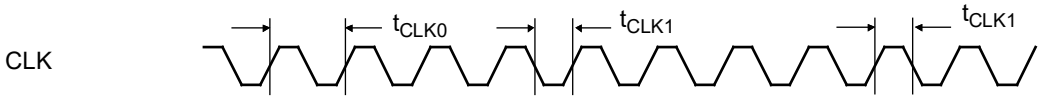


Figure 18. System clock timing waveforms

Table 140. System clock timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	Parameter	Reference edge	Min	Max	Unit
t_{CLK0}	Clock period [2]	-	20	[1]	ns
	Clock period [3] [4]		20	50	ns
t_{CLK1}	Clock high/low pulse length	-	0.40 x t_{CLK0}	0.60 x t_{CLK0}	ns
t_{CLK2}	Clock cycle jitter [3]	-	-	2	ns

Note 1: Max value can not be larger than 8 x clock period of internal SpaceWire clock.

Note 2: Only applicable when system clock is not used as input to the PLL

Note 3: Only applicable when PLL use CLK to generate internal SpaceWire clock (see section 4)

Note 4: When SpaceWire clock is used as source to the PLL the internal SpaceWire clock shall not exceed 200MHz

15.7.2 External SpaceWire clock timing

The timing waveforms are shown in figure 19, and the timing parameters are defined in table 141.

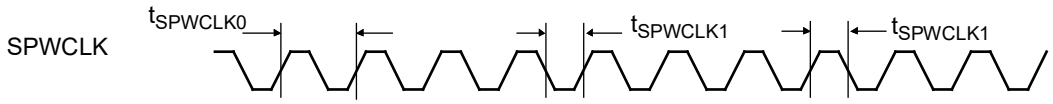


Figure 19. External SpaceWire clock timing waveforms

Table 141. External SpaceWire clock timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	Parameter	Reference edge	Min	Max	Unit
$t_{SPWCLK0}$	Clock period [1]	-	10	-	ns
	Clock period [2] [3]	-	20	50	ns
$t_{SPWCLK1}$	Clock high/low pulse length [1]	-	0.45 x $t_{SPWCLK0}$	0.55 x $t_{SPWCLK0}$	ns
	Clock high/low pulse length [2]	-	0.40 x $t_{SPWCLK0}$	0.60 x $t_{SPWCLK0}$	ns
$t_{SPWCLK2}$	Clock cycle jitter [1]	-	-100	100	ps
	Clock cycle jitter [2]	-	-	2	ns

Note 1: Only applicable when PLL is bypassed (see section 4).

Note 2: Only applicable when PLL use SPWCLK to generate internal SpaceWire clock (see section 4)

Note 3: When SpaceWire clock is used as source to the PLL the internal SpaceWire clock shall not exceed 200MHz

15.7.3 LOCK timing

The timing waveforms are shown in figure 20, and the timing parameters are defined in table 142.

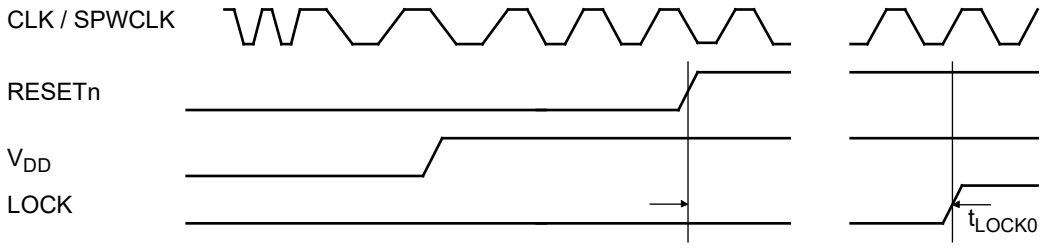


Figure 20. LOCK timing waveforms

*Table 142.*LOCK pin timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	**Parameter**	**Condition**	**Typ**	**Max**	**Unit**
t_{LOCK0}	PLL lock time	Clock input stable [1], RESETn de-asserted, and V_{DD} and V_{DDA} at least minimum operating voltages.	5	30	us

Note 1: Either CLK or SPWCLK can be used by PLL to generate the internal SpaceWire clock. See section 4.1

15.7.4 Reset timing

The timing waveforms are shown in figure 21, and the timing parameters are defined in table 143.

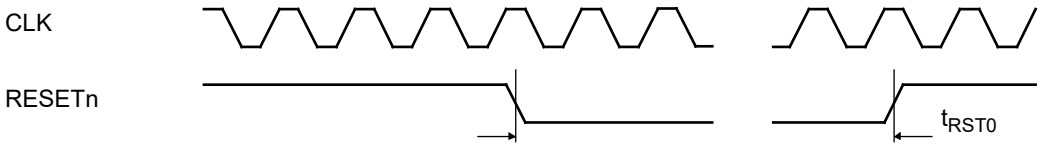


Figure 21. Reset timing waveforms

*Table 143.*Reset timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	Parameter	Reference edge	Min	Max	Unit
t_{RST0}	Asserted period	-	10 x t_{CLK0}	-	ns
Note 1:	The RESETN input is re-synchronized internally, and does not have to meet any setup or hold requirements.				
Note 2:	V_{DD} must reach at least minimum operating voltage before start for t_{RST0} before RESETn is de-asserted.				
Note 3:	The internal reset for the system clock domain is released 5 x t_{CLK0} after RESETn is de-asserted. The internal reset for the SpaceWire clock domain is released 5 x internal SpaceWire clock cycles after RESETn is de-asserted and PLL has acquired lock.				

15.7.5 SPI interface timing

The timing waveforms are shown in figure 22, and the timing parameters are defined in table 144.

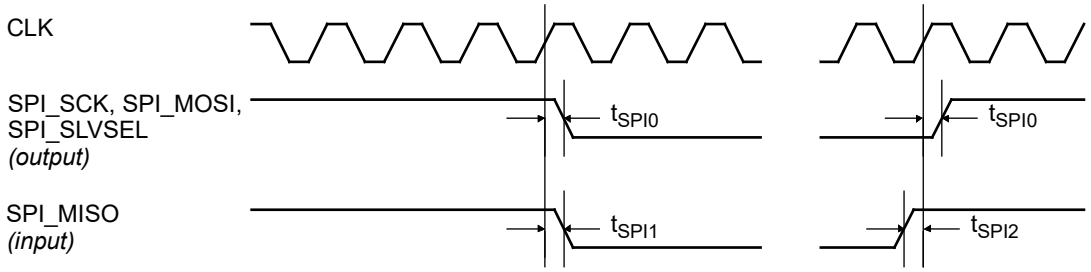


Figure 22. SPI interface timing waveforms

*Table 144.*SPI interface timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	**Parameter**	**Reference edge**	**Min**	**Max**	**Unit**
t_{SPI0}	Clock to output delay	Rising CLK edge	0	20	ns
t_{SPI1}	Input to clock hold	Rising CLK edge	-	-	ns
t_{SPI2}	Input to clock setup	Rising CLK edge	-	-	ns
Note 1:	The SPI_MISO input is re-synchronized internally, and does not have to meet any setup or hold requirements.				

15.7.6 SpaceWire LVDS interface timing

The timing waveforms are shown in figure 23, and the timing parameters are defined in table 145.

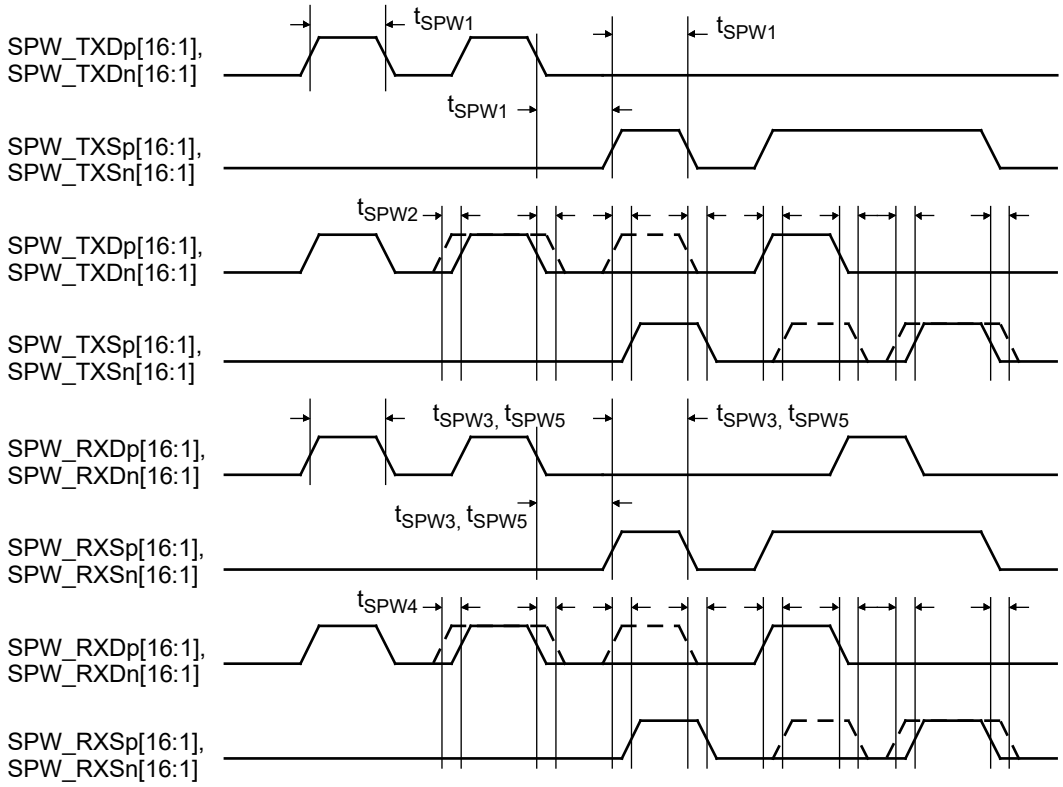


Figure 23. SpaceWire LVDS interface timing waveforms

Table 145. SpaceWire LVDS interface timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	Parameter	Reference edge	Min	Max	Unit
t_{SPW1_LVDS}	Output data bit period	-	5	500	ns
t_{SPW2_LVDS}	Data & strobe output skew & jitter	-	0	150	ps
t_{SPW3_LVDS}	Input data bit period	-	5	500	ns
t_{SPW4_LVDS}	Data & strobe input skew, jitter & hold	-	-	500	ps
t_{SPW5_LVDS}	Data & strobe edge separation	-	2.5	-	ns

Note 1: Only applies when characterizing one SpaceWire link at a time

Note 2: V_{DDIO} = 3.3 V +/- 0.3 V only applies with external LVDS voltage reference. See table 134 under recommended operating conditions for V_{DDIO} restrictions when using internal LVDS voltage reference.

15.7.7 SpaceWire LVTTL interface timing

The timing waveforms are shown in figure 24, and the timing parameters are defined in table 146.

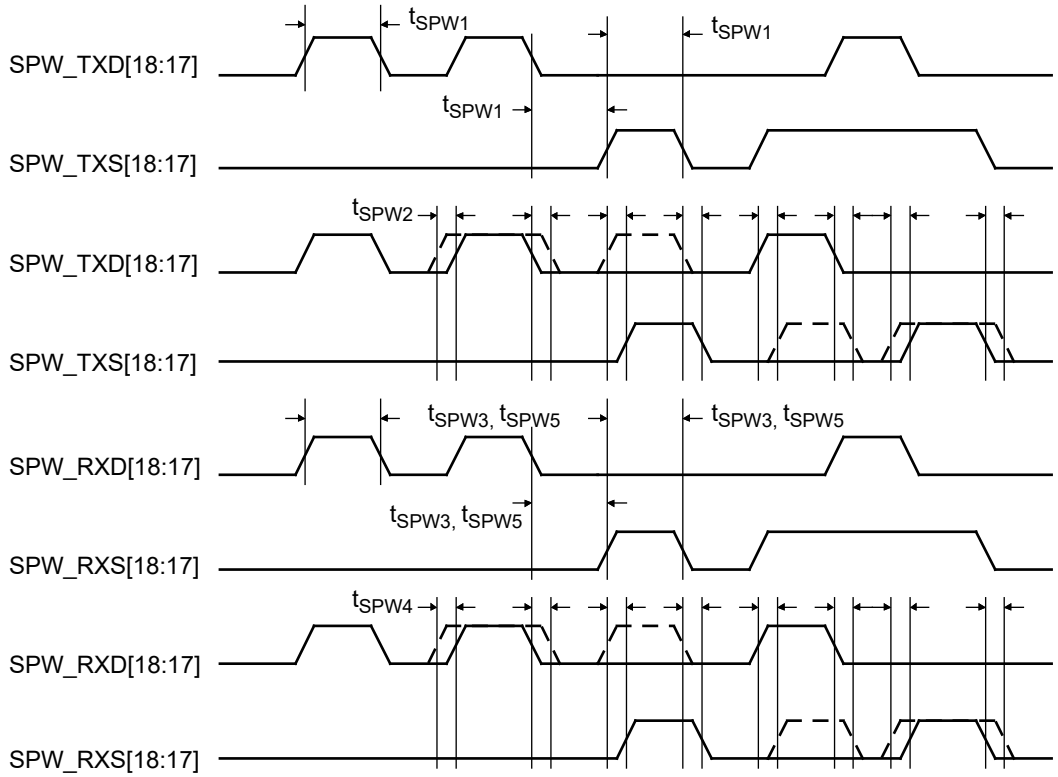


Figure 24. SpaceWire LVTTL interface timing waveforms

Table 146. SpaceWire LVTTL interface timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	Parameter	Reference edge	Min	Max	Unit
t_{SPW1_LVTTL}	Output data bit period	-	5	500	ns
t_{SPW2_LVTTL}	Data & strobe output skew & jitter	-	0	150	ps
t_{SPW3_LVTTL}	Input data bit period	-	5	500	ns
t_{SPW4_LVTTL}	Data & strobe input skew, jitter & hold	-	-	500	ps
t_{SPW5_LVTTL}	Data & strobe edge separation	-	2.5	-	ns

Note 1: Only applies when characterizing one SpaceWire link at a time

15.7.8 JTAG interface timing

The timing waveforms are shown in figure 25, and the timing parameters are defined in table 147.

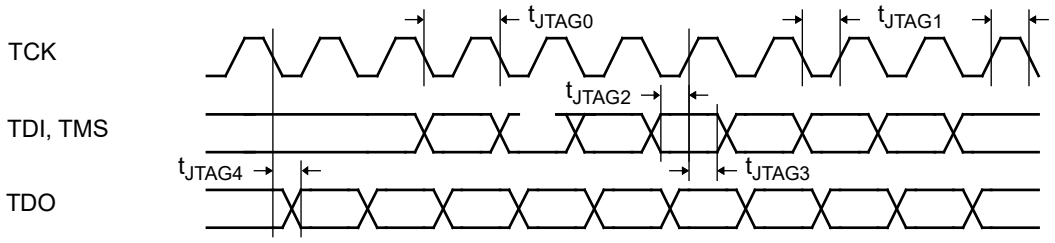


Figure 25. JTAG interface timing waveforms

*Table 147.*JTAG interface timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	Parameter	Reference edge	Min	Max	Unit
t_{JTAG0}	Clock period	-	100 [1)]	-	ns
t_{JTAG1}	Clock high/low pulse length	-	40	-	ns
t_{JTAG2}	Input to clock setup	Rising TCK edge	10	-	ns
t_{JTAG3}	Input to clock hold	Rising TCK edge	10	-	ns
t_{JTAG4}	Clock to output delay	Falling TCK edge	2	20	ns

Note 1: Minimum clock period t_{JTAG0} = 4 x t_{CLK0}

15.7.9 UART interface timing

The timing waveforms are shown in figure 26, and the timing parameters are defined in table 148.

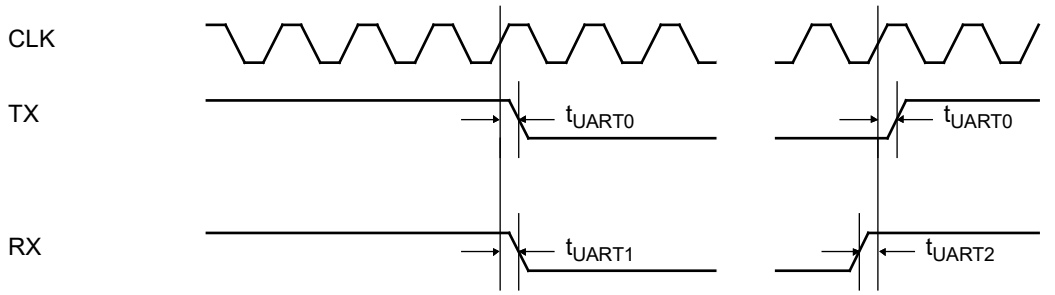


Figure 26. UART interface timing waveforms

*Table 148.*UART interface timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	Parameter	Reference edge	Min	Max	Unit
t_{UART0}	Clock to output delay	Rising CLK edge	0	20	ns
t_{UART1}	Input to clock hold	Rising CLK edge	-	-	ns
t_{UART2}	Input to clock setup	Rising CLK edge	-	-	ns

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

15.7.10 General purpose I/O timing

The timing waveforms are shown in figure 27, and the timing parameters are defined in table 149.

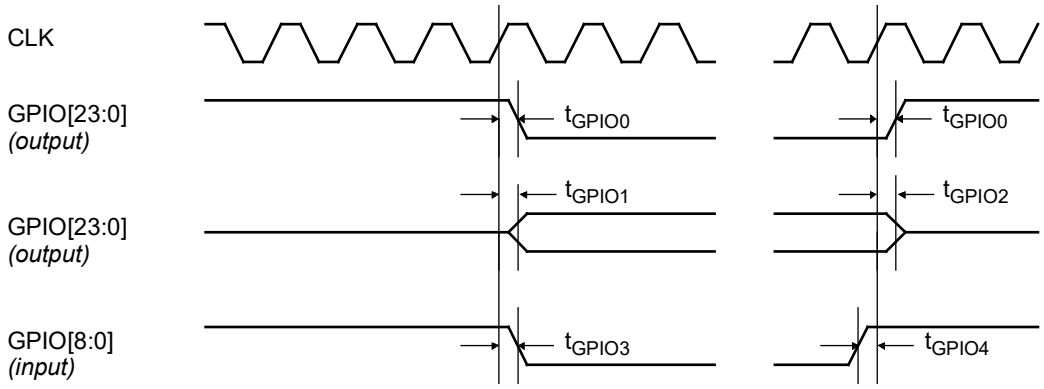


Figure 27. General purpose I/O timing waveforms

*Table 149.*General purpose I/O timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	Parameter	Reference edge	Min	Max	Unit
t_{GPIO0}	Clock to output delay	Rising CLK edge	0	20	ns
t_{GPIO1}	Clock to non-tri-state delay [1)]	Rising CLK edge	0	25	ns
t_{GPIO2}	Clock to tri-state delay [1)]	Rising CLK edge	0	50	ns
t_{GPIO3} [2)]	Input to clock hold	Rising CLK edge	3		ns
t_{GPIO4} [2)]	Input to clock setup	Rising CLK edge	3		ns

Note 1: Outputs are pulled with 500ohm to V_{DDIO} or 0V.

Note 2: Requirement only applies to GPIO[8:0] when they are in auxiliary time- / interrupt-code mode (external pin GPIOSEL = 0). Other GPIOs, and GPIO[8:0] in GPIO mode (GPIOSEL = 1), are re-synchronized internally, and do not have to meet any setup or hold requirements.

15.7.11 Auxiliary time- / interrupt-code timing

The timing waveforms are shown in figure 28, and the timing parameters are defined in table 150.

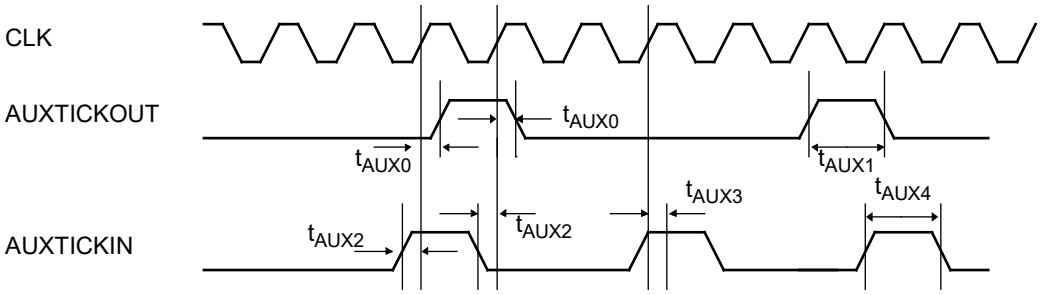


Figure 28. Auxiliary time- / interrupt-code timing waveforms

Table 150. Auxiliary time- / interrupt-code timing parameters (V_{DD} = 1.8 V +/- 0.15 V, V_{DDIO} = 3.3 V +/- 0.3 V, T_{case} = -55°C to +125°C)

Name	Parameter	Reference edge	Min	Max	Unit
t_{AUX0}	Clock to output delay	Rising CLK edge	0	20	ns
t_{AUX1}	Output pulse width	Rising CLK edge	t_{CLK0} - 5	t_{CLK0} + 20	ns
t_{AUX2}	Input to clock setup 1)	Rising CLK edge	3		ns
t_{AUX3}	Input to clock hold 1)	Rising CLK edge	3		ns
t_{AUX4}	Input pulse width 2)	Rising CLK edge	t_{CLK0} + 6		ns

Note 1: Only applies when auxiliary time-code / distributed interrupt code interface is in synchronous mode.

Note 2: Only applies when auxiliary time-code / distributed interrupt code interface is in asynchronous mode.

16 Mechanical description

16.1 Component and package

GR718B is provided in a CQFP256 package.

16.2 Pin assignment

The pin assignment in table 151 shows the implementation characteristics of each signal indicating how each pin has been configured in terms of electrical levels, drive capability and internal pull-up or pull-down.

*Table 151.*Pin assignment

Name	I/O	Pin CQFP256	Level [3)]	Drive [mA]	Pull	Active	Note
RESETn	in	218	LVTTL			Low	Reset input with Schmitt trigger
TESTEN[0]	in	13	LVTTL		Down	High	Test mode enable
TESTEN[1]	in	9	LVTTL		Down	High	Test mode enable
CLK	in	220	LVTTL			-	System clock
SPWCLK	in	222	LVTTL			-	SpaceWire clock
LOCK	out	8	LVTTL	2		High	Internal SpaceWire clock lock
IRQ	out	211	LVTTL	2		High	Status change interrupt
TX	out	209	LVTTL	6		Low	Debug UART data transmit
RX	in	198	LVTTL			Low	Debug UART data receive
TCK	in	199	LVTTL			-	Debug JTAG clock
TMS	in	200	LVTTL			High	Debug JTAG mode
TDI	in	201	LVTTL			High	Debug JTAG input
TDO	out	208	LVTTL	2		High	Debug JTAG output
SPW_RXDp[1]	in	23	LVDS			High	SpaceWire data
SPW_RXDn[1]	in	24	LVDS			Low	
SPW_RXSp[1]	in	21	LVDS			High	SpaceWire strobe
SPW_RXSn[1]	in	22	LVDS			Low	
SPW_TXDp[1]	out	17	LVDS			High	SpaceWire data
SPW_TXDn[1]	out	16	LVDS			Low	
SPW_TXSp[1]	out	19	LVDS			High	SpaceWire strobe
SPW_TXSn[1]	out	18	LVDS			Low	
SPW_RXDp[2]	in	33	LVDS			High	SpaceWire data
SPW_RXDn[2]	in	34	LVDS			Low	
SPW_RXSp[2]	in	31	LVDS			High	SpaceWire strobe
SPW_RXSn[2]	in	32	LVDS			Low	
SPW_TXDp[2]	out	28	LVDS			High	SpaceWire data
SPW_TXDn[2]	out	27	LVDS			Low	
SPW_TXSp[2]	out	30	LVDS			High	SpaceWire strobe
SPW_TXSn[2]	out	29	LVDS			Low	

*Table 151.*Pin assignment

Name	I/O	Pin CQFP256	Level [3)]	Drive [mA]	Pull	Active	Note
SPW_RXDp[3]	in	43	LVDS			High	SpaceWire data
SPW_RXDn[3]	in	44	LVDS			Low	
SPW_RXSp[3]	in	41	LVDS			High	SpaceWire strobe
SPW_RXSn[3]	in	42	LVDS			Low	
SPW_TXDp[3]	out	38	LVDS			High	SpaceWire data
SPW_TXDn[3]	out	37	LVDS			Low	
SPW_TXSp[3]	out	40	LVDS			High	SpaceWire strobe
SPW_TXSn[3]	out	39	LVDS			Low	
SPW_RXDp[4]	in	53	LVDS			High	SpaceWire data
SPW_RXDn[4]	in	54	LVDS			Low	
SPW_RXSp[4]	in	51	LVDS			High	SpaceWire strobe
SPW_RXSn[4]	in	52	LVDS			Low	
SPW_TXDp[4]	out	46	LVDS			High	SpaceWire data
SPW_TXDn[4]	out	45	LVDS			Low	
SPW_TXSp[4]	out	48	LVDS			High	SpaceWire strobe
SPW_TXSn[4]	out	47	LVDS			Low	
SPW_RXDp[5]	in	61	LVDS			High	SpaceWire data
SPW_RXDn[5]	in	62	LVDS			Low	
SPW_RXSp[5]	in	59	LVDS			High	SpaceWire strobe
SPW_RXSn[5]	in	60	LVDS			Low	
SPW_TXDp[5]	out	56	LVDS			High	SpaceWire data
SPW_TXDn[5]	out	55	LVDS			Low	
SPW_TXSp[5]	out	58	LVDS			High	SpaceWire strobe
SPW_TXSn[5]	out	57	LVDS			Low	
SPW_RXDp[6]	in	75	LVDS			High	SpaceWire data
SPW_RXDn[6]	in	76	LVDS			Low	
SPW_RXSp[6]	in	73	LVDS			High	SpaceWire strobe
SPW_RXSn[6]	in	74	LVDS			Low	
SPW_TXDp[6]	out	67	LVDS			High	SpaceWire data
SPW_TXDn[6]	out	66	LVDS			Low	
SPW_TXSp[6]	out	69	LVDS			High	SpaceWire strobe
SPW_TXSn[6]	out	68	LVDS			Low	
SPW_RXDp[7]	in	85	LVDS			High	SpaceWire data
SPW_RXDn[7]	in	86	LVDS			Low	
SPW_RXSp[7]	in	83	LVDS			High	SpaceWire strobe
SPW_RXSn[7]	in	84	LVDS			Low	
SPW_TXDp[7]	out	78	LVDS			High	SpaceWire data
SPW_TXDn[7]	out	77	LVDS			Low	
SPW_TXSp[7]	out	80	LVDS			High	SpaceWire strobe
SPW_TXSn[7]	out	79	LVDS			Low	

*Table 151.*Pin assignment

Name	I/O	Pin CQFP256	Level [3)]	Drive [mA]	Pull	Active	Note
SPW_RXDp[8]	in	95	LVDS			High	SpaceWire data
SPW_RXDn[8]	in	96	LVDS			Low	
SPW_RXSp[8]	in	93	LVDS			High	SpaceWire strobe
SPW_RXSn[8]	in	94	LVDS			Low	
SPW_TXDp[8]	out	88	LVDS			High	SpaceWire data
SPW_TXDn[8]	out	87	LVDS			Low	
SPW_TXSp[8]	out	90	LVDS			High	SpaceWire strobe
SPW_TXSn[8]	out	89	LVDS			Low	
SPW_RXDp[9]	in	105	LVDS			High	SpaceWire data
SPW_RXDn[9]	in	106	LVDS			Low	
SPW_RXSp[9]	in	103	LVDS			High	SpaceWire strobe
SPW_RXSn[9]	in	104	LVDS			Low	
SPW_TXDp[9]	out	100	LVDS			High	SpaceWire data
SPW_TXDn[9]	out	99	LVDS			Low	
SPW_TXSp[9]	out	102	LVDS			High	SpaceWire strobe
SPW_TXSn[9]	out	101	LVDS			Low	
SPW_RXDp[10]	in	115	LVDS			High	SpaceWire data
SPW_RXDn[10]	in	116	LVDS			Low	
SPW_RXSp[10]	in	113	LVDS			High	SpaceWire strobe
SPW_RXSn[10]	in	114	LVDS			Low	
SPW_TXDp[10]	out	110	LVDS			High	SpaceWire data
SPW_TXDn[10]	out	109	LVDS			Low	
SPW_TXSp[10]	out	112	LVDS			High	SpaceWire strobe
SPW_TXSn[10]	out	111	LVDS			Low	
SPW_RXDp[11]	in	125	LVDS			High	SpaceWire data
SPW_RXDn[11]	in	126	LVDS			Low	
SPW_RXSp[11]	in	123	LVDS			High	SpaceWire strobe
SPW_RXSn[11]	in	124	LVDS			Low	
SPW_TXDp[11]	out	118	LVDS			High	SpaceWire data
SPW_TXDn[11]	out	117	LVDS			Low	
SPW_TXSp[11]	out	120	LVDS			High	SpaceWire strobe
SPW_TXSn[11]	out	119	LVDS			Low	
SPW_RXDp[12]	in	138	LVDS			High	SpaceWire data
SPW_RXDn[12]	in	139	LVDS			Low	
SPW_RXSp[12]	in	136	LVDS			High	SpaceWire strobe
SPW_RXSn[12]	in	137	LVDS			Low	
SPW_TXDp[12]	out	130	LVDS			High	SpaceWire data
SPW_TXDn[12]	out	129	LVDS			Low	
SPW_TXSp[12]	out	132	LVDS			High	SpaceWire strobe
SPW_TXSn[12]	out	131	LVDS			Low	

*Table 151.*Pin assignment

Name	I/O	Pin CQFP256	Level [3)]	Drive [mA]	Pull	Active	Note
SPW_RXDp[13]	in	148	LVDS			High	SpaceWire data
SPW_RXDn[13]	in	149	LVDS			Low	
SPW_RXSp[13]	in	146	LVDS			High	SpaceWire strobe
SPW_RXSn[13]	in	147	LVDS			Low	
SPW_TXDp[13]	out	143	LVDS			High	SpaceWire data
SPW_TXDn[13]	out	142	LVDS			Low	
SPW_TXSp[13]	out	145	LVDS			High	SpaceWire strobe
SPW_TXSn[13]	out	144	LVDS			Low	
SPW_RXDp[14]	in	158	LVDS			High	SpaceWire data
SPW_RXDn[14]	in	159	LVDS			Low	
SPW_RXSp[14]	in	156	LVDS			High	SpaceWire strobe
SPW_RXSn[14]	in	157	LVDS			Low	
SPW_TXDp[14]	out	151	LVDS			High	SpaceWire data
SPW_TXDn[14]	out	150	LVDS			Low	
SPW_TXSp[14]	out	153	LVDS			High	SpaceWire strobe
SPW_TXSn[14]	out	152	LVDS			Low	
SPW_RXDp[15]	in	166	LVDS			High	SpaceWire data
SPW_RXDn[15]	in	167	LVDS			Low	
SPW_RXSp[15]	in	164	LVDS			High	SpaceWire strobe
SPW_RXSn[15]	in	165	LVDS			Low	
SPW_TXDp[15]	out	161	LVDS			High	SpaceWire data
SPW_TXDn[15]	out	160	LVDS			Low	
SPW_TXSp[15]	out	163	LVDS			High	SpaceWire strobe
SPW_TXSn[15]	out	162	LVDS			Low	
SPW_RXDp[16]	in	176	LVDS			High	SpaceWire data
SPW_RXDn[16]	in	177	LVDS			Low	
SPW_RXSp[16]	in	174	LVDS			High	SpaceWire strobe
SPW_RXSn[16]	in	175	LVDS			Low	
SPW_TXDp[16]	out	171	LVDS			High	SpaceWire data
SPW_TXDn[16]	out	170	LVDS			Low	
SPW_TXSp[16]	out	173	LVDS			High	SpaceWire strobe
SPW_TXSn[16]	out	172	LVDS			Low	
SPW_RXD[17]	in	191	LVTTL			High	SpaceWire data
SPW_RXS[17]	in	190	LVTTL			High	SpaceWire strobe
SPW_TXD[17]	out	186	LVTTL	6		High	SpaceWire data
SPW_TXS[17]	out	187	LVTTL	6		High	SpaceWire strobe
SPW_RXD[18]	in	6	LVTTL			High	SpaceWire data
SPW_RXS[18]	in	5	LVTTL			High	SpaceWire strobe
SPW_TXD[18]	out	1	LVTTL	6		High	SpaceWire data
SPW_TXS[18]	out	2	LVTTL	6		High	SpaceWire strobe
SPWEN[17]	out	192	LVTTL	2		High	Transceiver enable for SpW link 17
SPWEN[18]	out	7	LVTTL	2		High	Transceiver enable for SpW link 18

*Table 151.*Pin assignment

Name	I/O	Pin CQFP256	Level [3)]	Drive [mA]	Pull	Active	Note
AUXTICKIN	in	197	LVTTL			High	Auxiliary Tick input
AUXTICKOUT	out	210	LVTTL	2		High	Auxiliary Tick output
GPIO[0]	inout	227	LVTTL	2		-	See section 5 for functionality
GPIO[1]	inout	228	LVTTL	2		-	
GPIO[2]	inout	229	LVTTL	2		-	
GPIO[3]	inout	230	LVTTL	2		-	
GPIO[4]	inout	231	LVTTL	2		-	
GPIO[5]	inout	232	LVTTL	2		-	
GPIO[6]	inout	233	LVTTL	2		-	
GPIO[7]	inout	234	LVTTL	2		-	
GPIO[8]	inout	237	LVTTL	2		-	
GPIO[9]	inout	238	LVTTL	2		-	
GPIO[10]	inout	239	LVTTL	2		-	
GPIO[11]	inout	240	LVTTL	2		-	
GPIO[12]	inout	241	LVTTL	2		-	
GPIO[13]	inout	242	LVTTL	2		-	
GPIO[14]	inout	243	LVTTL	2		-	
GPIO[15]	inout	244	LVTTL	2		-	
GPIO[16]	inout	247	LVTTL	2		-	
GPIO[17]	inout	248	LVTTL	2		-	
GPIO[18]	inout	249	LVTTL	2		-	
GPIO[19]	inout	250	LVTTL	2		-	
GPIO[20]	inout	251	LVTTL	2		-	
GPIO[21]	inout	252	LVTTL	2		-	
GPIO[22]	inout	253	LVTTL	2		-	
GPIO[23]	inout	254	LVTTL	2		-	
GPIOSEL	in	224	LVTTL			High	Function selector for GPIO
SPI_MISO	in	202	LVTTL			-	SPI master in slave out
SPI_MOSI	out	205	LVTTL	2		-	SPI master out slave in
SPI_SCK	out	207	LVTTL	2		-	SPI clock
SPI_SLVSEL	out	206	LVTTL	2		-	SPI slave select

*Table 151.*Pin assignment

Name	I/O	Pin CQFP256	Level [3]	Drive [mA]	Pull	Active	Note
SPWCLKSEL[0]	in	10	LVTTL			High	SpaceWire clock rate selector
SPWCLKSEL[1]	in	11	LVTTL			High	SpaceWire clock rate selector
SPWCLKSEL[2]	in	12	LVTTL			High	SpaceWire clock rate selector
SPWCLKDIV[0]	in	180	LVTTL			High	SpaceWire clock divider LSB
SPWCLKDIV[1]	in	181	LVTTL			High	
SPWCLKDIV[2]	in	182	LVTTL			High	
SPWCLKDIV[3]	in	183	LVTTL			High	
SPWCLKDIV[4]	in	184	LVTTL			High	
SPWCLKDIV[5]	in	185	LVTTL			High	SpaceWire clock divider MSB
CFGLOCK	in	219	LVTTL			High	Lock configuration port
SPILLIFNOTREADY	in	195	LVTTL			High	Spill if not ready
PNPEN	in	196	LVTTL			High	Plug-and-play enable
LINKSTARTONREQ	in	223	LVTTL			High	Link start on request
AUTODCONNECT	in	221	LVTTL			High	Automatic disconnect
STATICROUTEEN	in	212	LVTTL			High	Static routing enable
SELVREF	in	20	LVTTL			High	Select external LVDS voltage reference. Digital high selects external LVDS voltage reference and digital low input will select internal generated voltage reference.
LVDSREF	out	72	-			-	LVDS reference. Connect to ground via resistance of 6k8 Ohm +/-4%. The resistor shall be placed close to the pin. The total capacitance on the PCB shall be less than 5pF.
VREFEXT	in	135	-			-	External analog reference voltage used by LVDS Drivers, e.g. used for common mode regulation. This reference voltage must be in the range of 1.25 V ± 4%, see table 134, and it will bypass the internally generated reference voltage when SELVREF is set to logic high. If not selected (i.e. SELVREF set to logic low) tie this pin to ground. The external analog reference voltage input has leakage current according to I_{ILEAK_1} in table 136, and it has ESD protection diodes to VDDIO and GND, which is the same electrical characteristics as for standard LVTTL inputs. The absolute maximum ratings are according to VIN_LVTTL and IIN in table 133.
GND	-	4, 15, 26, 36, 50, 64, 65, 71, 82, 92, 98, 108, 122, 127, 133, 140, 154, 168, 178, 188, 193, 203, 213, 217, 225, 235, 245, 255					Ground [1]

*Table 151.*Pin assignment

Name	I/O	Pin CQFP256	Level [3]	Drive [mA]	Pull	Active	Note
VDDIO	-	3, 14, 49, 63, 81, 97, 107, 128, 141, 155, 169, 189, 204, 226, 246, 256					3.3 V supply [1]
VDD	-	25, 35, 70, 91, 121, 134, 179, 194, 216, 236					1.8 V supply [1]
VSSA	-	214					Ground [2]
VDDA	-	215					1.8 V supply [2]

Note 1: External decoupling should be added as close as possible to the supply pins.

Note 2: To improve PLL jitter characteristics, external decoupling between VDDA and VSSA should be added as close as possible to the pins. When the PLL is used to generate high-speed SpaceWire clocks, it is recommended to insert a local LP filter for VDDA, such that the supply noise becomes lower than ~1mV_rms between VDDA and VSSA pins (LP filter ground directly connected to VSSA pin by short wire). The total series resistance must not be larger than approximately 2ohm due to DC voltage drop. For example, assume a 1.8V digital supply on PCB that has ~100mV peak-to-peak disturbance (and roughly one third expressed in Vrms) at 100kHz or higher. A 1st order RC filter needs a cut-off frequency of fc<~100kHz/((100mV/3)/1mV)=~3kHz, which is fulfilled by e.g. 2.2ohm and 33uF (ESR<2.2ohm/((100mV/3)/1mV)=0.06ohm). A 2nd order RC filter needs fc<~100kHz/sqrt((100mV/3)/1mV)=~17kHz. Using 1ohm for each resistor gives 10uF (ESR<~1ohm/sqrt((100mV/3)/1mV)=~0.17ohm) per capacitor.

Note 3: No on-chip support for cold spare or fail safe for any input pr output pins.

16.3 Mechanical package drawings

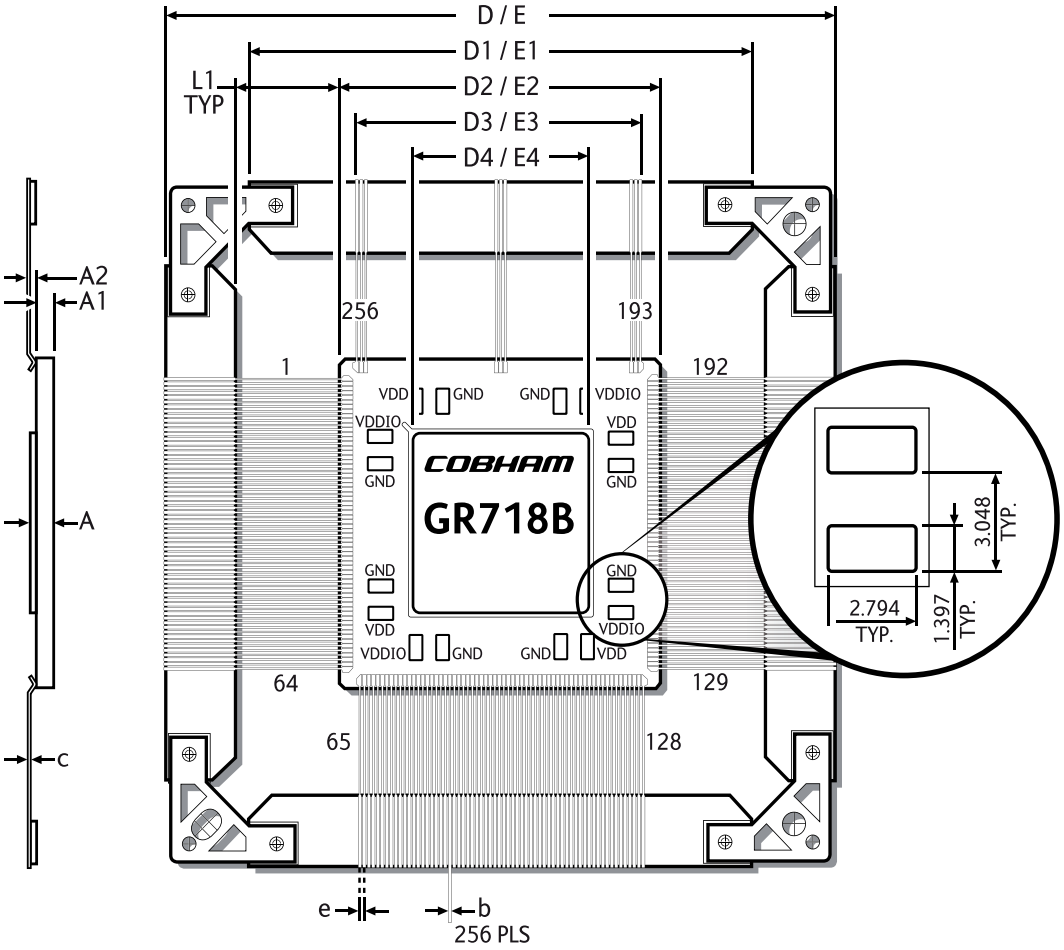


Figure 29. Package top view

Table 152. Package dimensions

Name	Parameter	Min	TYP	Max	Unit
A		2.44		4.00	mm
A1		2.11		2.56	mm
A2		0.15		0.55	mm
b		0.15		0.25	mm
c		0.10		0.20	mm
D/E			74.1		mm
D1/E1			57		mm
D2/E2		35.64		36.36	mm
D3/E3			31.5		mm
D4/E4		19.35		20.27	mm
e			0.50		mm
L1		11.35		12.15	mm

Note 1: The seal ring and lid are connected to ground

Note 2: Mass of case, including the lead frames, shall be 20±1 grams.

Note 3: Eight pairs of pads connected to GND, VDD and VDDIO, respectively, are placed on top of the package body. These are only intended for prototyping purpose.

17 Ordering information

Ordering information is provided in table 153 and a legend is provided in table 154.

*Table 153.*Ordering information, available models

Product	Description
GR718B-MS-CQ256	Flight model
GR718B-MP-CQ256	Electrical qualification model (prototype)
GR718B-CP-CQ256	Engineering model (prototype)

*Table 154.*Ordering legend

Designator	Option	Description
Product	GR718B	Radiation-Tolerant 18x SpaceWire Router rev B
Temperature Range	M	-55°C to +125°C (Military range)
	C	0°C to +70°C (Commercial range)
Screening Level	S	Space grade
	P	Prototype grade
Package Type	CQ	Ceramic Quad Flat Pack (CQFP) 1)
Lead Count	256	Number of leads

Note 1: Gold lead finish

Frontgrade Gaisler AB
Kungsgatan 12
411 19 Göteborg
Sweden
www.frontgrade.com/gaisler
sales@gaisler.com
T: +46 31 7758650