SpaceWire FAQ

Application note 2015-10-27

Doc. No GRLIB-AN-0001

Issue 1.0

CHANGE RECORD

Issue	Date	Section / Page	Description
1.0	2015-10-27		First issue of this document. Content moved from GRLIB's doc/spacewire.pdf

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Scope of the Document

This document collects frequently asked questions for Cobham Gaisler SpaceWire products.

2 APPLICATION NOTES / FREQUENTLY ASKED QUESTIONS

2.1 Configuring LVDS terminations for SpaceWire on Cobham Gaisler boards

2.1.1 Introduction

Cobham Gaisler provides SpaceWire IP cores/components on a wide range of FPGA/ASIC technologies. To evaluate these components a set of mezzanine boards with SpaceWire connectors are provided that can be used together with the different motherboards hosting the FPGA/ASIC. Due to varying support for correct termination of the LVDS drivers and receivers in the FPGA or on the motherboard this can easily lead to incorrectly matched connections with a failing or unstable link as a result. This application will gives an overview of the different options on the GR-RTAX-MEZZ, used together with the GR-CPCI-AX and GR-CPCI-XC4V FPGA boards.

2.1.2 Difference in LVDS drivers

Although the mezzanine board is called “GR-RTAX-MEZZ” it can be used with both the Actel Axcelerator GR-CPCI-AX and Xilinx Virtex 4 GR-CPCI-XC4V development boards.

A big difference between the AX/RTAX FPGAs and the Virtex 4 is that the former two require external termination on both the LVDS receiver and transmitter to function. The receiver requires a 100 ohm parallel resistor while the transmitter must have a 165 ohm resistor in series on the P and N lines followed by a 140 ohm resistor in parallel. More details on the Actel LVDS functionality can be found in the “Using LVDS for Actel’s Axcelerator and RTAX-S/SL devices” application note.

The Virtex 4 LVDS outputs do not require any external termination while the inputs can be configured to use either internal terminations or external. If the DIFF_TERM property is set on the LVDS input pair in the ucf file then the XC4V will contain a 100 ohm parallel resistor internally. Another possibility is to use the LVDS_25_DCI or LVDSEXT_25_DCI I/O standards in which case split terminations are used providing an equivalent of 100 ohm in parallel. This configuration requires bias resistors connected to the VRN and VRP pins. For more detailed information on Virtex4 LVDS termination see the Virtex 4 datasheet.

2.1.3 Board configurations

The GR-RTAX-MEZZ only provides MDM9 connectors with direct paths to the FPGA on the motherboard. The output path can optionally have 140 ohm parallel resistors mounted and the input paths can have 100 ohm resistors in parallel.

The GR-CPCI-XC4V board has matched traces for the LVDS signals but does not have and does not provide the possibility to put any terminations on the board. The GR-CPCI-AX board also has matched traces and has locations for mounting 165 ohm and 140 ohm resistors in the output path and 100 ohm resistors in the input path. These are mounted on a case by case basis.

2.1.4 Combining the mezzanine and motherboards

The properties of the boards presented above results in several different situations. When using the GR-RTAX-MEZZ with the GR-CPCI-XC4V the 140 ohm output resistances must not be mounted on the mezzanine. The 100 ohm input termination can be mounted provided the FPGA has not been configured with internal terminations. It is recommended that the internal terminations is used by setting the DIFF_TERM property.

When used with the GR-CPCI-AX board there are two different situations. If terminations are mounted on the motherboard in both directions as outlined in the previous section the terminations must not be present on the mezzanine. In the receive direction the 100 ohm parallel resistance can be placed on the mezzanine but in the output path the mezzanine only provides a place for 140 ohm resistors in parallel so in that case the series 165 ohm resistors would have to be placed on the motherboard.

2.2 Implementing Broadcast/Multicast

Broadcast/Multicast is used by activating more than one port for a physical or logical address (it is normally only used with logical addresses but the GRSPWROUTER provides the flexibility to use it with both). This is called packet distribution in SpaceWire. So for example logical address 58 can be used as a broadcast address. The routing table in each router in the system has to be configured in a way so that all nodes in the system receives the packet. All routers have to set the routing table entry 58 so that an incoming packet with address 58 is forwarded to all ports connected to nodes. Some routers also need to forward the packet to ports connected to other routers but care has to be taken to avoid infinite loops where a packet circles between two or more routers forever. This restriction also means that one has to define a source node of the broadcast packet. Its impossible to have two sources of broadcast packets (using the same logical address) on different routers without creating loops in the system. It is possible to have multiple sources if they are all located on a single router.

In the same way multicast can be implemented the only difference from broadcast being that not all ports connected to nodes in a router are enabled for the multicast address.

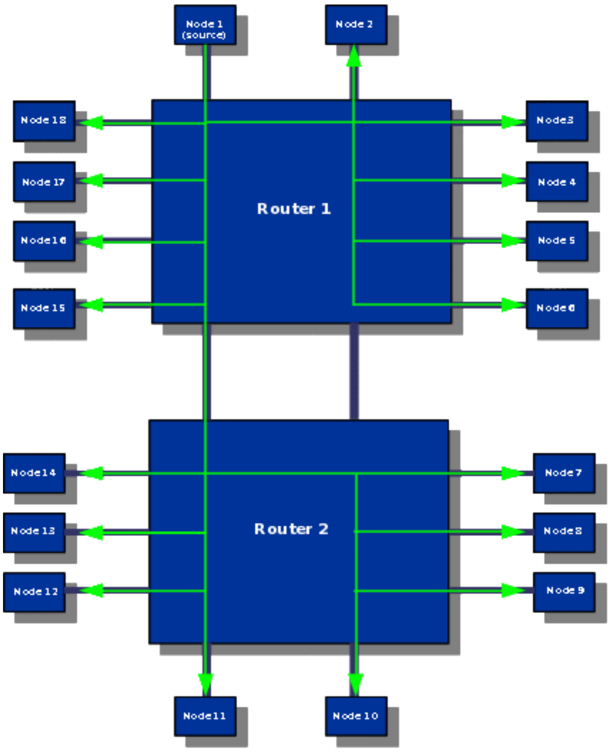

Figure 1: Example of Broadcast in a two router system

Figure 1 shows an example of a system with two routers where packet distribution is used. The routers have 10 ports each of which two on each are used for connecting the routers together while the other ports are connected to nodes. The router ports are numbered from 1 to 10 in clockwise order starting from the leftmost one on the upper side.

Node 1 is the source of the broadcast packets and to enable broadcast to all other nodes in the system using address 58 the following configuration has to be done. Address 0xE8 in the configuration port (port setup register for address 58) for router 1 is written with the value 0x5FD. If it is desired that the broadcast packet is received at the source as well the value 0x5FF should be written instead. Bits with indexes 1-10 each correspond to the port with the same number. Setting an index to 1 enables the corresponding port. Bit 0 selects between group adaptive routing (0) and packet distribution (1) which is set to 1 in this case. All bits are set to 1 except index 9 (and index 0 in the second case) since the packet should only be propagated to the second router through one port.

The logical address must also be enabled which is done by writing the routing table entry at address 0x4E8 with the value 0x4. This last write also sets header deletion to 0 (disabled) and priority to 0 (standard priority).

The same procedure is done with the second router and the routing table entry at address 0x4E8 is written with the same value (0x4) as for router 1. The port setup register at address 0xE8 is written with 0x7F9 which enables all ports except 1 and 2 which are connected to router 1 and the packet must not be sent back to avoid loops.

Writing to the configuration can be done from any port using RMAP commands or the AHB slave interface (if present). See the GRSPWROUTER section in the SpaceWire cores user manual for details on using the configuration port.

2.3 Lockup problem with Packet Distribution

If the destination of two ports is the same when using packet distribution and they use the same receive channel then the packet size is limited by the amount of buffering available to avoid a deadlock situation. An example is if packet distribution is used to send from port x and the address has packet distribution enabled to ports y and z. Y and z are connected to the same destination node with a common receive channel. Y gets access first but transmission will deadlock when the packet size reach the amount of buffering. This happens because due to wormhole routing each character has to be sent on all active packet distribution ports simultaneously and since y blocks the destination z has to wait until y has transmitted the whole packet. Y has to wait for more data from the source since the source has to wait for z getting access to the destination thus creating a deadlock situation.

A special case of this problem is when transmitting from one port going to several ports which are loopbacked back into the router (with header deletion) and then all the packets go the same destination. This situation is illustrated in figure 21 where SpaceWire port 4 is transmitting a distributed packet to SpaceWire ports 5, 6 and 7. The latter three ports are loopbacked to SpaceWire ports 3, 2 and 1 respectively. The packet distribution address is removed with header deletion when being routed from port 4 and all three copies are then addressed to AMBA-AHB port 1. The packet from port 1 gets access first in this case and because the packet size is bigger than the available buffering in each port deadlock occurs. Port 4 wants to write more data to port 5 but it cannot because the buffers in port 6 and 7 are full and with wormhole routing a character is written at the same time to all destination ports.

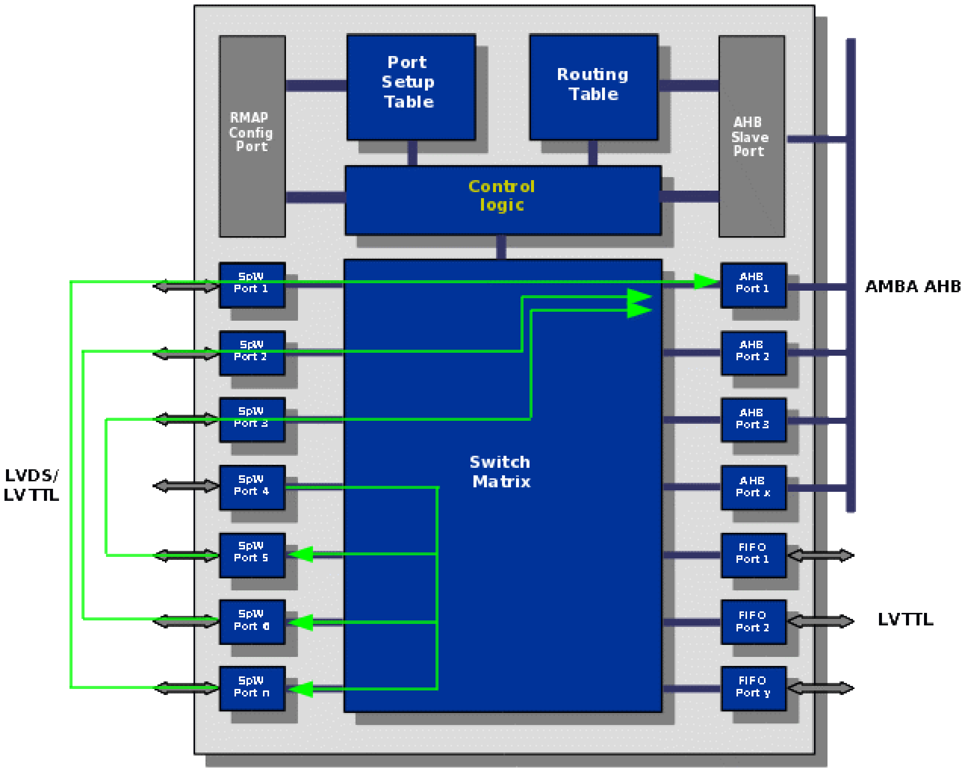

Figure 2: Illustration of the lock-up special case

2.4 Transmitter frequency division in DDR mode

When the transmitter is configured in DDR mode the bitrate is twice the actual clock frequency. In this case the divisor value should be set corresponding to the bitrate not the clock frequency. In the same way as for SDR outputs the effective value is the value in the register incremented by one. So value 0 means the bitrate is divided by 1, 1 divides by 2 and so on. For example if the frequency is 100 MHz resulting in 200 Mbit/s data rate the initialization divisor should be set to 19 to get the 10 Mbit/s startup rate. This will also set the FSM timeouts (6.4 and 12.8 us) correctly. Although the divisor is calculated corresponding to the bitrate the physical limitations are still imposed by the actual clock. This means that with a bitrate of 50 Mbit/s and a 25 MHz clock it is not possible to get the exactly 10 Mbit/s in initialization rate since 25 is not an integer multiple of 10. The FSM timeouts will also be incorrect.

In summary, regardless of output configuration the clock must be an integer multiple of 10 and the divisor is calculated with respect to the bitrate.

2.5 Standalone router startup to nominal operation

After power-on and reset the router has been set in the desired startup state using configuration pins. The SpW links should have been configured to start at 10 Mbit/s and are set in autostart mode so that they automatically enter run-state if another correctly configured link is connected to it. Initially only physical addressing can be used which is also used to access the configuration port (the configuration port is only allowed to be accessed using physical addressing). So if only physical addressing is used no more configuration has to be done and the router is immediately operational. The links will operate in 10 Mbit/s also in run-state after reset (uses the same divisor as the initialization reset value). If this needs to be changed one writes the run-state divisor for the desired port through the configuration port with an RMAP command.

If logical addressing is needed this is also setup through the configuration port by writing the routing table and port setup registers.

FIFO ports are also operational after reset. They can be set in either bridge mode or normal mode after reset to achieve the desired operation after reset. The bridge mode is used to automatically interconnect two router chips with automatical flow of packets and time-codes between the routers through the FIFO ports without any glue logic. In normal mode an external entity can write and read data and time-codes to/from the FIFO port.

AMBA ports typically do not need any configuration from the router side. They are controlled using for example a processor from the external side of the port.

On the SpaceWire packet level all three port types look are the same and packets flow in the same way through the router.

2.6 How is the router configured?

It is configured through port 0 which is the configuration port. It is specified in the SpaceWire standard that the configuration port must be port 0 (but it does not say that one has to be available). The router uses the RMAP protocol for configuration. All configuration registers are located in the RMAP address space and are accesses using RMAP write, read and rmw commands.

Some of the register bits have hardcoded reset values while other bits have reset values set from signals (which can be assigned to pins in an implementations). It is thus possible to select different startup configurations for the router without software intervention.

The router will be operable at reset but some features (such as logical addressing, packet distribution, group adaptive routing) cannot be used until software has configured it through the configuration port.

In systems where the router is connected to an AMBA AHB bus and has an AHB slave interface it can be used to directly read or write the configuration space from the AMBA bus. It is much more efficient for a CPU to configure the router in this way than sending RMAP packets on an AMBA port for example.

Which ports can configure it? Normally all ports can access the configuration port and thus configure the router. There is an external signal called cfglock which disabled accesses to the configuration port from all ports except port 1. Configuration accesses can also be disabled from ports individually through registers in the configuration port. Writes to the configuration port can be disabled for all ports by writing a bit in a configuration port register. This disables write accesses to all bits except the write enable bit itself. This will prevent registers from accidentally be overwritten. The cfglock signal will override the port's individual enable/disable bit. If the router is connected to an AMBA bus and has a AHB slave interface all the registers can be accesses through the AMBA bus. This is more efficient than using RMAP commands for the processor.

2.7 Which ports can configure the router?

Normally all ports can access the configuration port and thus configure the router. There is an external signal called cfglock which disables accesses to the configuration port from all ports except port 1. Configuration accesses can also be disabled from ports individually through registers in the configuration port. Writes to the configuration port can be disabled for all ports by writing a bit in a configuration port register. This disables write accesses to all bits except the write enable bit itself. This will prevent registers from accidentally be overwritten. The cfglock signal will override the port's individual enable/disable bit. If the router is connected to an AMBA bus and has a AHB slave interface all the registers can be accesses through the AMBA bus. This is more efficient than using RMAP commands for the processor.

	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7	Port 8
Cfglock=0, WriteEn=1	CfgEnable[1]	CfgEnable[2]	CfgEnable[3]	CfgEnable[4]	CfgEnable[5]	CfgEnable[6]	CfgEnable[7]	CfgEnable[8]
Cfglock=1, WriteEn=1	CfgEnable[1]	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cfglock=1, WriteEn=0	CfgEnable[1] => read-only	No Access	No Access	No Access	No Access	No Access	No Access	No Access
Cfglock=0, WriteEn=0	CfgEnable[1] => read-only	CfgEnable[2] => read-only	CfgEnable[3] => read-only	CfgEnable[4] => read-only	CfgEnable[5] => read-only	CfgEnable[6] => read-only	CfgEnable[7] => read-only	CfgEnable[8] => read-only

2.8 How are transmit clocks established for both initialization and regular operation?

There is a single transmitter clock for each router device which determines the maximum clock frequency. One global register divisor register is used for setting the initalization frequency for all links (typically 10 Mbit/s). Each port has an individual divisor register which sets the frequency in regular operation. These registers are accesses through the configuration port 0 as described above.

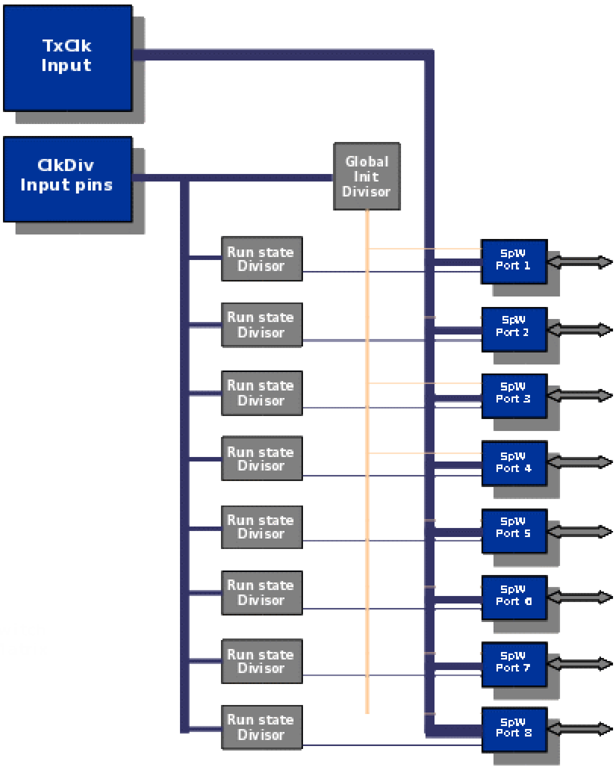

Figure 3: Transmit clock frequency division

2.9 How is the clock recovery done?

It depends on the configuration. There are two sampling modes SDR and DDR and in both cases a single clock is used for all links to sample the inputs. The standard xor gate self clocked recovery is also supported and in that case the receive clock is recovered on-chip with one xor gate per SpW link (and thus one rx clock domain per link). Finally, the UT200SpWPHY is supported in which case it recovers the clock off-chip and one clock per link has to be provided to the router FPGA/ASIC.

2.10 How does the message timeout work?

There is global prescaler in the router which is configured to generate a suitable tick rate. The number of ticks before a timeout is set with a timer register individually for each port. This determines the timeout period. If a packet is received on a port and is routed to another port the timer is started in certain cases. For example if the destination port is free the tranmission starts and the timeout is enabled. If no character is transmitted for the timeout period the packet is dropped. The timer is reset each time a character is transmitted. The timer is not started if the destination port is busy with another transmission but it is started if the destination port's link is not running. The purpose of this is to prevent that a packet blocks a route in router because a destination is blocked.

See the router user's manual for a description of all timeout cases. Two examples are shown in the figures below.

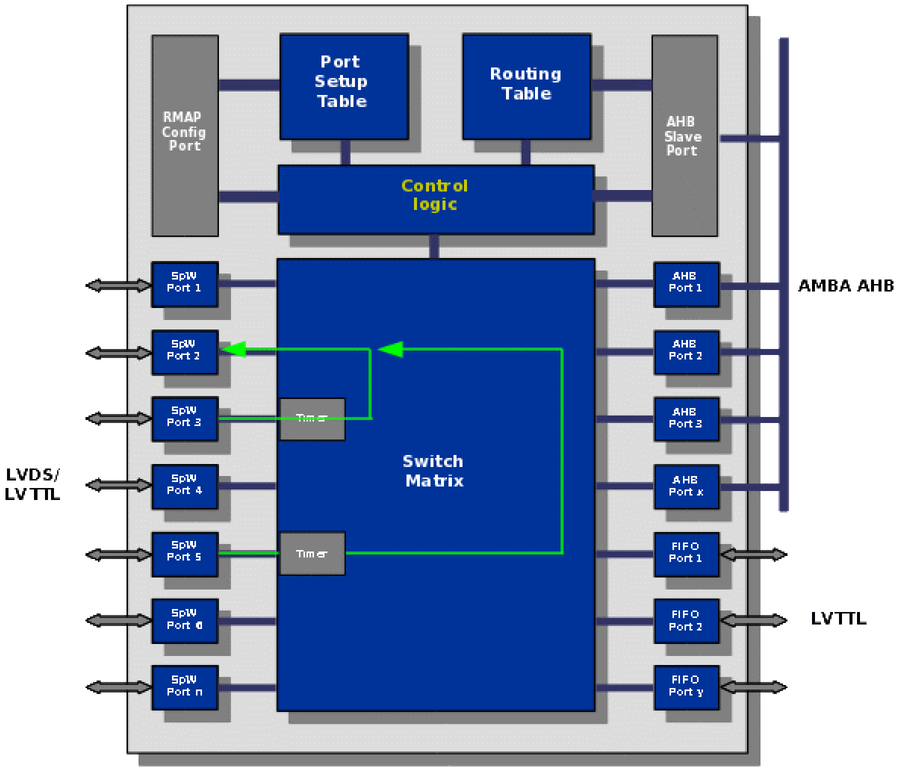

Figure 4: Times enabled when two ports are contending for access. No timeout occurs.

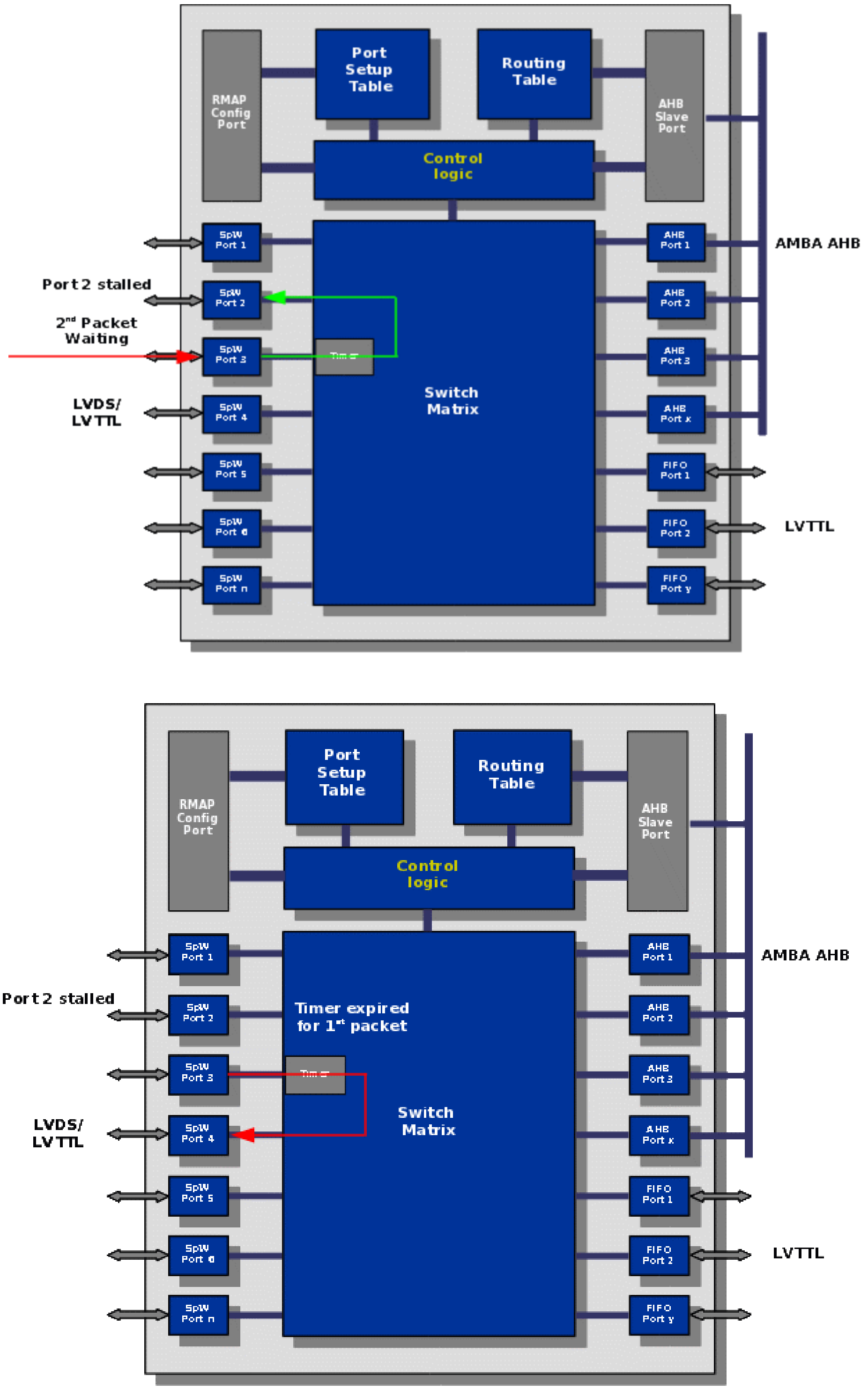

Figure 5: Stalled destination. Timeout occurs and first packet is dropped.

2.11 How is RMAP configuration and register access done?

No register setup is needed for the router to operate nominally after reset provided that the reset values for configuration registers have been properly setup. To tweak the configuration one basically writes (and reads) the registers using an RMAP command. For example to change the run state frequency for port 1 register address 0x804 is read using an RMAP read command. Bits 31 downto 24 is modified to the desired divisor value and the new value is written back to 0x804 using an RMAP write command. These accesses can nominally be done from any port. This can be done in one operation using a RMW command where the mask is set so that only bits 31 downto 24 are modified. Commonly only the routing table and link divisors are modified when the router is up and running.

2.12 Has any power analysis been done for the Router 10x configuration for RTAX2000?

Yes, For the 10x configuration a maximum value of 2500 mW has been estimated using SmartPower and an Actel defined switching rate for a 200 Mbit/s routing using sampling. This case overestimates the I/O power since a lot of the pins are static and only used for power-on configuration.

A simulated case with a maximum throughput waveform gives approximately 1900 mW which is probably closer to the real value. Power has been measured on AX2000 components which indicate that the simulated values are pretty close to real values.

2.13 How is switchover done between the primary and redundant port in the GRSPW?

The receiver logic is duplicated and the codec contains two sets of lvds rx/tx signals. In one mode forced port selection is used and the active port is selected using a register bit (the force or non-forced mode is also selected using a register bit). So if the port select bit is set to 0 the codec will try to start the link on the nominal lvds pins and if set to 1 the redundant port is used. In non-forced mode the link will stay at the currently active link as long as it is running. If it is disconnected and there is activity on the other link it will try to start the link on that port if not succesful on that one it will switch to the original one again and retry. As soon as a running state has been established on one of the ports it will not switch again until disconnected.

Information furnished by Cobham Gaisler is believed to be accurate and reliable. However, no responsibility is assumed by Cobham Gaisler for its use, or for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Cobham Gaisler.

All information is provided as is. There is no warranty that it is correct or suitable for any purpose, neither implicit nor explicit.