LeventOzturk.com

SpaceWire
(Links, nodes, routers and networks)
Verilog / VHDL design for FPGA / ASIC

Levent Ozturk

leventozturk.com
(323) 988-0345

Version 1.0
31st May, 2003

Table of Contents

leventozturk.com
(323) 988-0345

List of Figures

leventozturk.com
(323) 988-0345

1 Introduction

A SpaceWire network is made up of a number of links, nodes and routing switches.

The nodes are the sources and destinations of packets.

SpaceWire links provide the means for passing packets from one node to another. SpaceWire is a full-duplex, bidirectional, serial, point-to-point data link between nodes and routing switches. Every link encodes data using two differential signal pairs in each direction. That is a total of eight signal wires, four in each direction for each link. Electrical standard for links is defined as LVDS.

SpaceWire packets are variable length data with header including the destination address.

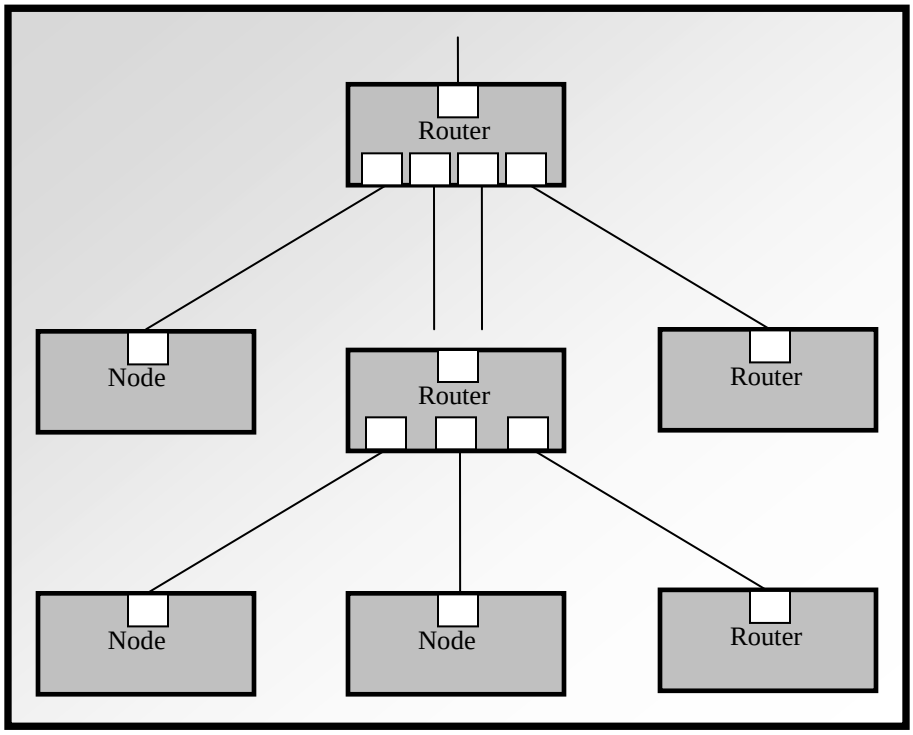


Figure 1 Sample SpaceWire Network

1.1 Features

The SpaceWire implementation;
Supports routing switches with up to 32 links.
Instantiates Altera or Xilinx (parameterized) libraries and can easily be converted other FPGA vendors or ASIC RTL.
Instantiates LVDS drivers and receivers.
Runs at frequency up to xxx MHz.
Uses xxx LE in Altera xxx device for node and xxx for xxx link routing switch.
Supports flow control, wormhole routing, header deletion, virtual channels in network level
Supports path addressing, logical addressing and regional and logical addressing, interval labeling, group adaptive routing for packets

Supports broadcast and multicast packets
Supports application level, exchange level and network level error handling and link error recovery
Disconnect error, Parity error, Escape sequence error, Character sequence error, Credit error.
A local node can be connected to a router to achieve router-node functionality without using external LVDS links.

1.2 File Structure

2 Functional Description

The SCPS network has two entities. These are Switching routers and nodes. This section describes the functional description of these entities.

Nodes provide an interface to transmit and receive data from the SCPS network. There are credit management and error recovery functions implemented in the nodes.

Routing switches routs the received data to the destination ports. They maintain a routing table to map the destination addresses to the physical ports.

leventozturk.com
(323) 988-0345

3 I/O Signal Descriptions

Table 1 SpaceWire Node I/O Signal Descriptions

Name	Direction	Description
reset	I	Global reset
clk	I	Global clock
tx_serial_clk	I	Fast Serial clock to be used in SERDES blocks
link_disabled	I	Configuration
Link_start	I	Configuration
Link_auto_start	I	Configuration
Time_master	I	Configures the node as Time master which sends ticks out
Time_tick_period[31:0]	I	Time tick period. Used when configured as master
Time_out[7:0]	O	Time value received within tick message
Tick_out	O	Tick Pulse
Tx_ready	O	Node is ready to transmit a character
Tx_write	I	A character is ready to transmit
Tx_ctl_flag	I	Control Flag to be transmitted along with data
Tx_data[7:0]	I	Transmit Data
Rx_data_buffer_write	O	Received character is ready.
Rx_data_buffer_ready	I	User is ready to receive a new character
Rx_ctl_flag	O	Received Control Flag
Rx_data[7:0]	O	Received data
Rx_clk	O	Receive clock. Extracted from din, sin pair.
Dout	O	Serial data output
Sout	O	Serial strobe output
Din	I	Serial data input
Sin	I	Serial strobe input

leventozturk.com
(323) 988-0345

4 Detailed Design

This section describes the implementation details of tee routing switch, node and sub modules.

4.1 Routing Switch

The routing switch contains routing tables and nodes. The routing switch can be configured to instantiate up to 32 links. It maintains a routing table which is configured by the Software.

The routing switch data ports are single ended. The LVDS interface is instantiated outside the routing switch module to so that the routing switch is independent of the hardware.

The routing switch can be configured as a switch or as a node.

Software can configure the routing switch as Tick master. In this case an internally generated Tick will be transmitted through all the ports. If the routing switch is not configured as Tick master, any received tick message will be distributed to all the other ports.

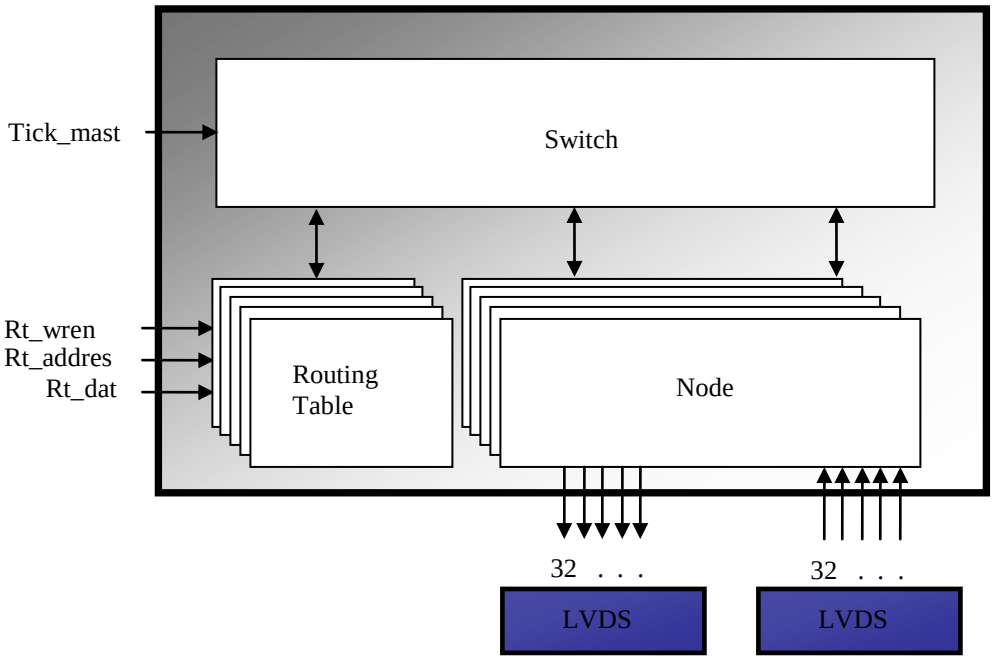
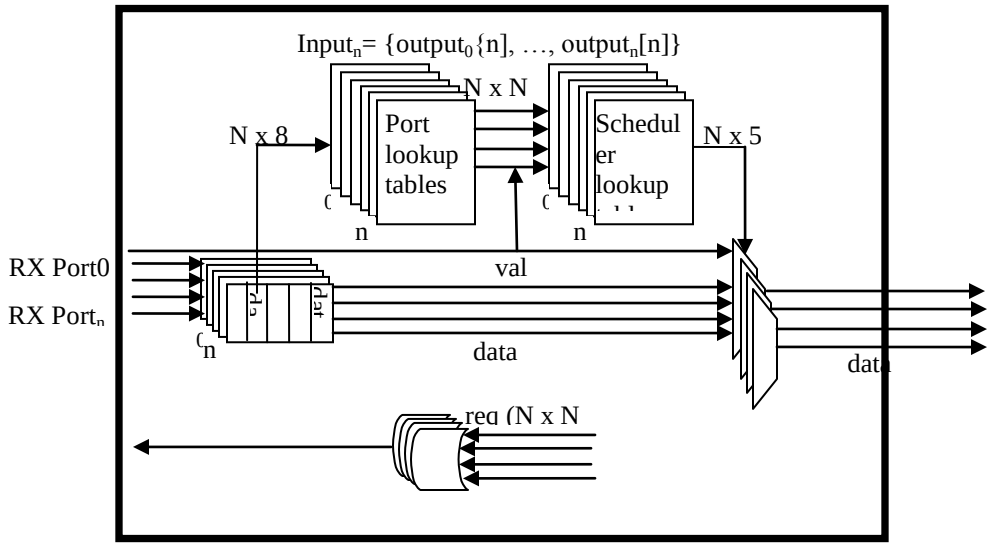


Figure 2 SpaceWire Routing Switch Block Diagram

The routing switch is implemented with no queuing, no switching delay, and distributed arbitration with priority scheduling.
There is no queuing algorithm. Each source port can be directly selected by each destination port.
The switching logic runs at low speed parallel data side clock domain with no switching delay. Short packets will not degrade the bandwidth. Header processing is achieved on the fly.
There is one scheduling algorithm for each destination port. Each destination port will decide which source port to schedule. If multiple source ports are waiting for a single destination port, the lowest numbered source port will be scheduled. If multiple destination port is enabled for a source port(load distribution), the lowest destination port will reserve the source port. If the destination port is busy, the next destination port that is enabled for a source port will reserve the source port.

leventozturk.com
(323) 988-0345



Routing table is duplicated for each link and structured the same. Write address

4.2 Node

This module contains the link interfaces and serialisers. The number of link interfaces is parameterized and configurable.

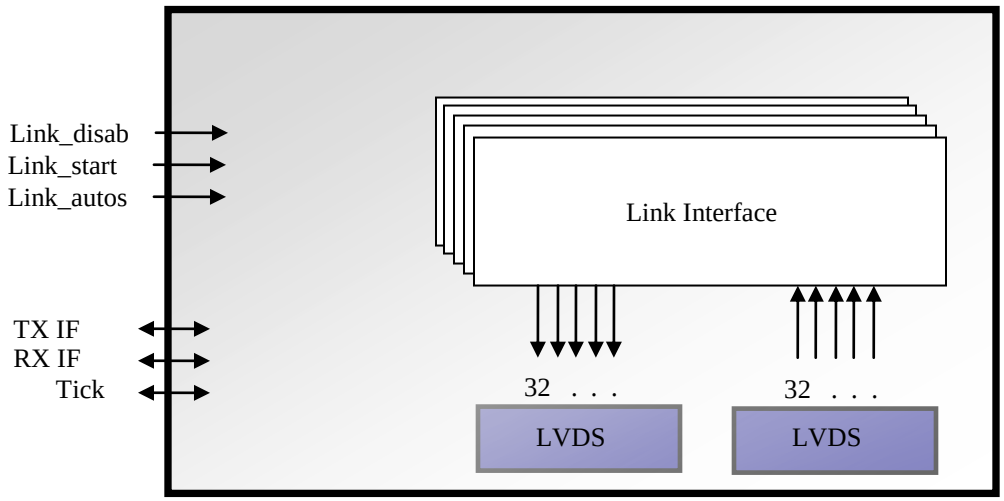


Figure 14 SpaceWire Node Block Diagram

Table 2 SpaceWire Node I/O Signal Descriptions

Name	Direction	Description
reset	I	Global reset
clk	I	Global clock
tx_serial_clk	I	Fast Serial clock to be used in SERDES blocks
link_disabled	I	Configuration
Link_start	I	Configuration
Link_auto_start	I	Configuration
Time_master	I	Configures the node as Time master which sends ticks out
Time_tick_period[31:0]	I	Time tick period. Used when configured as master
Time_out[7:0]	O	Time value received within tick message
Tick_out	O	Tick Pulse
Tx_ready	O	Node is ready to transmit a character
Tx_write	I	A character is ready to transmit
Tx_ctl_flag	I	Control Flag to be transmitted along with data
Tx_data[7:0]	I	Transmit Data
Rx_data_buffer_write	O	Received character is ready.
Rx_data_buffer_ready	I	User is ready to receive a new character
Rx_ctl_flag	O	Received Control Flag
Rx_data[7:0]	O	Received data
Rx_clk	O	Receive clock. Extracted from din, sin pair.
Dout	O	Serial data output
Sout	O	Serial strobe output
Din	I	Serial data input
Sin	I	Serial strobe input

4.3 Link Interface

Link Interface module implements the full featured link layer complying SpaceWire Link Interface. The output is not serialized to allow serialization to be done by an external serializer. This also allows connecting a local node to a routing switch internally.

The Link FSM module controls the link operation. The state machine described in Chapter 8.5 Figure 20 and Annex B of SpaceWire Standard is fully implemented.

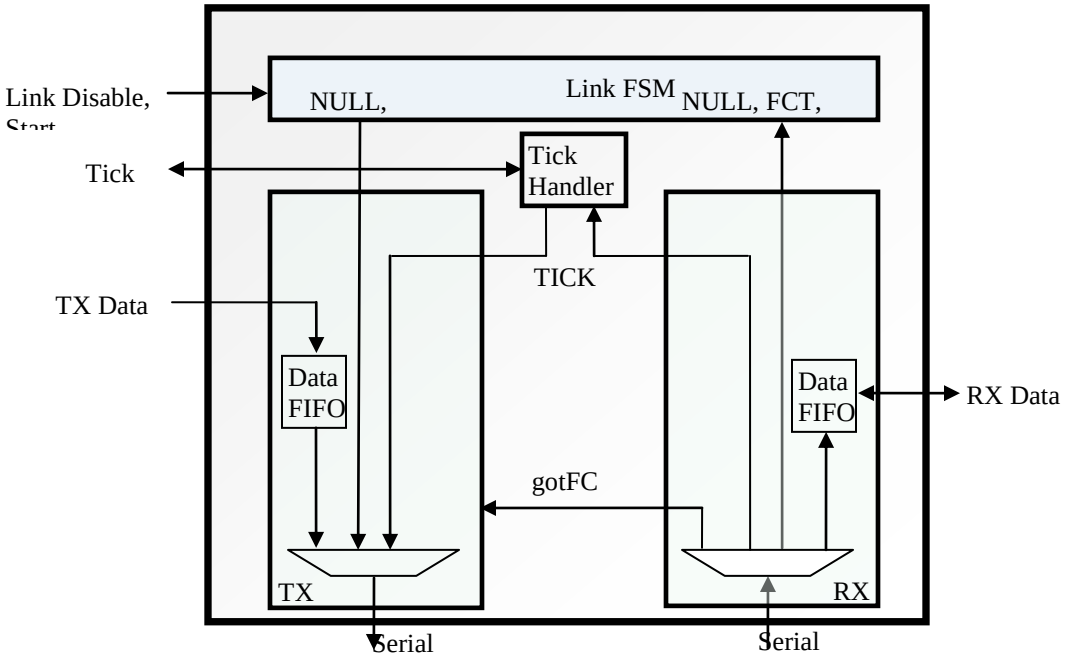


Figure 15 SpaceWire Link Interface Block Diagram

Table 3 SpaceWire Link Interface I/O Signal Descriptions

Name	Direction	Description
reset	I	Global reset
clk	I	Global clock
tx_serial_clk	I	Fast Serial clock to be used in SERDES blocks
link_disabled	I	Configuration
link_start	I	Configuration
link_auto_start	I	Configuration
time_master	I	Configures the node as Time master which sends ticks out
time_tick_period[31:0]	I	Time tick period. Used when configured as master
time_out[7:0]	O	Time value received within tick message
time_tick_out	O	Tick Pulse
tx_ready	O	Node is ready to transmit a character
tx_write	I	A character is ready to transmit
tx_ctl_flag	I	Control Flag to be transmitted along with data
tx_data[7:0]	I	Transmit Data
rx_data_buffer_write	O	Received character is ready.
rx_data_buffer_ready	I	User is ready to receive a new character
rx_ctl_flag	O	Received Control Flag

rx_data[7:0]	O	Received data
rx_clk	O	Receive clock. Extracted from din, sin pair.
dout[7:0]	O	Serial data output
sout[7:0]	O	Serial strobe output
din	I	Serial data input
sin	I	Serial strobe input

4.3.1 Credit Management

Credit management is achieved by using two counters in both TX and RX for each direction.

On the TX side, the credit counter is incremented by 8 for each FCT that is received from RX and decremented by 1 for each data transmitted. If credit count is less than 56 and FCT is received, a credit error is generated.

The RX credit count is implemented in the Link FSM module. The RX credit counter is decremented by 8 for each FCT that is transmitted from TX and incremented by 1 for each data received. The Link FSM sends FCTs to keep the credit count at 56. If the RX credit count is 0 and new data is received, a credit error is generated.

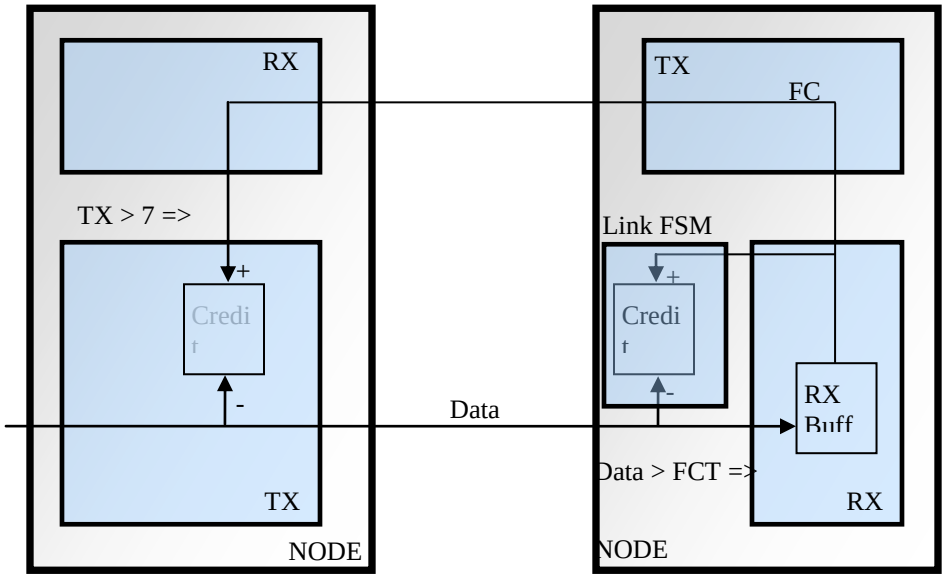


Figure 16 SpaceWire Credit Management

4.3.2 Error Handling and Recovery

The following errors are fully implemented.
Link level errors
RxErr
Disconnect Error
Parity Error
Escape Error
Credit Error

Character Sequence Error
Network level errors
Link error
EEP received
Invalid destination address

4.4 Resets

The SpaceWire routing switch module uses an active high reset.

4.5 Interrupts

5 Resource Utilization

SpaceWire Node and routing switch I implemented in several FPGA and ASIC devices. The following tables provide typical utilization figures using standard synthesis tool for different SpaceWire configurations.

Table 4SpaceWire Node with 1 link resource utilization

Device	Logic Cells			Memory		DSP	PLL	Utilization
	Combinatorial	Sequential	Total	Bits	Blocks			%
StratixII EP2S15	563	470	470	1344		0	1	5

5.1 Clocks

SpaceWire clock rate can be programmed to be50 to 150 MHz. All the families listed above meet the required performance.

5.2 LE

6 AC Characteristics

leventozturk.com
(323) 988-0345

7 Software Interface

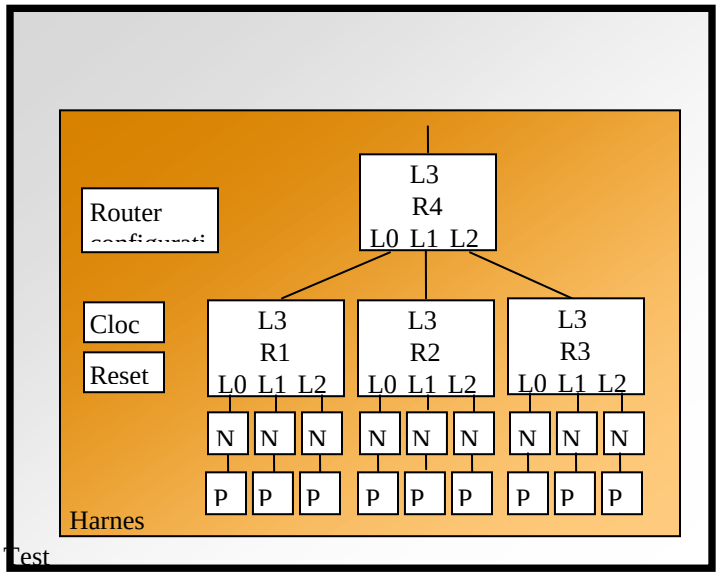
7.1 Generic Register Interface

The SpaceWire does not contain any configuration, control or status registers.

7.2 NiosII Avalon Interface

leventozturk.com
(323) 988-0345

8 Simulation



8.1 Test Bench

8.2 Harness

P1-9: Packet Managers
N1-9: Nodes
R1-4: Routers
The Packet managers are configured to loop back the packets replacing the destination address except N1 which generates and receives the packets.

P1->N1->N7->N8->N4->N2->N5->N9->N3->N6->N1->P1

At startup routers are configured by the harness module as in Table1

Table 5

Destination Address	Node
0x29	1
0x2A	2
0x2B	3
0x 81	4
0x 82	5
0x 83	6
0x A4	7
0x A3	8
0x A2	9

8.3 Test Cases

8.4 Behavioral Models

8.5 Simulator Settings

leventozturk.com
(323) 988-0345

9 Lab Verification

9.1 Lab settings

9.2 Test Cases

leventozturk.com
(323) 988-0345

10 References

SpaceWire – Links, nodes, routers and networks, Draft ECSS-E-50-12A, 6 November 2002
Altera Data Sheet
Xilinx Data Sheet

leventozturk.com
(323) 988-0345