SpaceWire Router

Parkes, Steve; McClements, Chris; Kempf, Gerald; Fischer, Stephan; Leon, Agustin

Publication date:
2003

Document Version
Publisher's PDF, also known as Version of record

SpaceWire Router

International SpaceWire Seminar (ISWS 2003)

4-5 November 2003, ESTEC Noordwijk, The Netherlands

Steve Parkes [(1)], Chris McClements [(1)], Gerald Kempf [(2)], Stephan Fischer [(3)], Agustin Leon [(4)]

[(1)] University of Dundee

Applied Computing, Dundee, DD1 4HN, Scotland, UK

Email:sparkes@computing.dundee.ac.uk, cmcclements@computing.dundee.ac.uk

[(2)] Austrian Aerospace

Complete mailing address (including country)

Email:

[(2)] EADS Astrium GmbH

Complete mailing address (including country)

Email:

[(2)] European Space Agency

ESTEC, Noordwijk, The Netherlands

Email: agustin.leon@esa.int

ABSTRACT

A SpaceWire network comprises SpaceWire links, nodes and routers. The nodes are the functional units that wish to use the onboard communication services of the SpaceWire network and are fitted with one or more SpaceWire interfaces. These units are connected together directly using point-to-point SpaceWire links or indirectly via SpaceWire routers. SpaceWire interfaces, links and routers are the three elements of a SpaceWire network. This paper explains the operation of a SpaceWire router and describes the radiation tolerant SpaceWire router ASIC being developed for ESA.

A SpaceWire routing switch is able to connect together many nodes, providing a means of routing packets between the nodes connected to it. A SpaceWire routing switch comprises a number of SpaceWire link interfaces and a routing matrix. The routing matrix enables packets arriving at one link interface to be transferred to and sent out of another link interface on the routing switch

The radiation tolerant routing switch currently being developed is fully SpaceWire compliant and has the following facilities:

- Eight SpaceWire ports.
- Two external parallel ports, each comprising an input FIFO and an output FIFO.
- A non-blocking crossbar switch connecting any input port to any output port.
- An internal configuration port accessible via the crossbar switch from the external parallel port or the SpaceWire ports.
- A routing table accessible via the configuration port which holds the logical address to output port mapping.

- Control logic to control the operation of the switch, performing arbitration and group adaptive routing.
- Control registers than can be written and read by the configuration port and which hold control information e.g. link operating speed.
- An external time-code interface comprising tick_in, tick_out and current tick count value.
- Internal status/error registers accessible via the configuration port
- External status/error signals

This paper describes the operation of a SpaceWire Router covering path and logical addressing schemes, priority and arbitration mechanisms, and group adaptive routing. The architecture of the router ASIC is described and the results of initial FPGA implementation and testing reported. The development schedule and expected performance are provided.

SPACEWIRE ROUTERS

In this section the basic operating principals of a SpaceWire router are described.

SpaceWire Networks

A SpaceWire network is made up of a number of SpaceWire nodes interconnected by SpaceWire routers. SpaceWire nodes are the sources and destinations of packets and provide the interface to the application system(s) using the communication services provided by the SpaceWire network. SpaceWire nodes may be directly connected together using SpaceWire links or they may be interconnected via SpaceWire routers using SpaceWire links to make the connection between node and router.

A typical SpaceWire network is illustrated in Fig. 1. The processors (P1 to P4) in the processor array are directly connected to one another. The sensors, memories and processor array are interconnected via the routers. Redundancy is provided in this example network by the use of redundant links and a pair of routers. If data is being sent from Sensor 1 to Memory 1 via Router 1 and the link between the sensor and the router fails then data can be sent from Sensor 1 to Memory 1 via Router 2.

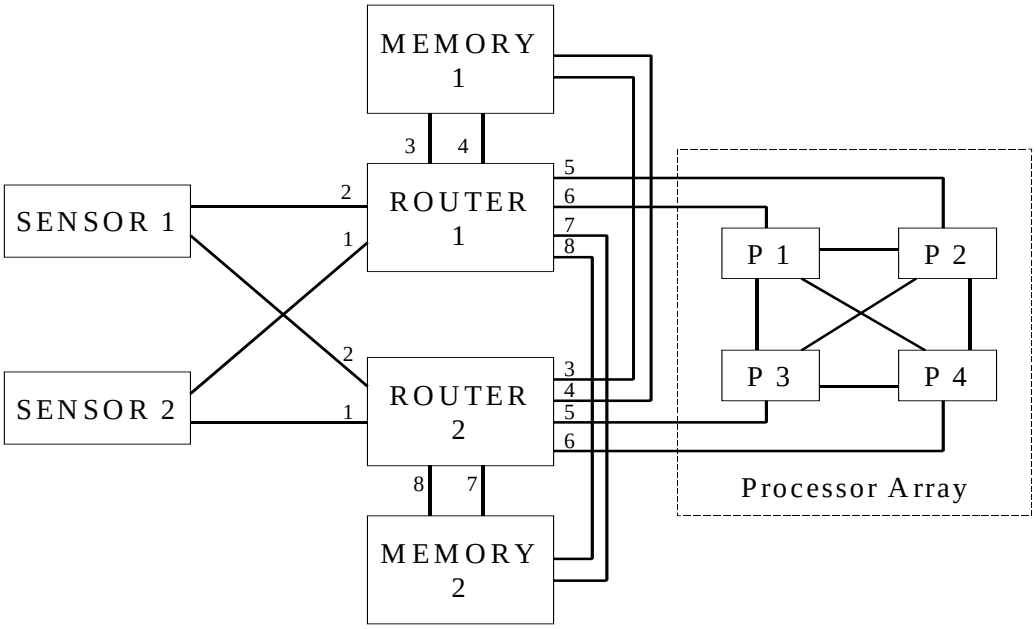

Fig. 1 Example SpaceWire Network

SpaceWire Packets

Information is transferred across a SpaceWire network in distinct packets. The format of a packet is

<Destination Address><Cargo><End_of_Packet>

The “Destination Address” is the first part of the packet to be sent and is a list of zero or more data characters that defines the node on the network for which the packet is intended. This list of data characters represents either the identity code of the destination node or the path that the packet will take to get to the destination node. The “Cargo” is the data to be transferred from source to destination. The “End_of_Packet” is used to indicate the end of a packet. The data character following an End_of_Packet is the start of the next packet.

Routers

A SpaceWire router connects together many nodes and provides a means of routing packets from one node to one of many other possible nodes. A SpaceWire router comprises a number of SpaceWire link interfaces and a switch matrix. The switch matrix enables packets arriving at one link interface to be transferred to and sent out of another link interface on the router. Each link interface may be considered as comprising an input port (the link interface receiver) and an output port (the link interface transmitter) as illustrated in Figure 3.

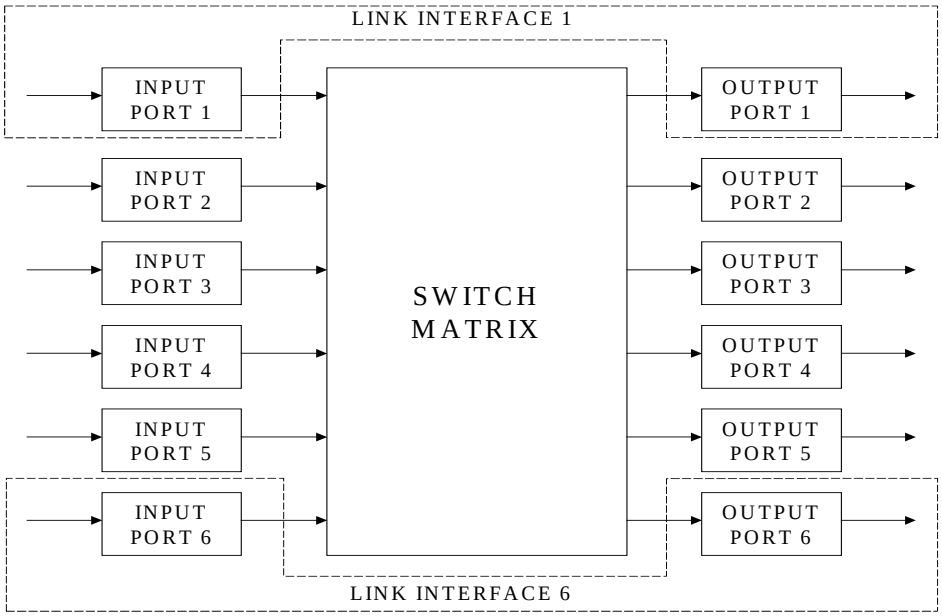

Fig. 2 Router Switch Matrix

A SpaceWire router transfers packets from the input port of the switch where the packet arrives, to a particular output port determined by the packet destination address. A router uses the leading data character of a packet (one of the destination identifier characters) to determine the output port of the router to which the packet is to be routed. If there are two input ports waiting to use a particular output port when the previous packet has finished being sent then an arbitration mechanism in the output port decides which input port is to be served.

Addressing Packets

There are two ways of addressing SpaceWire packets: path addressing or logical addressing.

Path addressing is used to specify the path through a network directly. The leading data character of a packet (the destination identifier) contains the required output port number of the router. For example, to route a packet from Sensor 1 to Memory 1 in Figure 1 a packet is sent to Router 1 with its leading data character (destination identifier) set to 3 (or 4). When this packet arrives at Router 1 it is forwarded out of port 3 (or 4). For path addressing the leading data character is removed after a router has used it to determine the output port for a packet. If a packet has to pass through several routers to reach its destination then a separate data character is used to specify the output port for each router. Hence if there are three routers to be traversed, three data characters will be needed to specify the destination.

Logical Addressing is used to specify the path through a network indirectly via routing tables held in the routers. The leading data character of a packet holds the logical address, which is used to look up the required output port number in the routing table of the router. For example, to route a packet with logical address 73 from Sensor 1 to Memory 1 in Figure 1, the routing table entry in Router 1 for logical address 73 must be set to either 3 or 4. With logical addressing

the leading data character is not normally deleted by the router since the logical address identifies the destination node and will be used by each router encountered on the path to the required destination.

Routing Tables

A routing table within the router holds the logical-physical mapping. The router addresses are assigned as shown in Table 1. The internal configuration port is used to access the routing table and other configuration information held in a router.

Table 1 Router Addresses

Address Range	Function
0	Internal Configuration Port
1-31 (01-1F hex)	Physical Output Ports
32-254 (20-FE hex)	Logical Addresses, which are mapped on to the physical output ports.
255 (FF hex)	Reserved

An example routing table is illustrated in Fig. 3. A '1' in the table maps an address to an output port number. The configuration port (port 0) is accessed only via path addressing with the address 0. Ports 1 to 4 are accessed using path addresses 1 to 4 respectively. Path addresses beyond address 4 have no meaning in a 4-port router and give rise to a routing error. Logical addresses can be used to access any of the four output ports depending on how the routing table is programmed. For example in Fig. 3, logical address 33 has been programmed to port 4. Any packets arriving with address 33 will be routed to output port 4.

	Address	Port 0	Port 1	Port 2	Port 3	Port 4
Configuration	0	1	0	0	0	0
Hardware Addressing	1	0	1	0	0	0
	2	0	0	1	0	0
	…					
Logical Addressing	32	0	0	1	0	0
	33	0	0	0	0	1
	34	0	1	0	0	0
	…					
	255	0	0	0	0	0

Fig. 3 Routing Table for 4-Port Router

Bandwidth sharing and Fault tolerance

Two of the key features of SpaceWire networks are bandwidth sharing and fault tolerance. This section describes how group adaptive routing implemented in SpaceWire routers provides these important capabilities.

Group Adaptive Routing

SpaceWire routers can implement group adaptive routing. When two or more output ports lead to the same destination (another router or a node) then these output ports may be configured as a group. When a packet arrives at an input port

for routing out of an output port that is busy, any other output port in the same group as the addressed output port may be used to forward the packet.

A routing table in a router that implements group adaptive routing is illustrated in Fig. 4. Logical address 33 has two output ports assigned to it. When a packet with this logical address is received the router has the choice of routing the packet out of port 3 or port 4.

	Address	Port 0	Port 1	Port 2	Port 3	Port 4
Configuration	0	1	0	0	0	0
Hardware Addressing	1	0	1	0	0	0
	2	0	0	1	0	0
	…					
Logical Addressing	32	0	1	1	0	0
	33	0	0	0	1	1
	34	0	1	1	0	0
	…					
	255	0	0	0	0	0

Fig. 4 Routing Table with Group Adaptive Routing

Bandwidth Sharing

Group adaptive routing gives rise to one of the most important features of a SpaceWire network – bandwidth sharing. Links in a group share the total data flow between them. If there are two equivalent links between a pair of routers (and the routing tables are configured appropriately) then data may pass over either of these links. There is twice the bandwidth of a single link available for transferring data between the two routers. If another equivalent link is added then the bandwidth become three times that of a single link. To use this extra bandwidth all that needs to be done is to configure the routing tables to identify the three links as belonging to a group.

Fault Tolerance

Bandwidth sharing leads to another important feature of a SpaceWire network – fault tolerance. Consider the previous example of three equivalent links organised into a group. If one of these links fails then the information flow will be automatically distributed over the other two equivalent links. There is no need for intervention by network management software to do this. Provided the routing tables identify the three links as belonging to a group, the fault recovery is automatic and immediate. The only information lost is the tail end of the packet that was being transferred when the fault occurred.

Priority Packet Delivery

As stated earlier, if there are two input ports in a router waiting to use a particular output port, then an arbitration mechanism is used to select which input port is to be served. The arbitration mechanism can include a priority scheme. There is no priority flag available within the header of a SpaceWire packet to specify its priority level. The SpaceWire header only contains address information, so packet priority must be associated with a logical address (or with the input port number). In the routing tables logical addresses may be assigned high or low priority. High priority logical addresses have preferential access to an output port when arbitration takes place. A logical address that has been assigned high priority, acts as a high priority channel across the network from many possible sources to the one destination. If high and low priority access to a particular destination is required then two logical addresses are required for a particular destination, one assigned high priority and the other low priority. A source can then decide which logical address to use when sending a packet to a destination, depending on the required priority of the packet. There is a compromise between the number of destinations that can be addressed and the number of priority levels. With two

priority levels it is possible to have, say 128 low priority destinations and 96 high priority destinations within the 224 logical addresses available. Priority is a means of providing quality of service control.

It is possible to ensure real-time, deterministic delivery of commands (packets) using priority addresses provided that there are no possible clashes between routes through the network operating priority addressing. This can be achieved, for example, if there is only one node sending out priority commands. If deterministic delivery is required there must also be a limit to the maximum packet size used on the network.

SPACEWIRE ROUTER ASIC

In this section the architecture of the SpaceWire router ASIC currently under development is described and the development schedule outlined.

Router Architecture

The architecture of the SpaceWire router ASIC is illustrated in Fig.5.

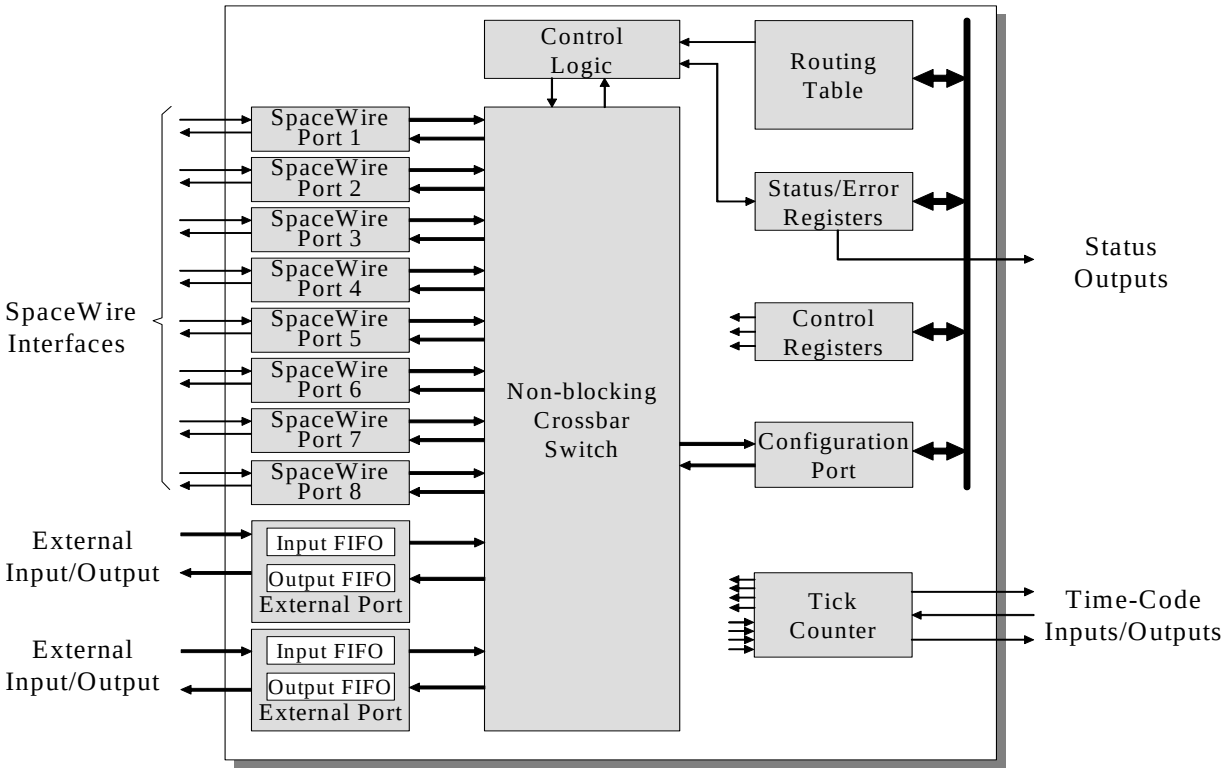

Fig. 5 SpaceWire Router ASIC Architecture

There are eight SpaceWire ports, two external ports and an internal configuration port in the SpaceWire router. A low latency, worm-hole routing, non-blocking, crossbar switch enables packets arriving at any SpaceWire port, external port or generated in the configuration port to be directed out of any other SpaceWire or external port or to be routed to the configuration port.

The SpaceWire ports are fully compliant with the SpaceWire standard [3] providing high-speed, bi-directional communications. The external ports each comprise an input FIFO and an output FIFO and can receive and send data characters and end of packet markers. A time-code port is also provided along with a time-counter to facilitate the propagation of time-codes [6]. When a valid time-code arrives at a router port it is sent out of all the other SpaceWire ports and a TICK_OUT signal is generated at the time-code port. The router can operate as a time-code master using the TICK_IN provided in the time-code port.

The configuration port is accessible via any of the SpaceWire or external ports. It contains registers which control the operation of the SpaceWire ports, external ports and the crossbar switch. The configuration port holds status registers

for the various ports and the switch. These registers can be read using a configuration read command to determine the status of the router and to access error information. Status and error information can also be selected for output on several status pins. The routing table is accessed via the configuration port. The logical address port mappings and priority bits can be set in the routing table. The routing table is used to control group adaptive routing and priority arbitration in the crossbar switch.

Development schedule

The SpaceWire Router project started in January 2002 with the definition of the router requirements and the basic design of the router and of the SpaceWire CODEC. By the end of 2003 a FPGA version of the router will have been designed. Furthermore, the testbed, the test metrics and the tools necessary for the validation of the FPGA will also have been developed in 2003. For the testing of the router, SpaceWire compliant nodes are essential; therefore a PCI card with a FPGA version of a SpaceWire compliant SMCS is to be used. Furthermore, a break-out box has been developed as a monitoring tool. With this equipment and the respective software the validation exercise will take place from January to April 2004. The results will be considered for the final design of the Router ASIC which will be delivered in December 2004. The SpaceWire Router project will be finished after the Router ASIC validation in March 2005.

Initial FPGA implementation

The SpaceWire router has been implemented in a Xilinx XCV400E Virtex-E device and has eight SpaceWire ports and one external port. The SpaceWire ports operate at a maximum baud rate of 200 Mbits/s. This device is fully compliant with the SpaceWire standard [3] and is functionally representative of the SpaceWire router ASIC, with the exception that the FPGA implements one external port while the ASIC will have two. The SpaceWire router design is currently being implemented with two external ports in a Xilinx Virtex-2 FPGA.

SpaceWire Router ASIC performance

The SpaceWire Router (SPROUT) ASIC will be implemented in an Atmel MH1RT gate array with maximum gate count of 519 kGates (typical). This technology uses a 0.35 μm CMOS process with a radiation tolerance of up to 300k rad and SEU free cells up to 100 MeV (used for all critical memory cells) as well as latch-up immunity up to 100 MeV. The maximum baud-rate of the SpaceWire link interfaces of the SPROUT will be 200Mbit/s, where LVDS I/Os integrated onto the chip are used. Its estimated power consumption at the maximum data rate will be about 4 W and it will operate from a single supply voltage of 3.3V (±0.3V). The package is a 196 pin ceramic Metric Quad Flat Package with 25 mil pin spacing.

CONCLUSIONS

An overview of SpaceWire router operation has been given and the architecture of the SpaceWire router ASIC currently under development has been described. An FPGA version of the SpaceWire router ASIC fully compliant with the SpaceWire standard has been developed and is currently being used to validate the router architecture. The SpaceWire router ASIC will be available at the beginning of 2005 and is expected to operate at a baud-rate of 200 Mbits/s.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of ESA for the High-Speed Interconnect and SpaceWire Router contracts under which the SpaceWire router ASIC is being developed. Specific thanks are due to Pierre Fabry and Agustin Leon the ESA study technical managers for these contracts and to Philippe Armbruster for overall leadership in this area.

REFERENCES

[1] S.M. Parkes and J. Rosello, “SpaceWire – Links, Nodes, Routers and Networks,” *DASIA 2001 DAta Systems In Aerospace (28th May - 1 June, Nice, France 2001) ESA Publication Number **SP-483**, ISSN 1609 042X, ISBN 92-9092-773-9.*

[2] J. Rosello-Guasch, “SpaceWire Homepage,” http://www.estec.esa.nl/tech/spacewire/

[3] S.M. Parkes et al, “SpaceWire: Links, Nodes, Routers and Networks,” *European Cooperation for Space Standardization, Standard No. ECSS-E50-12A,Issue 1, January 2003.*

[4] IEEE Computer Society, “IEEE Standard for Heterogeneous Interconnect (HIC) (Low-Cost, Low-Latency Scalable Serial Interconnect for Parallel System Construction)”, *IEEE Standard 1355-1995, IEEE, June 1996.*

[5] Tele-communications Industry Association, “Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits,” *Standard ANSI/TIA/EIA-644 1995, March 1996.*

[6] S.M. Parkes, “The Operation and Uses of the SpaceWire Time-Code,” *International SpaceWire Seminar, ESTEC Noordwijk, The Netherlands, November 2003.*