

XSwitch

Subjects: Computer Science, Software Engineering

Contributor: HandWiki Liu

The XSwitch is an interconnect used by the XCore processor. The interconnect protocol is defined by XMOS, and is based around routing messages comprising 9-bit tokens between cores on a network. The protocol is specifically designed for on-chip and board-level communication, but using LVDS drivers it can also run over longer cables.

Keywords: xswitch ; xcore ; xmos

1. Description

1.1. System Level Description

The interconnect routes sequences of messages. A message consists of a header that specifies the core address, a sequence of tokens, and an END-token. There are 512 tokens, 256 of which are data tokens, and 256 of which are control tokens. Data tokens are used to transport data (e.g. an audio data stream), the control tokens can be used to send in-band control data, in order to implement protocols over the interconnect. A message is routed through a sequence of switches interconnected by XLinks.

There are two control tokens that can be used to end a message: END which ends the message, and PAUSE which suspends the message. Both tokens will free the route through the switches. The difference between END and PAUSE is that the END token will be delivered to the receiver, whereas the PAUSE token will be silently thrown away by the last switch.

Since messages are terminated by a token (rather than have a packet length), the sender may choose not to terminate a message. This creates an open circuit with a guaranteed bandwidth and latency. Applications can use, for example, one series of links to form a circuit to stream audio, and use another series of links for small packets containing control data. Circuits guarantee in order delivery of all tokens transmitted. Messages may be delivered out of order if there are multiple paths. Circuits permanently occupy a series of links, whereas messages only occupy links for short periods of time.

The XSwitch interconnect architecture implements the media layer of the OSI model.

1.2. Physical Layer

At the physical layer the signal is either transmitted using a serial protocol over two wires, or using a fast protocol over five wires. In both cases the signal is transmitted as a series of transitions using a 1 out of M code.

On a two-wire system a transition on wire 0 signals a '0' bit, and a transition on wire 1 signals a '1' bit. A token is transmitted as a sequence of exactly 10 transitions, the first eight signal the eight data bits (MSB first) then one bit to signal whether this is a control token, and then finally a return-to-zero bit that returns the wires to zero (after nine transitions one wire will be high).

On a five-wire system there are four data wires (0, 1, 2, and 3). A transition on 0 signals '00', a transition on 1 signals '01', etc. A sequence of four transitions transmits eight data bits. The fifth wire is an escape wire and is used to transmit the control tokens. Unlike the two-wire system any even number of wires can be high after a token, and the wires only return to zero at the end of a message.

Both modes are to a degree asynchronous in that there is no clock; the transitions signal the data, the order of the transitions signal the order of the bits. A transmitter must not transmit data faster than the receiver can interpret the data, and it must make sure that there are sufficient gaps between edges for them not to overtake each other on pads. Data transmission can be suspended at any time.

1.3. Link Layer

At the link layer the protocol relies on the use of credits^[1] in order to signal whether data can be transmitted or not. Receivers always issue credits to the transmitter before the transmitter is allowed to send any data. A link can give 8, 32, or 64 credits, but it should never issue more than 127 credits which guarantees that the other side can always record credits using 7 bit counters. Conversely, if large numbers of credits are given out the link needs to have large receive buffers; the minimum size of a receive buffer is 8 tokens (72 bits).

1.4. Network Layer

The network layer routes messages through the interconnect. Nodes in the network are identified by means of a 16-bit address. Messages are routed by successively matching bits of the destination address. On every node the message is brought closer to the destination by taking the first non matching bit, and looking up a set of links to be used for that destination. The message is then routed over that link, and the route is "rolled up" when the END or PAUSE control token is found. Given that the header precedes a packet, this implements wormhole routing^[2]

The routing can be set up to be deadlock free.^[3] It is free of starvation if all interconnects arbitrate incoming messages without locking any link out.

2. Implementations

The switches of the XCore XS1-G4, XCore XS1-L1, and xCORE-200 processors are all implementations of the XSwitch. The implementations differ in terms of the number of internal and external links supported.

2.1. XS1-G4 Switch

The XS1-G4 switch has 32 links. 16 of the links are external links that can operate at a speed of up to 400 Mbits per second. The other 16 links are connected to each of the four cores - four links per core. Each of the internal links can operate at 3.2 Gbits per second. The maximum throughput of this implementation of the XSwitch is 57.6 Gbits per second: 6.4 Gbit per second over all external links and 51.2 Gbits per second on the internal links.

2.2. XS1-L1 Switch

The XS1-L1 switch has 12 links. Eight of the links are external links that can operate at a speed of up to 400 Mbits per second. The other four links are connected internally to the core. Each of the internal links can operate at 3.2 Gbits per second. The maximum throughput of this implementation of the XSwitch is 16 Gbits per second: 3.2 Gbit per second over all external links and 12.8 Gbits per second on the internal links.

2.3. xCORE200 Switch

The xCORE200 switch has 17 links. Eight of the input links are external links that can operate at a speed of up to 400 Mbits per second. Eight links are internal and connected to the cores; four to each core. Each of the internal links can operate at 4 Gbits per second. The final link is connected to the on-chip USB PHY. The maximum throughput of this implementation of the XSwitch is 35.25 Gbits per second: 3.2 Gbit per second over all external links and 32 Gbits per second on the internal links, plus 0.05 Gbit/s over the final internal link

References

1. William J. Dally, Brian Towles (2003). Principles and Practices of Interconnection Networks. ISBN 0-12-200751-4.
2. Universal Routing Strategies for Interconnection Networks. 1998. ISBN 978-3-540-64505-4.
3. P. W. Thompson, M.D. May, P. H. Welch (1994). Networks, Routers and Transputers: Function, Performance, and Application. p. 8. ISBN 90-5199-185-1.