Design of a Switching Node (Router) for On-Chip Networks

Sumant Sathe, Daniel Wiklund, Dake Liu
Department of Electrical Engineering
Linköping University, Sweden 581 83

Abstract

Managing the complexity of designing chips containing billions of transistors requires decoupling computation from communication. For the communication, scalable and compositional interconnects, such as on-chip networks (OCN), must be used. Our OCN is capable of providing a data transfer throughput of 19.2 Gbps/link. The key element of our OCN is the switching node. We present a prototype design of a 5-input, 5-output, scalable switching node. The switching node is constructed from a collection of parameterizable and reusable hardware blocks, and is a basic building block of our OCN. The switching node is characterized by an area of 0.06 mm sq. and a frequency of 1.2 GHz in 0.18 micron CMOS technology.

1. Introduction

On-Chip Networks provide an alternative to existing on-chip interconnects because [4-7], (a) they structure and manage global wires in new deep submicron technologies, (b) share wires, lowering their number and increasing their utilization, (c) can be energy efficient and reliable, and (d) are more scalable when compared to traditional buses.

The desired properties of On-Chip Networks are now reviewed. The OCN must provide *reliable communication*. This can be achieved by ensuring that data-dropping is not allowed. The OCN must be free from *deadlock*. This can be ensured if no resource in the OCN is allowed to be locked indefinitely while waiting for another resource. Deadlock can be avoided without dropping data by introducing constraints either in the topology or routing. The OCN must conform to *data ordering*. This is necessary to minimize buffer space, and eliminates the need for reordering modules. The *network flow control* used in the OCN can be of three types, viz. store-and-forward, virtual-cut-through and wormhole routing. Wormhole routing requires the least buffering (buffers flits instead of packets) and also allows low-latency communication. Our OCN provides a very high bandwidth on account of (a) the number of buses running in parallel between the switching nodes in the network, (b) the high throughput of data transfer of 19.2 Gbps/link in the network, and (c) the relatively low setup and buffering latencies in the network.

In this paper, the architecture and design of a 5-input, 5-output, switching node suitable for our OCN is introduced, and the impact of the number of ports on the area is investigated.

2. OCN architecture

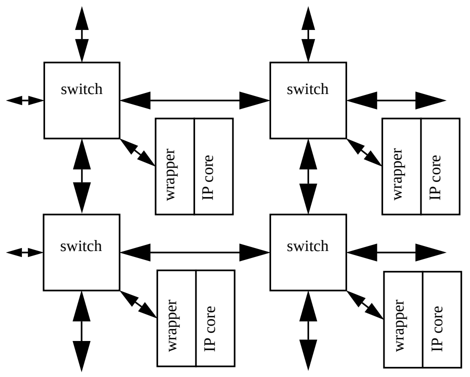

Fig. 1. A 2 x 2 OCN with switching nodes, wrappers and IP blocks

The OCN has been implemented as a 2-dimensional mesh as shown in Fig. 1 [1]. It uses 5-port switches that allocate four ports to connect to adjacent switches and one port to connect to the local IP block interface (port). The local IP port is connected through a wrapper to the switching node. The wrappers handle IP and network port differences such as transaction handling, port width, endianness, etc. The wrappers also contain buffers if necessary, and act as an interface between the IP block clock domain and the network clock domain.

3. Network switching options

A network using circuit switching has low complexity switching nodes, because their main function is to connect an incoming link to an outgoing link. Deadlock avoidance is easily achieved since the circuit setup can either succeed or fail, but it cannot stall somewhere in the process. The drawback of circuit switching is that it locks resources for the duration of the data transfer. However this can be alleviated by limiting the data payload size. This problem can also be addressed during the network synthesis phase, by ensuring that resources that communicate with each other frequently are placed in close proximity to each other. Packet switching suffers from latency problems where the packet delay through the network can be several hundreds or even several thousands of cycles depending on the routing algorithm and the switch implementation. Circuit switching has a clear advantage over packet switching since the data latency is only dependent on the distance and there is no dependency on other factors like other traffic in the network, etc. The only dependency on the traffic situation in a circuit switched network is when setting up a route. Our OCN uses a combination of circuit switching and packet switching, which is described in section 4.

4. Network transaction handling

4.1 Route setup flow

The network transactions consist of four to six phases depending on whether the first routing try is successful or not. A successful transaction has four phases. (I) First a request is sent from the source to the network. As this request finds it way through the network the route is temporarily locked and cannot be used for any other transaction. (II) The second phase starts when the request reaches its destination. An ‘acknowledge’ is sent back along the route and the locks are changed to permanent locks (referring to the current transaction time span only). (III) When the ‘acknowledge’ has returned to the source the third phase starts. This phase holds the actual transfer of the data payload. (IV) Finally after the data has been transferred a ‘cancel’ request is sent that releases all resources as it follows the route. Fig. 2(a) details a successful transaction.

If a route is blocked in a node the routing request is canceled by (Ia) the blocking switch returning a ‘negative acknowledgement’ to the source. The source must then retry (Ib) the route at a later stage, as shown in Fig. 2(b).

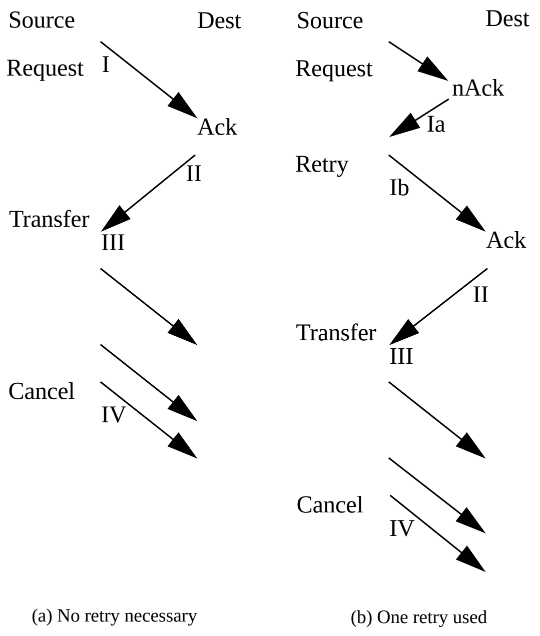

Fig. 2. Two successful circuit setups

4.2. PCC : Packet connected circuit

We refer to the novel hybrid circuit switching with packet based setup introduced in section 4.1 as “packet connected circuit” or PCC for short. The PCC has very nice properties as follows :-

(a) The PCC is deadlock free since no resources are locked while waiting (indefinitely) for other resources.

(b) The routing hardware in the switching node becomes very simple since no special cases, stalls, or virtual channels must be considered.

(c) The buffer space is minimized, since only the request packet is buffered in the switching node.

(d) There is no inherent limit on route selection algorithms in the PCC scheme.

4.3 Routing

A minimum path-length routing algorithm has been selected. Each switch has the knowledge of the general direction to the destination, i.e. north, south, east or west, and combinations of these, e.g. north-west. Since the network does not change when the chip has been designed, this knowledge is static and decided at the time of high-level synthesis of the network. The routing decisions are simply based on the destination address and the known direction. If there is more than one direction that leads to the destination, one is selected. If the primary selection is occupied, the second choice will be used. If there are no free outputs that lead towards the destination, the routing fails, and the switch will return a ‘negative acknowledgement’ to the source.

5. Switching Node design

The interface to the switching node is shown in Fig. 3. In total 19 wires are used in each direction. 16 wires carry forward going data and routing request packets, 1 wire is used for forward control, and 2 wires are used for reverse control. The forward control handles framing of transmissions and clock information to allow for easy retiming of the transfers when using mesochronous clocking (i.e. same frequency, but unknown phase). The reverse control carries the acknowledgements. The diagonal line in the figure represents a similar interface to the local IP wrapper.

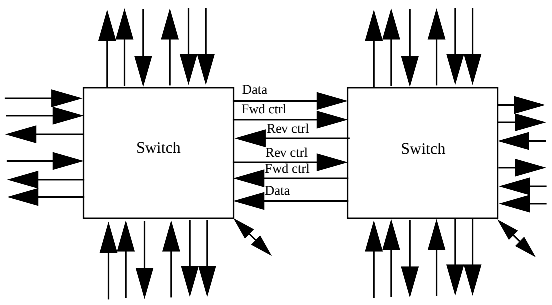

Fig. 3. Interface to the switching node

5.1 Clocking methodology

Considering the high clock rate and the distributed nature of an on-chip network, the wire delays between the components become a serious problem if using a traditional synchronous design methodology. In order to allow for wire delay and skew, we propose the use of mesochronous clocking with signal retiming in the system [8]. Using mesochronous clocking and retiming still requires the wire delays within a link to have reasonable skew but allows the design to use links that have differing delays without any problem. To further simplify the connection of network components, we propose using optimized drivers and transmission-style wires. This will ensure that no repeaters are necessary, the power consumption is kept to a minimum, and there is no need of laying out the network in an ordered fashion on the chip.

5.2 Switching Node block diagram

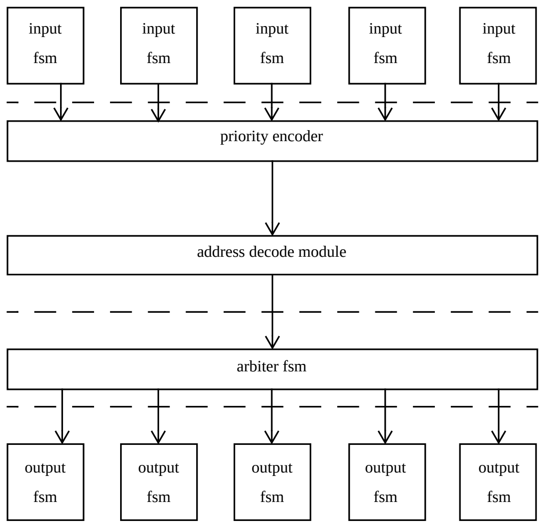

Fig. 4. 5-input, 5-output, pipelined Switching Node

Fig. 4. is the block diagram of the switching node. The dashed lines indicate pipelining. This implementation allows for easy scalability. The number of IP cores connected to the switching node can be increased simply by instantiating the additional number of input fsm's and output fsm's. Since only one routing packet is serviced at a time, the address decode module does not change with the number of IP cores. The priority encoder and the arbiter fsm can be changed easily using parameterizable Verilog modules.

5.3 Input FSM

The input fsm accepts the incoming routing packet, which is framed by the forward control signal. The output from the input fsm is a flag, which indicates that a routing packet has arrived. Fig. 5. shows the flow-chart

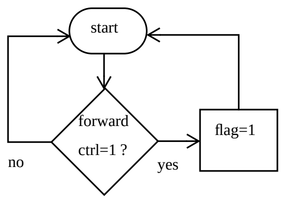

Fig. 5. Input FSM

for the input fsm.

5.4 Priority encoder

As seen in Table 1, the 5 inputs of the switching node have been assigned fixed priorities. Although this scheme is not fair to all inputs, simulations have indicated that this will not be a likely cause of congestion in the network. A 'high' output signal indicates that the corresponding routing packet at the input is chosen for routing. The priority encoder performs the 'input-side' arbitration for the case when multiple routing packets arrive at the different inputs of the switching node simultaneously.

Table 1: Truth table of a 5x5 priority logic block

flag [0]	flag [1]	flag [2]	flag [3]	flag [4]	output[0]	output [1]	output [2]	output [3]	output [4]
0	0	0	0	0	0	0	0	0	0
1	x	x	x	x	1	0	0	0	0
0	1	x	x	x	0	1	0	0	0
0	0	1	x	x	0	0	1	0	0
0	0	0	1	x	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1

5.5 Address Decode module

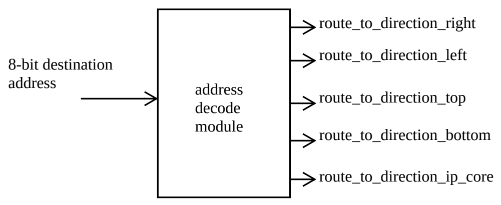

Fig. 6. Address Decode module

The destination address field, which is a part of the routing packet, is 8 bits wide. The address decode unit compares the higher 4 bits and the lower 4 bits of the destination address with the higher 4 bits and the lower 4 bits of the switching node address respectively to determine the routing direction for the input routing packet. The address decode module has been constructed using multiplexors and 4 bit comparators. The output from the Address Decode module gives possible routing directions to the destination. If the destination address is equal to the address of the switching node, this indicates that the destination of the input routing packet is the IP core connected to the switching node.

e.g. if the current address of the switching node is (2,2), and the destination address is (3,3), the x and y-coordinates of the destination address are greater than the x and y-coordinates of the address of the switching node. In this case the Address Decode module will assert the signals 'route_to_direction_right' and 'route_to_direction_bottom'. These signals are the inputs to the arbiter fsm, which will try to establish a route in either of these two directions.

5.6 Arbiter FSM

The arbiter fsm does the 'output-side' arbitration. An output port being used by an input port, is unavailable for use by the remaining input ports. The arbiter fsm routes the routing packet to the appropriate output port, if the output port is available, i.e. not locked by a routing packet from another input. If no output port is available, the arbiter fsm indicates its inability to establish a route by sending a ‘negative acknowledgement’. The arbiter fsm unlocks the output ports and makes them available for routing to other input ports upon successful completion of the data transfer, or if the destination indicates its inability to accept the data payload from the source.

5.7 Output FSM

The function of the output fsm is to forward the routing packet using the forward control framing signal.

6. Latency analysis

There are two primary types of latency in the network. One is related to the route setup time and another is related to the payload transfer. Both latencies are linearly dependent on the distance between the source and the destination. The route setup latency consists of first the request handling latency, which is 6 network clock cycles per switch for request buffering and route selection. The second part of the route setup latency is the acknowledgement latency that is one network clock cycle per switch.

Since the network is circuit switched the data transfer latency is just one network clock cycle per switch to allow for re-timing.

Due to the high internal clock rate in the network, the latency will appear lower from the IP block perspective. With a network clock frequency of 1.2 GHz and the typical IP block clock frequency of 300 MHz the apparent latency will be only a fourth. The route setup latency for every switch will appear as 1.75 cycles and the data transfer latency as 0.25 cycles.

The wrappers will incur some setup latency of their own. A discussion about the setup latency incurred by the wrappers has been omitted, since the wrappers have yet to be implemented. Also, a discussion about the data transfer latency on account of the wrappers has been omitted.

We emphasize that our OCN architecture provides guaranteed throughput and latency, after a route setup has been successful. While a successful route setup cannot be guaranteed, this problem can be solved by proper scheduling at the software level. This trade-off has enabled us to minimize the complexity of the switching node [2,3].

7. Experimental results

Switching nodes with a different number of input-output ports have been synthesized using Cadence PKS in 0.18 micron technology. Fig. 7. shows the area requirements for the different switching nodes. For future OCN’s, keeping the area of the switching fabric as low as possible is an important consideration. This will enable more IP’s to be integrated into the network. It is evident from the above figure that we have achieved this objective.

8. Conclusions

An area-efficient design of a switching node for future OCN applications has been presented. The switching node is ideal for applications requiring high throughput and low latency. The architecture of the switching node has been described, and the impact of the number of ports on the area has been shown.

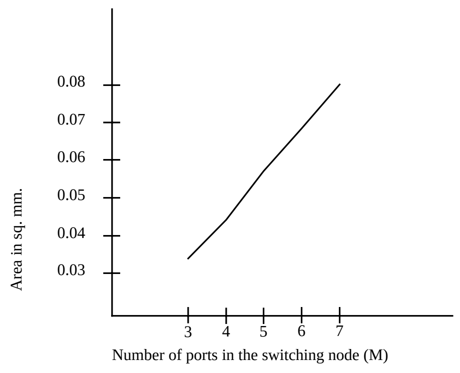

Fig. 7. The area of M-input, M-output, switching node

9. Future work

Wrappers for commonly used IP cores such as ARM, etc. will be designed. A demonstrator system is planned, which will feature the complete OCN solution consisting of the switching nodes, wrappers and IP cores.

References

[1] D. Wiklund and D. Liu. “SoCBUS : Switched Network on Chip for Hard Real Time Embedded Systems”. in IPDPS 2003.

[2] E. Rijpkema, et. al. “Trade Offs in the Design of a Router with Both Guaranteed and Best-Effort Services for Networks on Chip”. in DATE 2003.

[3] I. Saastamoinen, et. al. “Interconnect IP Node for Future System-on-Chip Designs”. in IEEE Internal Workshop on Electronic Design, Test and Applications, 2002.

[4] P. Guerrier and A. Greiner. “A generic architecture for on-chip packet-switched interconnections”. in DATE 2000.

[5] W. J. Dally and B. Towles. “Route packets, not wires: On-chip interconnection networks”. in DAC 2001.

[6] L. Benini and G. De Micheli. “Networks on Chips: A New SoC Paradigm”. in IEEE Computer, 35(1):70-78, 2002.

[7] K. Goossens, et. al. “Networks on silicon”. “Combining best-effort and guaranteed services”. in DATE 2002.

[8] F. Mu and C. Svensson. “Self-Tested Self-Synchronization Circuit for Mesochronous Clocking”. in IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 48, no. 2, Feb. 2001.