DESIGN OF A LOW LATENCY ROUTER FOR ON-CHIP NETWORKS

Sumant Sathe, Daniel Wiklund, Dake Liu,
Dept. of Electrical Engineering, Linköping University, Sweden

DOI: 10.36724/2664-066X-2021-7-4-16-20

ABSTRACT

A problem with long on-chip wires is the resulting delays, and repeater insertion becomes essential to mitigate this problem. Point-to-point wiring leads to an increase in the area and repeater insertion leads to an increase in power consumption. The complexity of System-on-Chip (SoC) designs con- tinues to increase, and traditional bus-based interconnects will not be sufficient to manage the communication requirements of future billion transistor chips. The properties of our OCN are now reviewed. The OCN provides reliable communication. This is achieved by ensuring that data dropping is not allowed. On-Chip Network's (OCN's) provide a scalable alternative to existing on-chip interconnects. The key element of the OCN is the router. We present a prototype design of a 5-input, 5-output, scalable router. The router is constructed from a collection of parameterizable and reusable hardware blocks and is a basic building block of the OCN. The router supports wormhole routing, and is characterized by an area of 0.3 mm2 in 0.18 micron CMOS technology.

KEYWORDS: *System-on-Chip, On-Chip Network's, On-Chip Network's.*

The article is reworked from unpublished 2nd IEEE International Conference on Circuits and Systems for Communications (ICCSC) materials.

INTRODUCTION

With the increasing complexity of SoC's, there is an increasing demand for high performance communication between the IP's (computation blocks) [1-4, 6-8]. The current solution for implementing SoC's with multiple processors, memories, etc. is to use a time-division multiplexed (TDM) bus, e.g. AMBA from ARM. All the communication is done over a shared transmission medium, so only a single master can drive the medium at any given instant. An arbitration mechanism has to be used to allow only one master to drive the shared medium at a time. This does not enable an extremely high-performance or a highly scalable solution.

A problem with long on-chip wires is the resulting delays, and repeater insertion becomes essential to mitigate this problem. Point-to-point wiring leads to an increase in the area and repeater insertion leads to an increase in power consumption.

OCN's provide an elegant solution to these problems because [1-4] they (a) 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, (d) are scalable when compared to traditional buses, and (e) allow multiple simultaneous transactions.

The properties of our OCN are now reviewed. The OCN provides reliable communication. This is achieved by ensuring that data dropping is not allowed. The OCN is free FROM deadlock. This can be ensured if no resource 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 conforms to data ordering to eliminate the need for reordering modules. This minimizes the buffer space. The overall latency and hardware overhead of the OCN is minimal. This is achieved by the use of wormhole routing.

In this paper, we describe our OCN architecture and explain how transactions are handled. We present the design of a 5-input, 5-output, scalable router suitable for our OCN.

OCN ARCHITECTURE

The OCN has been implemented as a 2-dimensional mesh as seen in Figure 1 [5]. It uses 5-port routers that allocate four ports to connect to other routers and one port to connect to the local IP block interface (port).

The local IP port is connected through a wrapper to the router. The wrappers handle IP and network port differences such as transaction handling, port width, endianness, etc. The wrappers act as an interface between the IP block clock domain and the interconnect (OCN) clock domain.

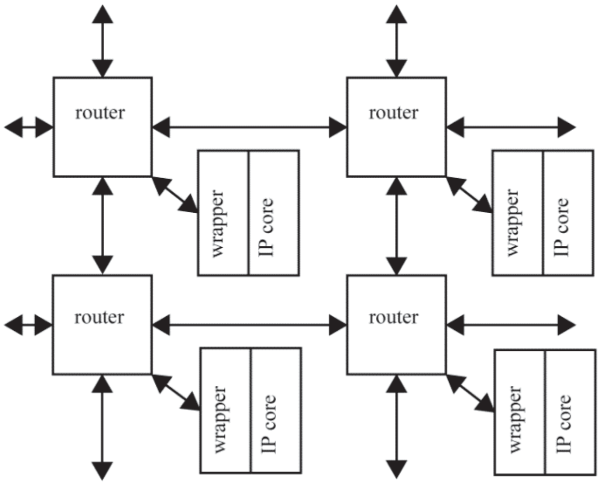

Fig. 1. A 2 x 2 OCN with routers, wrappers and IP blocks

NETWORK SWITCHING OPTIONS

A network using circuit switching has low complexity routers because their main function is to connect an incoming link to an outgoing link. Deadlocks are easily avoided since the circuit setup can either succeed or fail, but it cannot stall somewhere in this process.

Packet switching leads to more complex routers as the router has to buffer every packet before routing it, or the router has to use several virtual channels, or the router has to restrict the possible paths, to avoid deadlocks. Packet switching also suffers from latency problems, wherein the packet delay through the OCN can be several hundred or even several thousand cycles, depending on the routing algorithm and router implementation. Even for a wormhole routing network there is a possibility that the packets will be stalled for a long time due to other traffic.

This is because there is always a statistical distribution of packet delays in a packet switched network, which could also lead to the out-of-order arrival of packets at the destination. Circuit switching has an advantage over packet switching since the data transfer latency is only dependent on the distance, and there is no dependency on other factors, viz. other traffic in the network. The only dependency on the traffic situation in a circuit switched network is when setting up a route. All data is also guaranteed to arrive in the same order it is sent.

NETWORK TRANSACTION HANDLING

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 its 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 to the source. (III) When the "acknowledge" has returned to the source, the third phase starts. The actual transfer of the data payload is done during this phase. (IV) Finally after the data has been transferred, a 'cancel' request is sent that releases all resources as it follows the route. Figure 2 shows a successful transaction.

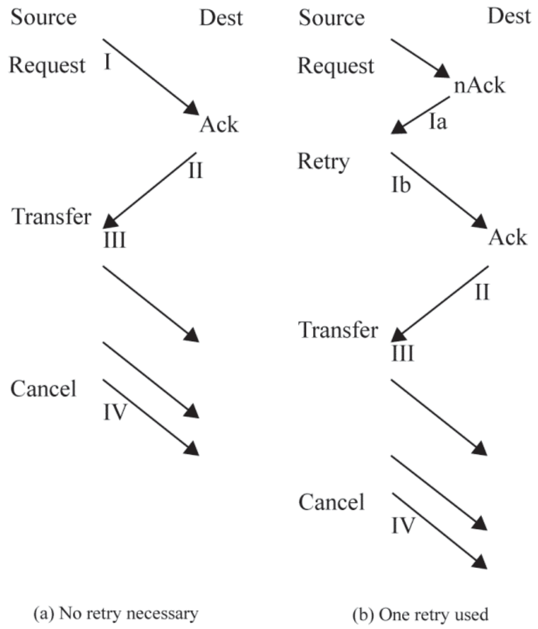

Fig. 2. Two successful circuit setups

PCC: Packet Connected Circuit

We refer to the novel hybrid circuit switching with packet-based setup introduced in section *Route Setup Flow* as "packet connected circuit" or PCC for short. PCC has the following nice properties:

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

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

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

(d) No central controller is required, and hence it is scalable.

Routing

A minimum path-length algorithm has been selected. Since the network does not change after the chip has been manufactured, 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 to the destination, routing is not possible and the router will send a "negative acknowledgement" to the source.

ROUTER DESIGN

The interface to the router is shown in Figure3. 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 routing request packet is 16 bits wide, and comprises of the 8 bit destination address and 8 auxiliary bits.

The forward control handles the 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 positive and negative acknowledgements. The diagonal line in the figure represents a similar interface to the local IP wrapper.

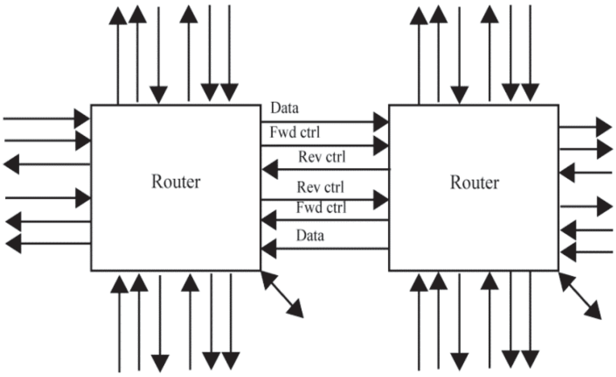

Fig. 3. Interface to the router

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 in 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 OCN [9].

Router Block Diagram

Fig. 4 is the block diagram of the router. There is one input fsm, one fifo, one address decode module, and one output fsm for each port of the router. This implementation allows for easy scalability.

The number of IP cores connected to the router can be increased by instantiating the additional number of input fsm's, fifo's, address decode modules, and output fsm's. The priority encoder and the arbiter can be easily scaled using parameterizable Verilog modules.

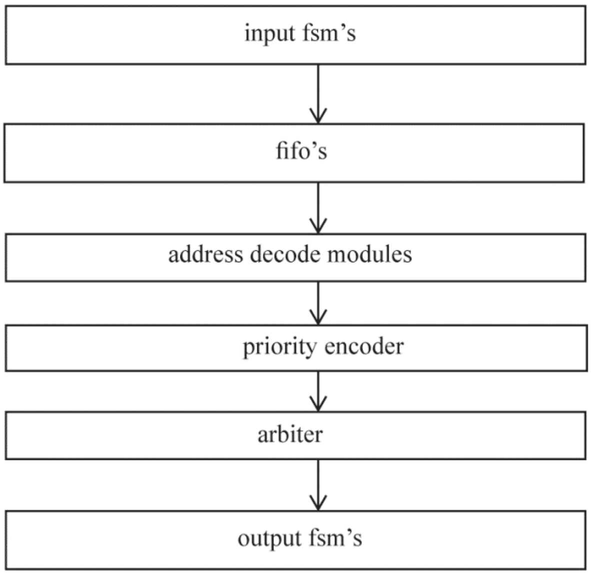

Fig. 4. 5-input 5-output router

Address Decode Module

In Figure 5, 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 router address respectively to determine the routing direction for the input routing packet.

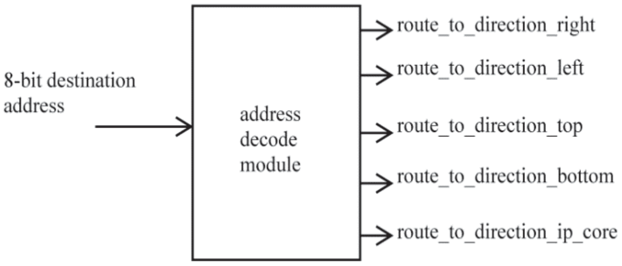

Fig. 5. Address Decode module

Priority Encoder

The 5 ports of the router have been assigned fixed priorities. The priority encoder performs the input-side arbitration for the case when multiple routing packets arrive at the different inputs of the router simultaneously. Although this fixed-priority scheme is not fair to all inputs, and contributes to congestion in the network, this scheme has been adopted to save area.

Arbiter

The arbiter 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 routes the routing packet to the appropriate output port, if the output port is available, i.e. not locked by another input. If no output port is available, the arbiter indicates its inability to establish a route by sending a 'negative acknowledgement'.

LATENCY ANALYSIS

There are two primary types of latency in the network. One is related to the route setup time and the other 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 4 network clock cycles per router for request buffering and route selection. The second part of the route setup latency is the acknowledgement latency, which is 1 network clock cycle per router. Since the network is circuit switched, the data transfer latency is just 1 network clock cycle per router to allow for retiming.

During the 4 clock cycles that it takes for a router to forward the routing packet to the next router, the incoming data is stored in the FIFO in the router. In the previous implementation of the router [10], this was not possible. In the previous version of the router, the data transfer from the source could not begin until the connexion was setup.

As soon as a connection is set up, this data (stored in the FIFO) is then streamed out to the destination. If a connexion cannot be set up, the data in the FIFO is cleared, and a negative acknowledgement is sent to the source. Since it takes 4 clock cycles for the router to forward the routing packet, the FIFO has to store 4 data packets in the best case. In the worst case, if all ports receive routing packets simultaneously, the FIFO in the lowest priority input has to store 8 packets.

The overall worst case FIFO size is thus equal to the worst case FIFO size per router multiplied by the worst case number of hops between the longest path through the OCN. On account of the high network speed desired, the FIFO's have been implemented using flip-flop's, and not dual-port RAM's. The FIFO area (16*5 slots) is a significant percentage of the router area, but this trade-off was made with the goal of minimizing the latency through the OCN.

Due to the high internal clock rate in the network, the latency will appear lower from the IP block perspective. With a targeted network clock frequency of 1.2 GHz and a targeted route setup latency of 6 clock cycles, and a typical IP block clock frequency of 300 MHz, the apparent route setup latency will be only 1.75 cycles. The data transfer latency will appear as 0.25 cycles per router.

The wrappers will incur some setup and data transfer latency of their own. A discussion about the latency due to the wrappers has been omitted, since the wrappers have yet to be implemented.

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 router.

CONCLUSIONS

An area-efficient design and architecture of a router for future OCN applications has been presented. The router is ideal for applications requiring high throughput and low latency. The router has been synthesized using Cadence PKS in 0.18 micron CMOS technology, and has an area of 0.3 mm^2.

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 routers, wrappers, and IP cores.

REFERENCES

[1] P. Guerrier, A. Greiner. A generic architecture for on-chip packet interconnections, *in DATE* 2000.

[2] W. Dally, B. Towles. Route packets, not wires: On-chip interconnection networks, *in DAC* 2001.

[3] L. Benini, G. De Micheli. Networks on Chips: A New SoC Paradigm, *IEEE Computer*, 35(1):70-78, 2002.

[4] K. Goossens, et. al. Networks on silicon: Combining best-effort and guaranteed services, *in DATE* 2002.

[5] D. Wiklund, D. Liu. SoCBUS: Switched Network on Chip for Hard Real-Time Embedded Systems, *in IPDPS* 2003.

[6] E. Rijpkema, et. al. Trade Offs in the Design of a Router with both Guaranteed and Best-Effort Services for Networks on Chips, *in DATE* 2003.

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

[8] P. Pande, et. al. Design of a Switch for Networks on Chip Applications, *in ISCAS* 2003.

[9] F. Mu, C. Svensson. Self-Tested Self-Synchronization Circuit for Mesochronous Clocking, *IEEE TCAS-II: Analog and Digital Signal Processing*, vol. 48, no. 2, Feb. 2001.

[10] S. Sathe, et. al. Design of a Switching Node (Router) for On-Chip Networks, *2003 5th International Conference on ASIC (ASICON2003).*