A High-Throughput Network-on-Chip Architecture for Systems-on-Chip Interconnect

A. Bouhraoua
Computer Engineering Department
King Fahd University of Petroleum and Minerals
P.O.Box 969, Dhahran, 31261
Saudi Arabia
Email: abouh@ccse.kfupm.edu.sa

M. E. Elrabaa
Computer Engineering Department
King Fahd University of Petroleum and Minerals
P.O.Box 969, Dhahran, 31261
Saudi Arabia
Email: elrabaa@ccse.kfupm.edu.sa

***Abstract*— A buffer-less, contention-free, Network-on-Chip architecture based on a modified Fat Tree is proposed. Simulations results show that the proposed architecture achieves maximum throughput (> 90%) way above the 40-50% seen in conventional Fat Trees. Contention is eliminated and latency is reduced through an improved topology and router architecture. Area of the network is kept to a minimum by pushing the buffers to the edge of the network at the client interface. Simulation results show that the required number of buffers at the client interface is a fraction of the theoretical maximum. This means that the actual number of buffers can be tailored to suit a class of applications running on a specific platform.**

I. Introduction

Networks-on-Chip (NoCs) have emerged as an alternative to ad-hoc wiring or bus-based global interconnect in SoCs [1-4]. This is to cope with the growth of Systems-on-Chips (SoCs) complexity and their very short time to market constraints. SoCs are mainly built by integrating many IP (Intellectual Property) blocks from different vendors to form the desired system. Connecting these different IP blocks together poses many issues at several levels of the design and implementation stages. Communication protocol compatibility, bandwidth requirements and performance are but some of the design issues. Clock distribution and the overall timing closure of the whole chip are implementation problems. NoCs provide a systematic solution for these issues. Hence, the general consensus is that the communication requirements, as well as the design flow, of billion transistors SoC are best accommodated by shared, segmented interconnection networks [1,2].

The intensive study of interconnection networks in the 80s/90s for the purpose of connecting parallel processors produced many solutions that are adapted now for the NoC area. Since [2], the majority of the proposed NoCs have been directly derived from existing topologies and routing algorithms [5-7]. Some other NoCs focus only on the router architecture alone neglecting the other components of the network like the topology, the flow control and the routing scheme [8-9].

As a result of the above strategies, the complexity of routers did not change significantly compared to inter-chip interconnection networks, where routers were designed to fit on a single chip. However, NoC routers should be designed so that several instances can be easily integrated on-chip with a relatively negligible overhead.

Other contributions have introduced new routing schemes in order to reduce the router gate count [10][15]. The Nostrum network, built using a 2-D mesh with deflection routing, has taken the direction of reducing the size of the router. Deflection, routing enables a buffer-less router design thus maintaining a relatively low-cost. However, because of deflection routing and the mesh structure, latency and throughput are both sacrificed [10]. The throughput levels reached by the Nostrum router are just a small fraction of the maximum available bandwidth [11] and saturate very quickly [10]. Deflection routing imposes the adoption of a store-and-forward routing mechanism instead of wormhole routing because of deadlocks [12]. The store-and-forward mechanism introduces more delay. Packet size is very small (96 bits) to keep the inter-node wiring acceptable (128 bits) and the size of the input buffers small enough to claim the buffer-less status.

Repeating the same network topologies and routing algorithms as in inter-chip interconnection networks does not fully take advantage of the on-chip property. Designers are no longer limited by the number of I/O ports they can use. This is a real advantage that has not been stressed enough in the existing solutions.

In this work, a novel approach for efficiently designing NoCs is presented. The adopted design strategy is outlined in section 2 along with the supporting arguments. This includes an analysis of NoC designs in general and their key properties. A detailed description of the proposed approach is presented in section 3. The basic properties of the FT are analyzed first to lay the background for the proposed improved FT and router architecture. Simulation results, that show the performance of the proposed approach, are presented in section 4 followed by conclusions in section 5.

II. Design Strategy

The adopted strategy can be summarized as follows:

1. To take advantage on the on-chip context and produce a high-throughput reliable architecture. This means contention has to be eliminated.

Contention occurs when packets compete for the same output port. Hence, having more output ports than input ports in the router eliminates contention.

2. Adopt a wormhole routing to eliminate buffers (and reduce the router's footprint) and reduce latency. Since contention is also eliminated, wormhole routing won't have any adverse affect on link utilization or cause deadlocks.

Hence adopting wormhole routing and increasing the number of output ports would simplify the router design and eliminates buffers thus compensate for the increased router size. Buffers are pushed to the edge of the network at the client side where they can be tailored for a specific class of applications. Also, adopting a parameterized architecture would enable the use of a single HDL (Hardware Description Languages) model to describe a family of routers with some parameterization, especially I/O ports.

III. The Proposed Approach

The proposed approach aims at developing a new class of NoCs based on a sub-class of Multistage Interconnection Networks topology (MIN). More particularly, a class of bidirectional folded MINs; chosen for its properties of enabling adaptive routing. This class is well known in the literature under the name of Fat Trees (FT) [5]. Although known under the same name, this class of networks is very similar but not identical with the original FTs as defined in [16]. The main goal of this approach is to arrive to a topology derived from the FT topology by increasing the number of links in order to eliminate contention.

Before introducing the adopted modified FT topology the regular FT, which is a type of Multi-Stage Interconnection Networks (MIN), properties are analyzed below.

A. *Fat-Tree MIN Properties*

1) *Routing*

Routing in folded FTs is reduced to routing in a binary tree [5], as shown in Figure 1. The output ports of a router that are connected to the input ports of upper stage routers are labeled as the UP links. The output ports of a router that are connected to the input ports of lower stage routers on the right are labeled RIGHT links. The output ports of a router that are connected to the input ports of lower stage routers on the left are labeled LEFT links. Routing in a tree makes packets take one of the three directions according to their destination address.

Figure 1 shows a regular fat tree. The "R" label designates routers while the "C" label designates clients. Input and output ports are connected using two unidirectional links represented as single bidirectional link in Figure 1.

A packet is routed up until it reaches a router that has a downward connection for it to reach its destination. This router is called routing summit for convenience. The FT structure, based on a superimposition of trees, naturally provides packets with several upward paths (adaptive). Any upward path will eventually lead to a "summit" where a downward path to the packet's destination is provided. The downward path to the packet's destination is unique (deterministic) as it is the case for any regular tree structure.

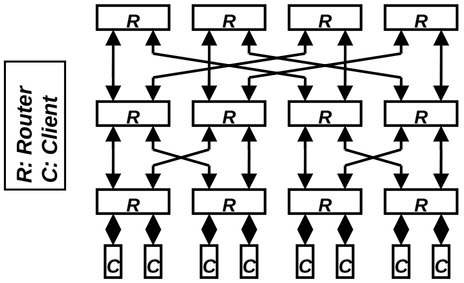

Figure 1 – Regular Fat Tree Topology

2) *Contention in FT MINs*

Considering a single router, packets coming from the bottom links are either routed up or routed down while packets coming from the up links are always routed down. The router where a packet coming from the bottom links is routed down is called a summit.

This means that packets coming from the up links are never routed up only packets coming from the bottom links are routed up. Since the number of up links is equal to the number of bottom links, there cannot be any contention when routing up. Contention occurs when going down. Because of the fact that the bottom links are split in right and left links, deterministic routing of packets will lead to contention. Clearly, if several packets coming from the up links need to go right, there will be a contention. This means that one of them will earn the right to use the link while the others will be waiting for it to complete.

B. *Modified FT MIN*

Contention can be removed if there are enough output ports in a router so that they can accommodate any combination of incoming packets. Since Contention occurs only on the downward path, doubling the number of output ports in the downward direction only will eliminate the contention. This is the adopted strategy for the proposed modified FT.

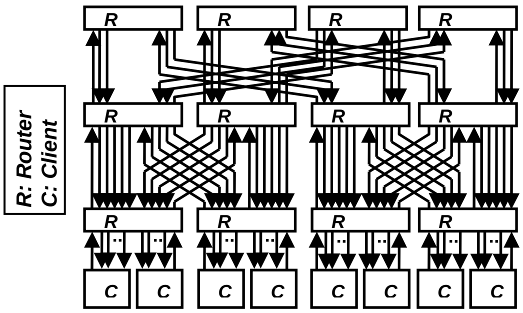

Figure 2 – Modified FT Topology

Figure 2 shows the modified FT where the down links are doubled. The links are shown as unidirectional links to show the doubling of the down links. Doubling the output ports of a router also means doubling some of the input ports of the adjacent router, of lower stage, to which it is connected. This is needed to be able to connect all the output ports of the upper stage router. To avoid contention in the lower stage router, its output ports are twice as much as its input ports and four times the input ports of its upper stage adjacent router.

This modification does not induce any changes to the routing function which stays the same as for the regular Fat-Tree topology.

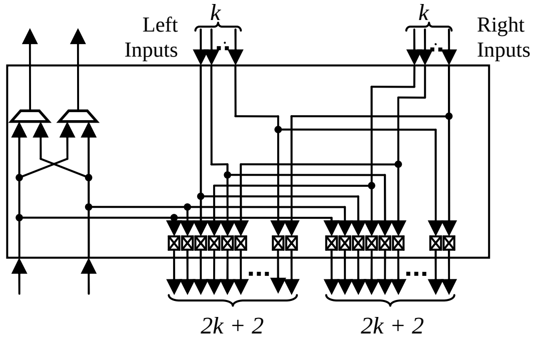

Figure 3 – The router Architecture

Figures 3 show the adopted router architecture. A simple routing function circuit has been implemented. The router does not contain any routing tables. The size of the clients' address space reachable using the downside ports of this router is equal to 2^{r-1} *x 2* which is 2^r. It is always a continuous interval of addresses of the form *[l, u]*. The lower bound *l* corresponds to the smallest address that can be reached from the router *(r,c)*. The upper bound *u* corresponds to the largest address that can be reached from the router *(r,c)*. Simple magnitude comparators can be used to determine the routing of a packet.

C. *Client Interface*

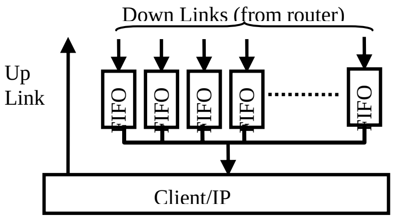

Figure 4 – Client Interface

In the proposed architecture, buffering is transferred from the routers to the client interfaces. Figure 4 shows a block diagram of the client interface. Since, it is possible for a client to receive several packets simultaneously, it is necessary to accommodate this traffic. To achieve that, each incoming link is terminated with a FIFO memory. The different FIFO memories are all connected to the client through a single shared bus. This bus can be wider to perform data transfers faster than what is received in the FIFOs. The size of these memories may vary according to what is required by the client interfaces.

IV. Performance Evaluation

Simulations were used to evaluate the performances of the modified FT. The simulation platform is a cycle-based C program. Two networks were simulated: the regular FT and the proposed modified FT. The traffic generator model follows a uniform distribution. Variable size packets were generated. Packet size was randomly generated within a predetermined range. Latency is measured by counting the delay (cycles) between the time the end of the packet enters the network and the time its last byte leaves one of the client's FIFOs. The link size is set to one byte.

A fully parameterizable, synthesizable RTL router model has been developed. This model constitutes the main platform from which all the router models, each with a different number of I/O ports can be derived. A simple script can then be used to instantiate and connect the different router instances that make the target network.

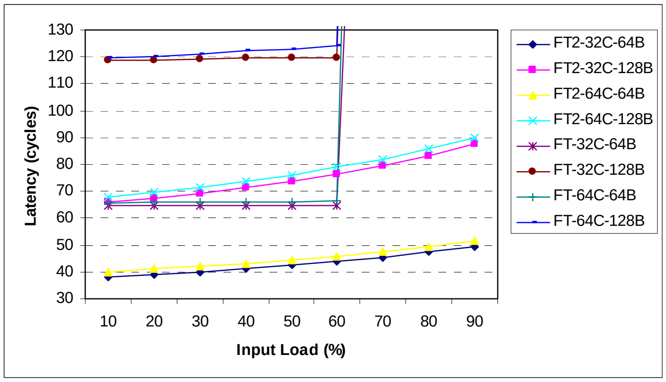

Figure 5 – Latency

A. *Throughput & Latency*

Simulations have shown that the regular FT (throughput) quickly saturates to around 60% of the maximum wire-speed bandwidth while the modified FT does not saturate at all. Figure 5 shows the average latency as a function of the input load for network sizes of respectively 32 and 64 clients for both the regular FT and the modified FT. The FT2 label on the figures refers to the modified FT. The label suffix of 32C or 64C designates a number of 32 and 64 clients respectively. Two packet sizes have been considered: 64 and 128 bytes. The label suffix of 64ML and 128ML designates a packet size of 64 and 128 bytes respectively.

The most important achievement is the fact that the throughput completely matches the input load proving the absence of contention.

B. *Area*

Doubling the links in the downward direction is the cornerstone of this design. One major concern from designers will be the cost estimation of the network size in the context of large SoCs with several connections. Table I gives the gate count and size of the proposed router compared to several published routers [5-7,17]. The size of the chosen router is the size of the middle row router in a 16 client network which represents a router with an average number of I/O ports (i.e. 8 ports). As this table shows, the proposed router has a significantly smaller area (0.06 mm^2). The area has been estimated from logic synthesis of the RTL model.

Routers in lower levels in NoCs with large number of clients would be larger; however they would also have an extremely high throughput. E.g. a bottom row in a network of 64 clients would have 22,750 gates, 32 inputs, 64 outputs, and an area of about 0.27 mm2. However, it would achieve a maximum throughput of 409.6 Gbits/s with a clock of 800 MHz in a 0.13 microns process.

TABLE I. ROUTER AREA

Source	Num. Links	Gates	Tech.(μ)	Area (mm^2)
[6]	5	20,000	0.13	0.25
[7]	5	-	0.35	0.61
[17]	5	-	0.25	2.89
[5]	8	-	0.25	0.8
[10]	5	13,000	Lsi10k	-
FT2	8	5,000	0.13	0.06

C. Buffer Utilization

A tradeoff of this architecture is that buffers are pushed out of the routers to the client interfaces. This means that a considerable amount of buffer lanes is necessary for every client interface. Simulations were conducted in order to measure the actual number of FIFO lanes that are simultaneously used at anytime. Figure 6 shows the result of these simulations. It shows a linear progression of the maximum number of lanes used during operation. The obtained figures are an order of magnitude lower than the number imposed by the architecture. Hence the number of buffer lanes in the client interface can be tailored for a specific platform to suit the class of applications at hand while reducing buffering area.

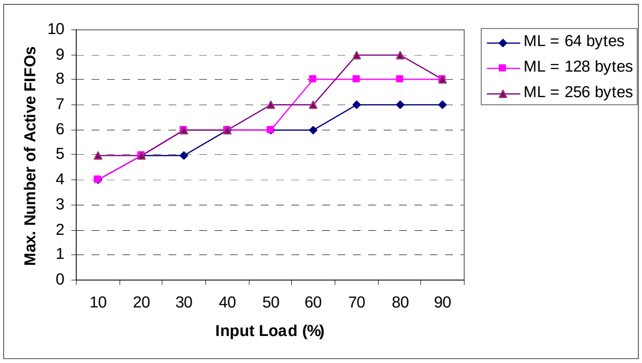

Figure 6 – Maximum number of FIFO lanes simultaneously active

V. CONCLUSIONS

A contention-free modified FT architecture is proposed. Cycle-accurate simulations show that the proposed architecture achieves maximum theoretical throughput and has smaller latency than conventional FTs. Simulations also show that latency increases linearly with input load. The achieved performance for the modified FT is actual performance using a contention-free network. The area of the network is kept small because of the absence of buffers in the router architecture. The number of buffer lanes in the client interfaces can be tailored for a specific platform to suit the class of applications at hand while reducing buffering area.

ACKNOWLEDGMENT

Facilities support by King Fahd University of Petroleum and Minerals is highly appreciated by the authors.

REFERENCES

[1] L. Benini and G. D. Micheli, “Networks on chips: A new SoC paradigm”, *IEEE Computer*, 35(1):70 – 78, January 2002.

[2] W. J. Dally and B. Towles. Route packets, not wires: On chip interconnection networks. In *Proceedings of the 38thDesign Automation Conference*, pages 684–689, June 2001.

[3] J. Henkel, W. Wolf, and S. Chakradhar, “On-chip networks: a scalable, communication-centric embedded system design paradigm,” in *Proc. 17th Int’l Conf. VLSI Design*, 2004, pp. 845-851.

[4] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Oberg, M. Millberg, and D. Lindqvist, “Network on chip: an architecture for billion transistor era,” in *Proc. IEEE NorChip Conf.*, 2000.

[5] P. Guerrier and A. Greiner, “A Generic Architecture for On-Chip Packet-Switched Interconnections”, *Proc. IEEE Design Automation and Test in Europe (DATE 2000), IEEE Press, Piscataway, N.J., 2000, pp. 250-256.*

[6] E. Beigne, F. Clermidy, P. Vivet, A. Clouart and M. Renaudin, “An Asynchronous NOC Architecture Providing Low Latency Service and its Multi-Design Framework”. In *Proceedings of the 11th Symposium on Asynchronous Circuits and Systems (ASYNC’05).*

[7] P. Zipf, H. Hinkelmann, A. Ashraf, M. Glesner, ” A Switch Architecture and Signal Synchronization for GALS System-on-Chips”, *SBCCI'04, September7-11 , 2004, Pernambuco, Brazil*

[8] S. K. Hasan, A. Landry, Y. Savaria, M. Nekili, “Design Constraints of a HyperTransport-Compatible Network-On-Chip”, *IEEE 2004.*

[9] M.D. Osso, G. Biccari, L. Giovannini, D. Bertozzi and L. Benini, “xpipes: A Latency Insensitive Parameterized Network-on-Chip Architecture For Multi-Processor SoCs”, *Proceedings of the 21st International Conference on Computer Design (ICCD’03)*

[10] E. Nilsson. “Design and Implementation of a hot-potato Switch in a Network on Chip”. Master’s thesis, Royal Institute of Technology, IMIT/LECS 2002-11, Sweden, June 2002.

[11] E. Nilsson, M. Millberg, J. Öberg, and A. Jantsch, Load Distribution with Proximity Congestion Awareness in a NoC, Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE’03)

[12] C. J. Glass and L. M. Ni, “The turn model for adaptive routing”. In Proceedings of 19th Annu. Int. Symposium on Computer Architecture, May 1992.

[13] E. Nilsson and J. Öberg, “Reducing power and latency in 2-D mesh NoCs using globally pseudochronous locally synchronous clocking”, CODES+ISSS 2004.

[14] E. Nilsson and J. Öberg, “Trading off power versus latency using GPLS clocking in 2D-mesh NoCs”, ISSCS 2005

[15] K. Goossens, J. Dielissen, A. Radulescu, “Æthereal network on chip: concepts, architectures, and implementations”, IEEE Design and Test of Computers, Volume 22, Issue 5, Sept.-Oct. 2005 Page(s)414 – 421

[16] C. Leiserson, "*Fat-Trees: Universal Networks forHardware-Efficient Supercomputing*", IEEE Transactions on Computers, vol. C-34, no. 10, pp. 892-901, October 1985.

[17] H. C. Chi and J. H.Chen, “Design And Implementation Of A Routing Switch For On-Chip Interconnection Networks”, *2004 IEEE Asia-Pacific Conference on Advanced System Integrated Circuits(AP-ASIC2004) Aug. 4-5, 2004, pp 392-395*