AN EFFICIENT NETWORK-ON-CHIP ARCHITECTURE BASED ON THE FAT-TREE (FT) TOPOLOGY

Abdelhafid Bouhraoua† and Muhammad E. S. Elrabaa
Computer Engineering Department
College of Computer Science and Engineering
King Fahd University of Petroleum and Minerals

الخلاصة:

سيتم وصف وتقديم طريقة جديدة لتصميم الشبكات الموجودة داخل الدارات الالكترونية ذات الكفاءة الفائقة باستخدام طوبولوجيا الشجرة العريضة المعدلة. هذا ويقوم النهج المقترح بإزالة التنافس وتخفيف الكمون من خلال تحسين الطوبولوجيا ومسارات التوجيه. وتؤدي الطوبولوجيا المعتمدة إلى زيادة الأداء بدون زيادة كبيرة في تكلفة التوجيه. وقد تم تحقيق هذا بتبني تصميم بسيط للتوجيه والذي لا يحتوي على ذاكرة للتخزين ويُتيح توجيها للبيانات خالياً من التصادم. وتُظهر نتائج المحاكاة أن مستوى الأداء الذي حققته كل من الطوبولوجيا ومسار البنية (مستوى الناتج > 90%) أعلى بكثير مّما يوجد حالياً في الشبكات الأخرى (40–50%). كما تُظهر تقديرات المساحة المبنية على التوليف المنطقي أن التوجيه المقترح يحتاج المساحة اقل بكثير من الموجهات الحالية. ويرجع ذلك إلى بساطة التوجيه المستخدم وإزالة الذاكرة من هذه الموجهات.

† To whom correspondence should be addressed.
P. O. Box 5065
King Fahd University of Petroleum & Minerals
Dhahran 31261, Saudi Arabia
E-mail: abouh@kfupm.edu.sa

Paper Received: 4 April 2007; Revised: 3 June 2007; Accepted: 18 September 2007

ABSTRACT

A novel approach for an efficient network-on-chip (NoC) using a modified Fat Tree is presented. The proposed approach totally eliminates contention and reduces the latency through an improved topology and router architecture. The adopted topology increases performance without a substantial increase in the routing cost. This is achieved by using an improved buffer-less, parameterizable router architecture. The proposed router architecture is simple to implement yet can achieve the required packet collision avoidance. Simulation results that show the level of performance achieved by both the topology and the router architecture are presented. A throughput of more than 90%, which is significantly above the 40–50% usually seen in other NoCs, is achieved. Area estimates based on logic synthesis results show that the proposed routers have significantly less area than other NoC routers. This is due to the simple routing function used and the removal of buffers.

***Key words*: networks-on-chip, systems-on-chip, ASICs, interconnection networks**

Abdelhafid Bouhraoua and Muhammad E. S. Elrabaa

AN EFFICIENT NETWORK-ON-CHIP ARCHITECTURE BASED ON THE FAT-TREE (FT) TOPOLOGY

1. INTRODUCTION

The current growth of systems-on-chips (SoCs) complexity and their very short time to market constraints had led to new design methodologies. 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.

The networks-on-chips (NoCs) paradigm has emerged as an alternative to ad-hoc wiring or bus-based global interconnect in SoCs [1-4]. It provides a systematic solution for issues of compatibility, bandwidth requirements, and performance, eases the clock distribution problem, and makes the over-all chip timing closure manageable. Hence, the general consensus is that the communication requirements, as well as the design flow, of billion transistor 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 [5–12]. 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 [9, 13, 14].

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. Also some of the serious limitations that were associated with inter-chip routers, such as pin count which limited the number of ports, are not significant within the on-chip context.

Many researchers have introduced new routing schemes in order to reduce the router gate count [15–20]. 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 [15]. The throughput levels reached by the Nostrum router are just a small fraction of the maximum available bandwidth [16] and saturate very quickly [15]. Deflection routing imposes the adoption of a store-and-forward routing mechanism instead of wormhole routing because of deadlocks [17]. 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.

2. THE ADOPTED DESIGN STRATEGY

As was explained in the introduction, most NoCs are based on derived solutions from the inter-chip interconnection networks area. Also, the throughput of most of these networks does not go beyond 40% to 50%. This is due to unavoidable conflicts in the allocation of router resources in these topologies. Conflicts occur when several packetss destinations are on the same path and are to be routed through the same output port. Ideally the router architecture should have more output ports than input ports to prevent such conflicts from occurring. This is the main strategy adopted in this work: to increase the number of output ports versus the number of input ports in the router. This might have posed many problems in the context of inter-chip interconnection networks where the number of pins is limited and the network's details are not known apriori. For NoCs, however, the number of pins is virtually unlimited and the network is frozen at design time when the number of client nodes (IP cores) and routers are fixed.

Also, adopting a parameterized router architecture would enable the use of a single HDL (Hardware Description Language) model to describe a family of routers with some parameterization, especially I/O ports. This approach was adopted as part of the design strategy.

3. THE PROPOSED APPROACH

The proposed approach aims at developing a new class of NoCs based on a sub-class of Multi-Stage 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 [21]. 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, properties of conventional FT MINs are analyzed below.

3.1. Topology of FTs

The network is organized as a matrix of routers with n rows; labeled from 0 to $n-1$; and $2^{(n-1)}$ columns; labeled from 0 to $2^{(n-1)} -1$. Each router of row 0 has 2 clients attached to it (bottom side). The total number of clients of a network of n rows is 2^n clients. The routers of other rows are connected only to other routers. Any router is identified by its position (r, c); r denoting its row index and c denoting its column index.

- A router $(0,c)$ is connected to two routers at row 1:
  - router $(1,c)$ and;
  - router $(1,c-1)$ or router $(1,c+1)$ depending on whether c is odd or even respectively.
- A router $(1,c)$ is connected to two routers at row 2:
  - router $(2, c)$
  - router $(2, c-2)$ or router $(2, c+2)$ depending on whether the value of $c/2$ is odd or even respectively
- In general, a router (r,c) is connected to two routers at row r+1:
  - router $(r+1, c)$
  - router $(r+1, c-2^r)$ or router $(r+1, c+2^r)$ depending whether the value $c/2^r$ is odd or even respectively

Hence, two clients can be reached from any router in row 0. A router at row 1 is connected downwards to two routers at row 0. This means that it can reach 2×2 = 4 clients. A router at row 2 is connected to two routers at row 1; thus reaching 2×4 = 8 clients. So, in general, a router at row r can reach $2^{(r+1)}$ clients.

Figure 1 shows a regular FT of $3 \times (2^{3-1})$ routers and 2^3 clients. Router (1, 2) is connected upwards to routers (2, 2) and routers (2, 2–2) because 2/2 is odd. Router (1, 2) is connected downwards to routers (0, 2) and router (0,3) because 3 is odd.

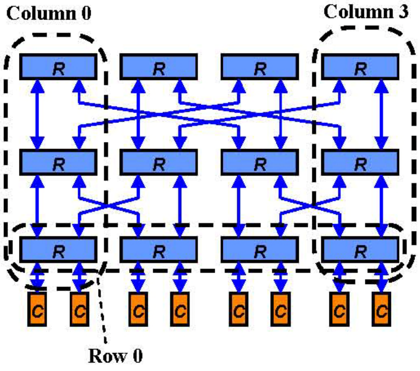

Figure 1. Regular fat tree topology

The FT network scales up to the next power of 2 (in terms of client size) by following a simple construct. A network of size 2^n is duplicated and each instance is instantiated side by side as shown in Figure 2. A new row of routers of width equal to 2 times the width of the original network is added on the top to enable communication between the two halves.

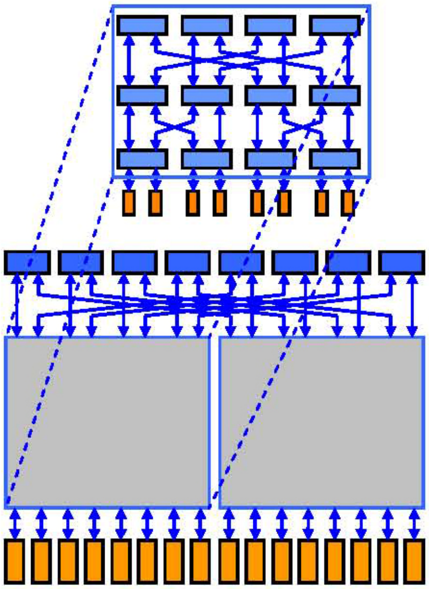

Figure 2. FT topology scaling

Figure 3 illustrates how the network is recursively built starting from a group of a single router with two clients and how these groups are connected to the upper rows routers.

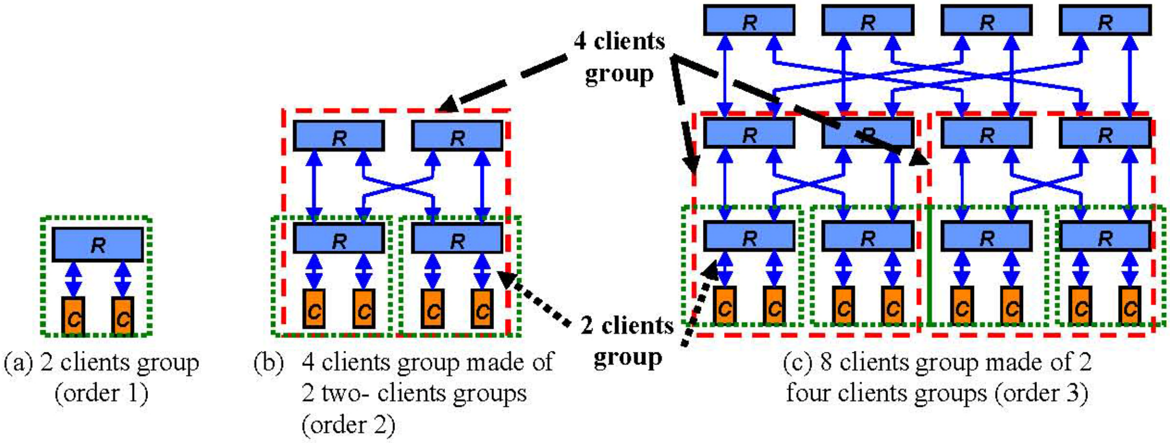

Figure 3. Router groups

A group of order *r* can be defined as follows (as illustrated in Figure 3):

- A structure of routers organized in *r* rows and 2^{r-1} columns.
- Routers at row 0 are always included in the group. Consequently, the clients, which are attached to routers at row 0, are also part of the group
- A group of order *r* is made of two adjacent groups of order *r–1*.
- The number of groups of order *r* in the network is equal to 2^{n-r} since the total number of columns is 2^{n-1}.

Any router of coordinates *(r, c)* is connected to two adjacent groups of order *r*. The same router belongs to the group of order *r*+1 that contains the two groups of order *r* it is connected to. Any router at row *r* has its left link connected to the group of order r on the left and its right link connected to the group of order r on the right.

3.2. Routing in FTs

A packet is routed up until it reaches a router that has a path to its destination. This router is called routing summit for convenience. The FT structure, based on a superposition of binary trees, naturally provides packets with several upward paths. 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 as it is the case for any regular tree structure. Hence routing up is adaptive while downward routing is deterministic.

Clients are addressed in an increasing order starting from the left to the right, *i.e.* all client addresses are within the interval $[0,\ 2^n\ –1]$. Each pair of clients are attached to a router at row 0, with the client addresses being related to the column coordinates c of the routers at row 0 as:

$$addr = 2*c + s\ ,$$

where $s = \{0,\ 1\}$, depending on whether the client is on the left (0) or the right (1). The routing direction is determined at each step by using the c index of the destination client address. Each router at position $(r,\ c)$ has two groups of clients of order r that can be reached from its downward links: a group G_L to its left with an associated address interval I_L and another to its right, G_R with a corresponding address interval I_R. Both address intervals are of length 2^r and are expressed as shown below:

$$I_L = [2c_L\ ,\ 2c_L + 2^r –1] \qquad \text{and} \qquad I_R = [2c_L + 2^r,\ 2c_L + 2^{r+1} –1]\ ,$$

where c_L is the lowest column index of routers in group G_L. Hence when a packet reaches a given router, the destination address (*daddr*) is compared to the router's I_L and I_R intervals to find whether the destination client belongs to G_L or G_R. If the destination client belongs to one of the groups, the packet is routed down on one of the downward links. If the destination client neither belongs to G_L nor to G_R, the packet is routed to one of the upward links as shown below.

```
if $daddr \in I_L$ then Route LEFT
else
    if $daddr \in I_R$ then Route RIGHT
    else Route UP
```

The intervals I_L and I_R are computed at design time and set in the routers as constants. For a router at location $(r,\ c)$, they are computed recursively as follows:

- If $r = 0$ then:

  $I_L(r) = [2c,2c]$ and $I_R(r) = [2c+1,2c+1]$ (*i.e.* one client in each interval)

- If $r \neq 0$ then:

  $I_L(r) = \{I_L(r–1)\ |\ I_R(r–1)\}_{\text{Left Link Router}}$ (*i.e.* the concatenation of intervals I_L and I_R of the router connected to its left link at row $r–1$)

  $I_R(r) = \{I_L(r–1)\ |\ I_R(r–1)\}_{\text{Right Link Router}}$ (the concatenation of intervals I_L and I_R of the router connected to its right link at row $r–1$)

Figure 4 shows the intervals associated with each routing direction where l, L, and u represent the lower limit of I_L, upper limit of I_L and upper limit of I_R, respectively. Packets with destination address outside I_L and I_R are routed up (left or right, whatever link available).

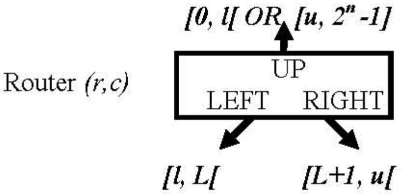

Figure 4. Routing in FT

3.3. Contention in FTs

Considering a single router, the cases for packet routing are:

- Packets coming from the bottom links are either routed up or routed down. In the latter case this router is a summit
- Packets coming from the top links are always routed down.

This means that packets coming from the top 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 only occurs when going down. Because of the fact that the bottom links are split into right and left links, deterministic routing of packets will lead to contention. For example, if several packets coming from the top 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 as shown in Figure 5.

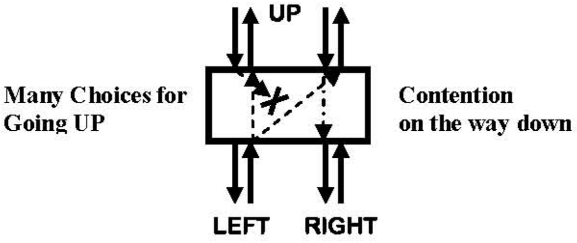

Figure 5. Contention in FTs

3.4. The Proposed Modified FT MIN

Contention can be removed if there are enough output ports in a router that 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 6 shows the modified FT where the down links are doubled from one level to the next, starting with two output (downward) ports per direction for the topmost routers. Hence at each level, the number of top input ports equal the number of downward output ports of the upper level. This modification does not induce any changes to the routing function which stays the same as for the regular Fat-Tree topology.

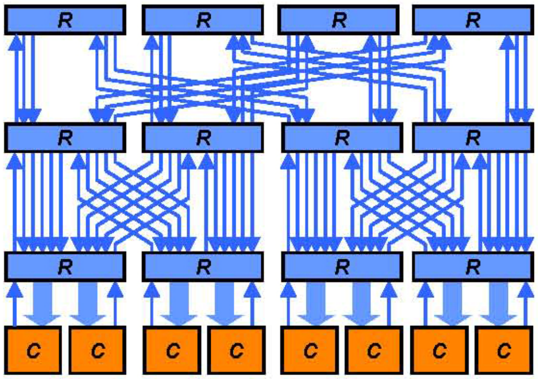

Figure 6. The proposed modified FT Topology

3.5. Router Architecture

Figure 7 shows the adopted router architecture with the doubling of the output ports on the downward direction. No internal FIFOs are required since no contention can occur. The circuit implementation of the routing function is shown in Figure 8. Simple comparators are used to determine the direction of outgoing packets based on the destination address and the interval limits (constants that are set at design time) with no need for storing any routing tables or any path calculations. The sheer simplicity of the router compensates for the increase in number of ports. This NoC structure can easily support backpressure implementation as part of the link control logic.

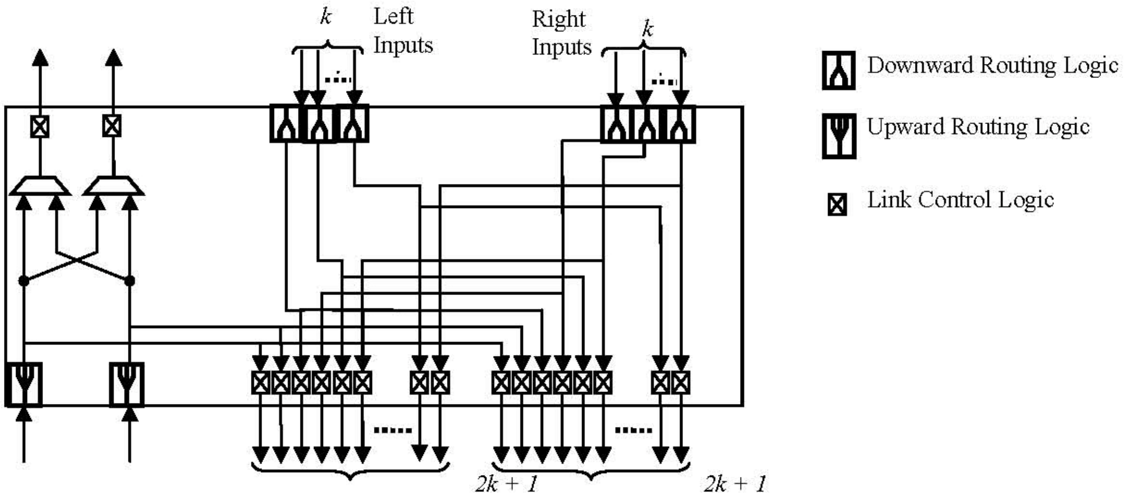

Figure 7. The adopted router architecture

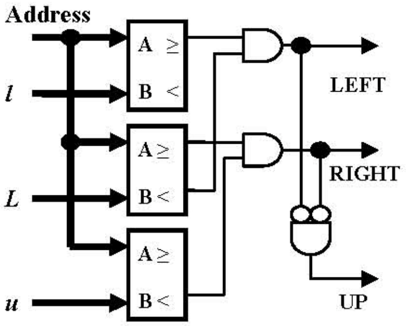

Figure 8. The routing function

3.6. Client Interface

In the proposed architecture, buffering is transferred from the routers to the client interfaces. Figure 9 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.

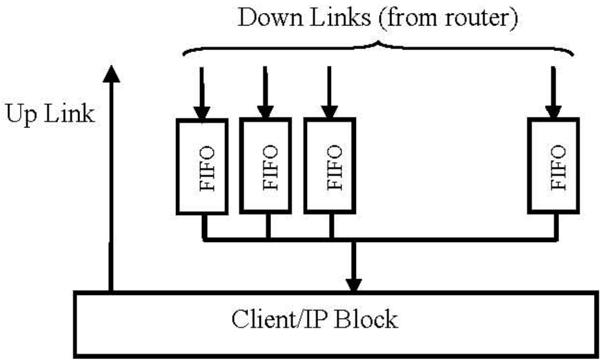

Figure 9. Client interface

4. 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 generation model follows a fixed rate with random destination addresses that follows a uniform distribution. It should be noted that this is by far a worse-than-average case scenario for packet routing: it basically assumes that the NoC designer did not take advantage of any communication locality between applications running on the clients (*i.e.* clients with maximum communications among them are not placed adjacently). It also means that 50% percent of all packets are reaching the highest routing level in the tree (maximum number of hops). Variable size packets were generated, with the packet size being randomly generated within a predetermined range. Every client will inject packets in the network at a fixed rate expressed as a fraction of the wire-speed link bandwidth. A rate of 90% means that a packet of 64 bytes is injected in the network every 64*1/0.9 ≈ 70 clock cycles. Because it takes only 64 clock cycles to send the packet, the remaining 6 cycles represent an inter-packet gap or idle time. Because of variability of packet size, there is no correlation between the generation dates of the different clients. On the receivers' side, at high injection rate, it can be stated that this method effectively simulates a continuous burst since several packets, injected from different sources, will be received continuously.

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.

4.1. Throughput and Latency

Simulations have shown, as illustrated in Figure 10 that the regular FT (throughput) quickly saturates to around 40 % of the maximum wire-speed bandwidth while the modified FT does not saturate at all. Figure 11 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 (labeled FT2 in the figures). The label suffix of 32C or 64C designates a number of 32 and 64 clients respectively. Two packet ranges have been considered: 64 and 128 bytes. The label suffix of 64ML and 128ML designates a packet size of 64 and 128 bytes respectively.

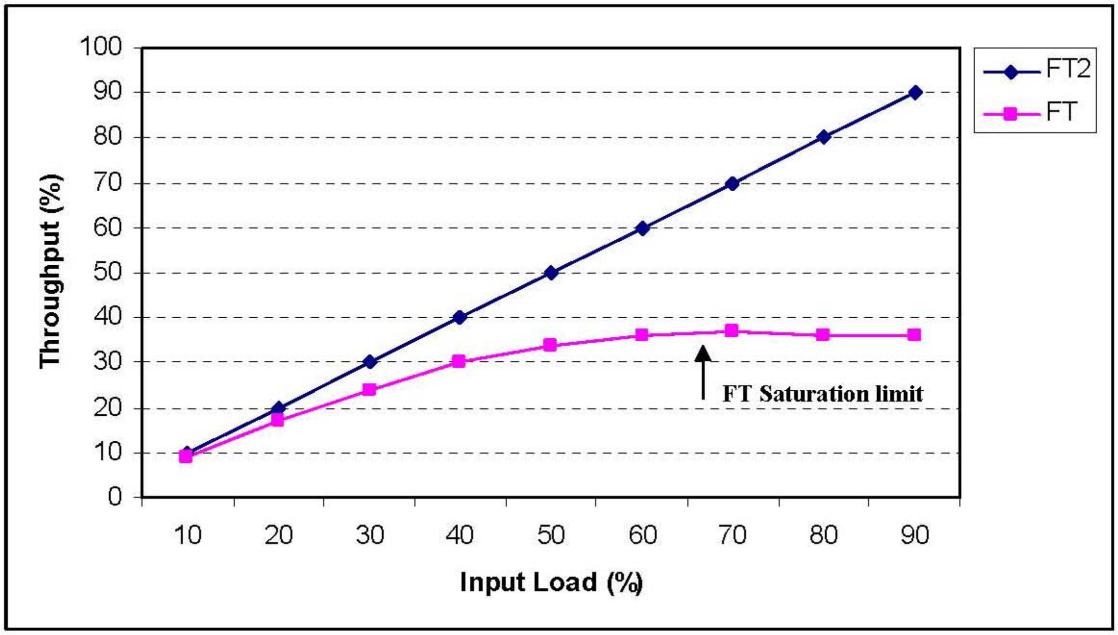

Figure 10. Throughput versus packet injection rate for regular Fat-Tree (FT) and the modified FT (FT2)

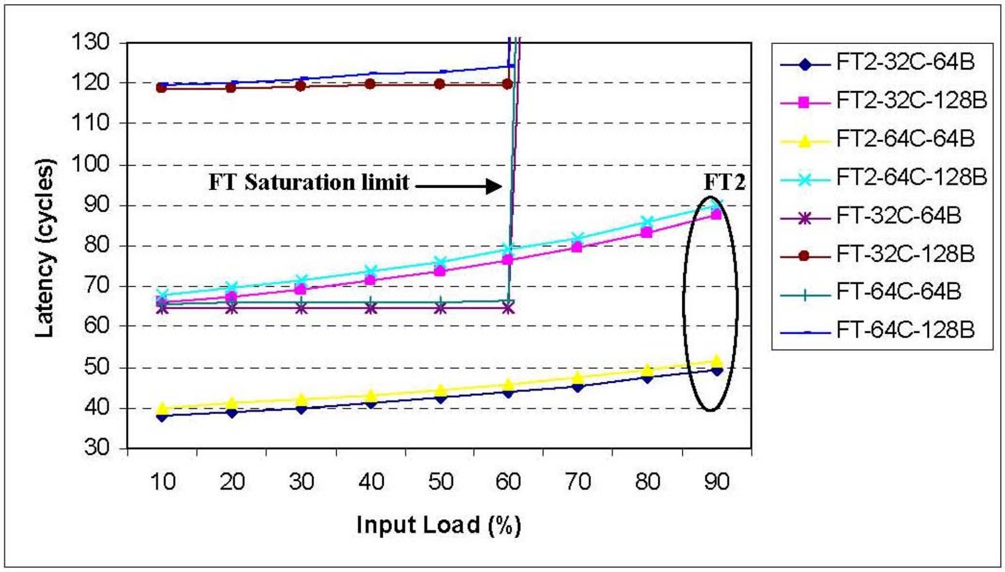

Figure 11. Latency versus packet injection rate for regular Fat-Tree (FT) and the modified FT (FT2). Combinations of two network sizes; 32 clients (32C) and 64 clients (64C) and two ranges of packet sizes; 64 bytes (64B) and 128 bytes (128B) were used. The actual packet size within the range was randomly generated (multiples of bytes)

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

4.2. Area Estimate

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. The area has been estimated from logic synthesis of the RTL model as the total gate area, *i.e.* with no place and route. The RTL design was synthesized using Synopsys and a custom-made 0.13 µm standard cell library. Then the area of each block was estimated as the sum of the areas of gates in that block. Though this is not very accurate, it gives an indication of the actual area if that block. A fully placed and routed implementation would require a target application with placed IP clients. Also, since routers have different number of down links depending on their level, the area per down link and area of uplinks per router were evaluated. These came to be 0.005 mm^2 and 0.02 mm^2, respectively. The uplinks (fixed number and fixed design for all routers) had significantly higher area because of the adaptive routing when going up. Hence the area of any router = number of down links * 0.005 mm^2 + 0.02 mm^2.

Table 1 gives the average gate count and size of the proposed router, for a 16 client network, compared to several published routers [5-7, 22]. The total size of all routers can be simply obtained by multiplying this average size by the total number of routers. As this table shows, the proposed router has a significantly smaller area (0.12 mm^2).

Routers in lower levels in NoCs with large number of clients would be larger (higher number of links); however they would also have an extremely high throughput. For example, 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 mm^2. However, it would achieve a maximum throughput of 409.6 Gbits/s with a clock of 800 MHz in a 0.13 µm process. This feature of increased performance with increased performance is not always achievable with other NoC architectures. For example, in a mesh NoC architecture, increasing the link size poses serious issues with the inter-node wiring and synchronization, and will not affect throughput.

It should also be noted that the larger the router, the closer it is to the client. Hence links in lower levels won't have to be routed for long distances. Only high level links, which are fewest, are routed for long distances.

Table 1. Router's Area Comparison With Several Published NoCs

Source	Num. Links	Gates	Tech.(µ)	Area (mm^2)
[6]	5	20 000	0.13	0.25
[7]	5	-	0.35	0.61
[21]	5	-	0.25	2.89
[5]	8	-	0.25	0.8
[15]	5	13 000	Lsi10k	-
FT2	8	10 100	0.13	0.12

4.3. FIFO Buffer Utilization

A tradeoff of this architecture is that buffers are pushed out of the routers to the client interfaces. Hence considerable amount of buffer lanes are required;. The obtained figures for active FIFOs (≤ 9) are about an order of magnitude lower than the number imposed 2^n *-1* FIFO lanes per client for a network of 2^n clients. Because of the network structure, any packet injected in the network will be received by one of the FIFO lanes in the destination interface. So the network accommodates the situation of *p* clients (*p*: [1 – (2^n–1)]) simultaneously and even continuously sending data to the same destination. In this case, at least *p* FIFO lanes will be active at the client interface. The only condition, which is imposed by the client themselves, not the NoC, is that the emptying rate of the FIFOs (F_{out}) should be at least equal to the filling rate (F_{in}); *i.e.* $F_{in} \leq F_{out}$. This means that the maximum burst frequency f_B, defined as the fraction of the maximum wire-speed bandwidth (which is a data unit per clock cycle), and *p* are limited by the FIFO emptying rate as:

$$p.f_B \leq F_{out}$$

Figure 12 below shows the maximum allowable burst frequency (f_B) versus the number of clients simultaneously sending data to the same destination (*p*). If this frequency is exceeded, FIFOs will be full and backpressure mechanisms will cause the sending client(s) to stop sending. The results are reported for several FIFO's emptying rates expressed in multiples of the filling rate, dubbed *OR* (output rate) in the Figure. A burst frequency of 100% means continuous data and an *OR* of 1 means that the FIFOs' emptying rate equals the filling rate of a single FIFO. As the figure shows, the limitations on the burst frequency and size are imposed by the emptying rate. For an *OR* equal to *p*, the burst frequency could reach 100%.

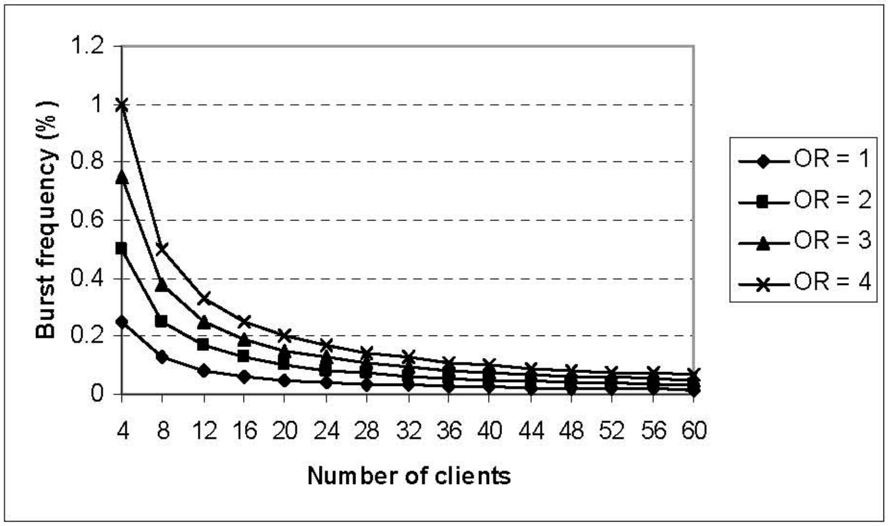

Figure 12. Maximum burst frequency as a function of the number of sending clients in the burst for several FIFO emptying rates

Simulations were carried to show how the actual number of FIFO lanes that are simultaneously used at anytime can be measured for different injection rates. A fixed rate traffic model with random destination address generation that follows a uniform distribution was used. A network of 64 clients, 63 FIFO lanes per client, and 2048 bytes per FIFO, with an output rate (*OR*) of 2, was simulated for different ranges of packet sizes. The actual packet size (within the range) and the start of packet injection into the network were also randomized. The results are reported in Figure 13 below. These results show that the used uniform distribution for the destination address and randomized packet sizes yielded a minimum of 4 clients sending to the same destination simultaneously (a *p* of 4) as evident from the maximum number of utilized lanes at very low injection rate. As the injection rate increases (thus increasing the burst frequency) *p* increases linearly. This is because as the burst frequency increases, the likelihood of more clients sending to the same destination increases too. The maximum number of FIFO lanes used (*i.e. p*) was 9. This is not an unlikely scenario for a heterogeneous SoC, where burst starts and message lengths cannot be 100% synchronized between different clients. Hence the number of buffer lanes in the client interface can be tailored, using the application's actual traffic models, for a specific platform to suit the class of applications at hand while reducing buffering area. The approach of hardware customization using application information is becoming a trend in the NoC area [23, 24]. Also, it should be noted that even if the designer underestimates the number of FIFO lanes required, the backpressure mechanism will guarantee that no packets are lost.

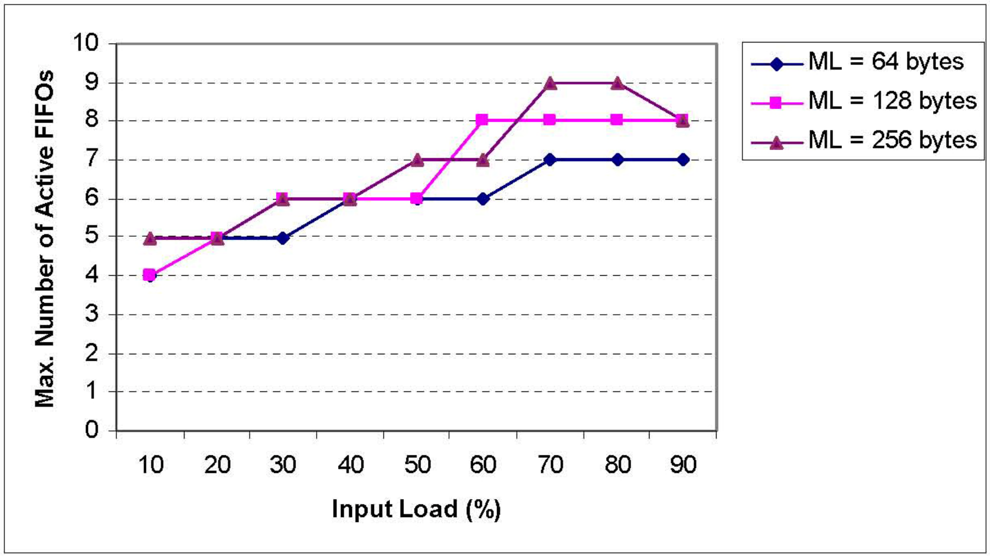

Figure 13. Maximum number of FIFO lanes simultaneously active per client versus injection rate for several packet sizes (ML)

5. CONCLUSIONS

A novel approach for efficiently dsigning NoC that takes full advantage of the on-chip context was introduced. The main strategy adopted for this approach was to make the number of output ports of each router twice the number of its input ports. This completely eliminates contention (and all the associated problems). As a result of this, the router architecture and routing function were significantly simplified with no need for buffering or routing tables. In each router, routing is done locally via a simple comparison between the destination address and constants. The sheer simplicity of the router compensates for the increased number of ports. Area estimates based on logic synthesis show that the average area/router is significantly lower than other published routers. Also, since the router architecture is very simple, it can be fully parameterized. Simulation results show a clear advantage over regular FT. The throughput completely matched the input load even for very high loads (simulated up to 99%) where other networks saturate.

ACKNOWLEDGMENT

This research project has been supported by King Fahd University of Petroleum and Minerals under project No. FT-2006/25. This support is highly appreciated by the authors.

REFERENCES

[1] L. Benini and G. D. Micheli, "Networks on Chips: A New SoC Paradigm", *IEEE Computer*, (**1**)(2002), pp. 70–78.

[2] W. J. Dally and B. Towles, "Route Packets, Not Wires: On Chip Interconnection Networks", in *Proceedings of the 38thDesign Automation Conference*, June 2001, pp. 684–689.

[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., November 2000*. Full text available on: http://web.it.kth.se/~axel/papers/2000/norchip-noc.pdf.

[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),* 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),* 2005, pp. 54–63.

[7] P. Zipf, H. Hinkelmann, A. Ashraf, and M. Glesner", A Switch Architecture and Signal Synchronization for GALS System-on-Chips", *Proc. SBCCI'04,* 2004, pp. 210–215.

[8] A. Radulescu, J. Dielissen, S. G. Pestana, O.P. Gangwal, E. Rijpkema, P. Wielage, and K. Goossens, "An Efficient On-Chip NI Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Configuration", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* **24**(**1**) 2005 pp. 4–17.

[9] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander, "Trade-Offs in the Design of a Router with Both Guaranteed and Best-Effort Services for Networks on Chip", *IEE Proceedings, Computer and Digital Techniques,* **150**(**5**)(2003), pp. 294–302.

[10] H. Zhao and Q. Wang, "Simulation and Evaluation for SS Network on Chip Architecture Using OPNET", *Proc. 8th International Solid-State and Integrated Circuit Technology (ICSICT'06)*, 2006, pp. 2098–2101.

[11] R. Holsmark, M. Palesi, and S. Kumar, "Deadlock Free Routing Algorithms for Mesh Topology NoC Systems with Regions", *Proc. 9th EUROMICRO Conference on Digital Systems Design, Architectures, Methods and Tools,* 2006, pp. 696–703.

[12] M. Zid, A. Zitouni, A. Baganne, and R. Tourki, "New Generic GALS NoC Architectures With Multiple QoS", *Proc. Intenational Conference on Design and Test of Integrated Systems in Nanoscale Technology,* 2006, pp. 345–349.

[13] S. K. Hasan, A. Landry, Y. Savaria, and M. Nekili, "Design Constraints of a HyperTransport-Compatible Network-On-Chip", *Proc. 2nd Annual IEEE Northeast Workshop on Circuits and Systems*, 2004, pp. 269–272.

[14] 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", *Proc. 21st International Conference on Computer Design (ICCD'03),* 2003, pp. 536–539.

[15] E. Nilsson "Design and Implementation of a Hot-Potato Switch in a Network on Chip", *Master's Thesis, Royal Institute of Technology*, *IMIT/LECS, Sweden*, June 2002.

[16] E. Nilsson, M. Millberg, J. Öberg, and A. Jantsch, "Load Distribution with Proximity Congestion Awareness in a NoC", *Proc. Design,Automation and Test in Europe Conference and Exhibition (DATE'03),* 2003, pp. 1126–1127.

[17] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing". *Proc. 19th Annual International Symposium on Computer Architecture*, 1992, pp. 278–287.

[18] E. Nilsson and J. Öberg, "Reducing Power and Latency in 2-D Mesh NoCs Using Globally Pseudochronous Locally Synchronous Clocking", *Proc. International Hardware/Software Codesign and System Synthesis (CODES+ISSS'04)*, 2004, pp. 176–181.

[19] E. Nilsson and J. Öberg, "Trading Off Power Versus Latency Using GPLS Clocking in 2D-Mesh NoCs", *Proc. International Symposium on Signals, Circuits and Systems, ISSCS*, 2005, vol. 1, pp. 51–54.

[20] K. Goossens, J. Dielissen, and A. Radulescu, "Æthereal Network on Chip: Concepts, Architectures, and Implementations", *IEEE Design and Test of Computers*, **22**(**5**)(2005), pp. 414–421.

[21] C. Leiserson, "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing", *IEEE Transactions on Computers*, **C-34**(**10**)(1985), pp. 892–901.

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

[23] P. Meloni, S. Murali, S. Carta, M. Camplani, L. Raffo, and G. De Micheli, "Routing Aware Switch Hardware Customization for Networks on Chips", *Proc. NanoNet'06,* 2006, pp. 1–5.

[24] B. Sethuraman and R. Vemur, "A Force-Directed Approach for Fast Generation of Efficient Multi-Port NoC Architectures", *Proc. 20th International Conference on VLSI Design,* 2007, pp. 419–426.