

Generic Low-Latency NoC Router Architecture for FPGA Computing Systems

Ye Lu, John McCanny, Sakir Sezer

The Institute of Electronic, Communication and Information Technology (ECIT)

Queen's University of Belfast, Northern Ireland, UK

Email: ylu10@qub.ac.uk

Abstract—A novel cost-effective and low-latency wormhole router for packet-switched NoC designs, tailored for FPGA, is presented. This has been designed to be scalable at system level to fully exploit the characteristics and constraints of FPGA based systems, rather than custom ASIC technology. A key feature is that it achieves a low packet propagation latency of only two cycles per hop including both router pipeline delay and link traversal delay - a significant enhancement over existing FPGA designs - whilst being very competitive in terms of performance and hardware complexity. It can also be configured in various network topologies including 1-D, 2-D, and 3-D. Detailed design-space exploration has been carried for a range of scaling parameters, with the results of various design trade-offs being presented and discussed. By taking advantage of abundant build-in reconfigurable logic and routing resources, we have been able to create a new scalable on-chip FPGA based router that exhibits high dimensionality and connectivity. The architecture proposed can be easily migrated across many FPGA families to provide flexible, robust and cost-effective NoC solutions suitable for the implementation of high-performance FPGA computing systems.

I. INTRODUCTION

A distinctive feature of FPGA (Field Programmable Gate Array) technology is that provides the means whereby hardware functionality can be electrically reconfigured following manufacturing [1]. This contrasts with other types of integrated circuits such as ASICs (Application Specific Integrated Circuits) where hardware tends to be fixed in function and deterministic in nature. This flexibility allows FPGAs to be configured to cover a very wide range of computationally intensive applications including signal processing, image processing, cryptology with designs typically achieving an order-of-magnitude performance advantage when compared with typical CPU based implementations [1].

As technology evolves, it is anticipated that new generations of FPGA devices will emerge that typically comprise millions of look-up tables (LUTS) and be populated with increasing numbers of soft processor cores operating in parallel in tandem with more general and configurable logic circuitry. As complexity increases, then the limitations of using traditional interconnection schemes become apparent, not least because of much more rigid resource constraints (i.e. the need to use pre-defined routing resources) when compared with on-chip router designs implemented using ASIC technology. In addition, the increasing complexity of such FPGAs makes traditional RTL based design flow inefficient. Increasingly, future emphasis will be on design and management of resources at higher

levels of abstraction i.e. at functional level, rather than at the detailed logic gate level. For example, it is anticipated that functionality will be implemented through the programmability of such cores operating in parallel rather than simply by the programmable wiring of on chip logic. This has also increasingly highlighted the limitations of traditional bus and point-to-point based interconnection schemes in terms of their scalability with on chip complexity.

Increasingly therefore interest is being given to the use of Networks-on-Chip (NoC) solutions for the provision of robust, flexible, and scalable communication services in these large systems [2], [3]. These allow bandwidth to be increased simply through the addition of new communication nodes (routers or switchers) and associated Processing Elements (PEs), with cost growing linearly with system size. This is in contrast with traditional connection-based interconnection schemes which, for example, scale exponentially when crossbar based interconnection schemes are used.

To date, both packet-switched and time-multiplexed NoC designs for FPGA have been investigated [4]. Time-multiplexed NoC schemes statically allocate network resource by pre-computing routing information off-line before run-time and because of this pre-characterized traffic patterns typically exhibit low network latency and a high utilization of network resources. Packet-switched NoCs negotiate network resources dynamically at run-time and are thus more flexible and more generally applicable, for example, by offering a robust service for non-deterministic communication patterns. Their main drawback tends to be higher latency.

An important characteristic of FPGAs when compared with ASICs is that the former inherently contain significant build-in local and global routing resources whether or not these are fully utilized. As has been highlighted in previous research, these resources can be exploited to create more complex, higher dimensional and higher connectivity network topologies than are typically used in ASIC design [5], [6], [7]. The research presented in this paper has investigated these issues in detail and presents a novel low-cost and low-latency router architecture that is applicable to a wide range of FPGA families. The architecture presented is generic and scalable in that it can be used in a multitude of configurations and configured in different network topologies. This allows application specific customizations of the NoC, taking advantage of the FPGA's programmable interconnect fabric.

Whilst investigations have previously been undertaken on high dimensional and high connective NoC topologies for FPGA design [5], [6], [7], to date these have been based on specific experimental configurations that use simple network interfaces that neglect the logic cost of practical routing systems. In practice, such implementations are heavily influenced by these characteristics as these play a vital role in the feasibility and performance of actual NoC designs and implementations. An important aspect of our research, based on the generic approach described, has been to investigate this in much more detail than before by implementing and undertaking a detailed analysis of the cost, timing and performance trade-offs for practical FPGA NoC routers implemented using a range of design parameters. As will be discussed, these implementations are based on a new cost-effective wormhole router architecture that has been optimized for FPGA technology and achieves a low zero-load latency of only two cycles per hop including both router pipeline delay and link traversal delay whilst maintaining a short critical path and hence speed performance.

The feasibility of the approach is demonstrated by applying this in different network topologies including a 1-D ring, a 2-D mesh, a 3-D Cube with the results of the analysis we have undertaken in terms of resource utilization, timing, and power efficiency for these various configurations and topologies presented in the paper. This includes detailed Static Timing Analysis (STA) and power analysis using post layout synthesis and layout simulation results. This we believe represents a significant enhancement compared with state-of-the-art NoC FPGA designs not least because, to the best of our knowledge, no work has been published on practical high-radix NoC routers for FPGA that exhibit high dimensionality and high connectivity. The approach presented then allows important design trade-offs and decisions for such routers to be rapidly made that are based on actual FPGA hardware design and implementation.

II. RELATED WORK

Previous research on NoCs suitable for FPGA include the work of [8], who studied a time-multiplexed router, [9], who presented a packet-switched router, and [10], who focused on the circuit-switched router. Typically a circuit-switched NoC has the drawbacks of long circuit setup latency and low link bandwidth utilization. However, as has been discussed by [10] and [11], once the circuit is set up, circuit-switched NoCs on FPGA do have the advantages of guaranteed QoS and low data transmit latency. A circuit-switched network can also be used together with a packet-switched network to combine the advantages of both [12], [13]. A time-multiplexed NoC can take advantage of a pre-characterized communication pattern to achieve low resource utilization and reduced latency. Kapre et al. [4] presented a comparison between a time-multiplexed NoC and a packet-switched NoC with a Butterfly Fat Tree (BFTs) topology on FPGA. Their results indicate that the time-multiplexed NoC outperforms the packet-switched NoC when communication loads are 100% and pre-characterized

but drops below the performance of a time-multiplexed NoC when communication loads drop below 40%. Accurate pre-characterization of communication patterns is usually less complex in DSP applications that have a regular and pre-determined data-flows, but impossible in more general purpose and non-deterministic applications such as network processing. It is well recognized that packet-switched NoCs are the most likely to offer robust and scalable communication services for a wide range of applications [2]. However, the main drawback of a packet-switched NoC is its latency as the routing and negotiation of resources must be calculated at run-time. Sethuraman et al. [14] have recently presented research on packet-switched NoCs suitable for FPGA, where a lightweight store-and-forward router has been proposed. This router has the advantage of low resource utilization but introduces high latency (the minimum latency is 8 cycles per hop) due to the nature of its store-and-forward switching functions. They later extended their proposed architecture to a multi-local port router and introduced an algorithm for automatic mapping task graphs onto their proposed NoC [15], [16]. This has been further improved by [17] who proposed a virtual-cut-through router, capable of achieving an impressive 357 MHz clock rate using Virtex-4 FPGA technology, introducing a 7 cycle latency per hop. FPGA is also widely used for NoC performance emulation, such as that presented in [18], [19], [20]. Rather than final implementation, these works have used FPGA technology as a prototyping platform with the purpose of predicting latency, throughput, cost and power dissipation of NoC designs before implementing these on ASIC. As summarised above, related researches have been targeting mainly predictable performance of NoC architectures for ASIC implementation using FPGA technology as a prototyping platform. Proposed solutions have many shortcomings if applied for FPGA technology, including large hop-by-hop latencies and high resources usage. The motivation of this research presented in this paper is to derive an optimized NoC architecture for FPGA technology that takes advantage of FPGA specific properties and features, such as programmable routing and logic resources, for optimizing link and routing fabric, topology and minimizing propagation latency.

III. GENERIC LOW LATENCY ROUTER ARCHITECTURE

For the proposed NoC architecture a reconfigurable wormhole router architecture is used, the details of which are presented below. This was chosen over other options (e.g. store-and-forward, virtual-cut-through) because of its low routing latency, low complexity and high buffer utilization. Two key attributes that have allowed this to be achieved are 1) high scalability, mainly in router radix in order to accommodate highly connected topologies, and 2) optimized pipeline organization in order to reduce hop delay.

A. Components

A basic block diagram of this is shown in Figure 1(a). The router mainly comprises input ports, switch arbiters, Finite

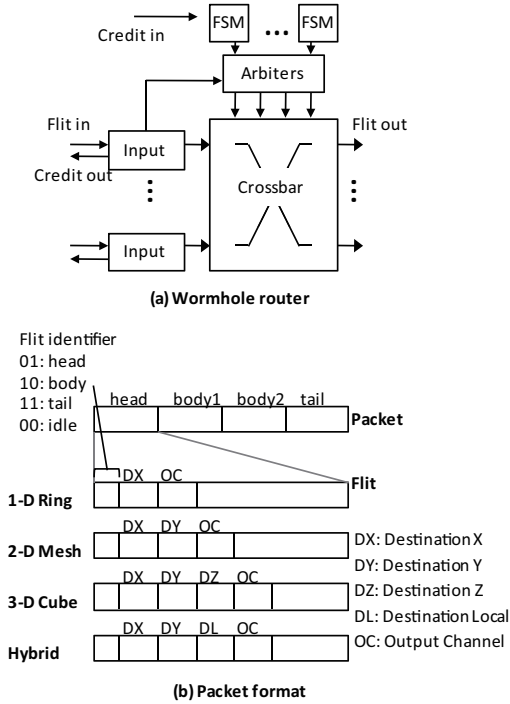


Fig. 1. Wormhole router block diagram and packet format

State Machines (FSM) and the crossbar. The packet format used is shown in Figure 1(b). This comprises a head flit, a body flit and a tail flit. Here the packet length is unfixed and longer packets can be formed by adding more data flits between the head and the tail. The format of the head flit is determined by the network topology as shown in Figure 1(b). The Output Channel (OC) field stores the output channel used by the packet. This is pre-computed using look-ahead routing, one hop ahead. The width of the address field and output channel field both vary with network size. The router is fully pipelined allowing flits to pass through it in this manner.

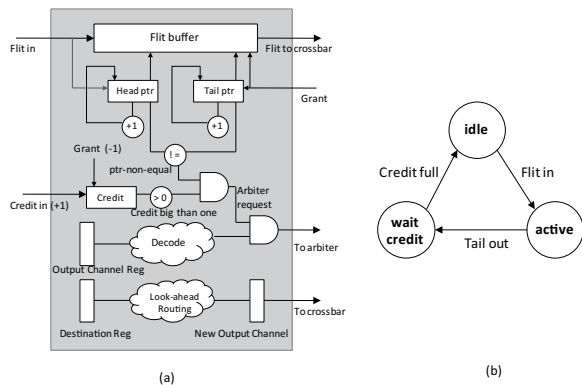


Fig. 2. (a) Input port block diagram, and (b) State diagram of output FSM

Input port. Figure 2 (a) shows the block diagram of the input port architecture. Here a single entry flit buffer memory has

been used and implemented using dual-port block memory. The look-ahead routing module performs Dimensional Ordering Routing (DOR) one hop ahead, decoupling the routing computation and switch arbitration functions. The routing result obtained is latched into the new output channel register and is then used by the downstream router. Two registers are used to store the head pointer and the tail pointer, indicating the beginning and the end respectively of the flits buffered in the memory. Once a new flit has been received, the head pointer increments by one. Also once the switch granted signal is set, the tail pointer also increments by one. The head pointer and the tail pointer are then compared to determine whether or not there is a flit in the buffer waiting to be transmitted. The router uses credit based flow control. Here the credit register records the count of the free buffer available in the downstream router and a flit can be only transmitted when the credit count is greater than one. The switch arbiter request signal is then set when there is flit in the buffer waiting to be transmitted and a free buffer is available in the downstream router.

Switch arbiter. A switch arbiter is located in each output port and arbitrates between multiple requests from input ports. The static priority arbiter, such as the one used in [12], can save logic resources, but has a serious fairness problem which adversely impacts network throughput. We have therefore used a ripple arbiter [21] and a matrix arbiter [22] to provide *round-robin* and *least recently served* schemes for switch arbitration. The arbiter type used is chosen using the router configuration file.

Output port and FSM. The Finite State Machine (FSM) maintains the state of the output port. The flit buffer used in the router has only a single entry, exclusively allowing a single packet to be held. Figure 2 (b) shows the state diagram for the output port FSM. The active state indicates that the output port has been associated with an input port and preventing the request from other input port being propagated to the switch arbiter. When the tail flit of the packet leaves the local buffer, the FSM assumes a wait credit state. In this state the output port waits for the credit request to come back. When the credit is full, indicating all flits have left the downstream router, the output port can be reallocated to deal with other requests, at which point the FSM goes into the idle state.

B. Pipeline organization and router hop delay

Figure 3 (a) shows the design of each pipeline stage in the proposed router. In this the destination address and the output channel of the packet are latched on the rising edge of the first clock cycle. Meanwhile, the flit is written into the flit buffer memory. During the first cycle, the register of the output port is decoded and the arbitration result is returned. The look-ahead routing logic computes the next output port to be used by the downstream router. The crossbar control signal is latched on the rising edge of the second clock cycle and the flit associated with the granted port read from the buffer. The flit then travels through the crossbar on the second clock cycle.

On an FPGA, the crossbar is implemented using multiplexers based on LUT logics. A 4 input multiplexer can be

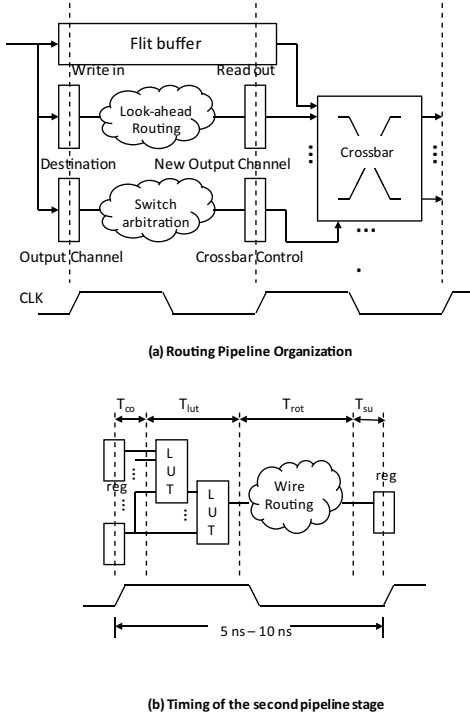


Fig. 3. (a) Router pipeline organization and (b) timing of crossbar stage

implemented using 6 input LUT, 2 LUT inputs used by the multiplexer control, and a 4 LUT inputs used for multiplexer input. For a router with 5 - 8 physical channels, two LUTs are thus needed for each bit in the output channel.

Figure 3 (b) shows the timing diagram on the crossbar (multiplexer) pipeline stage, which can be expressed by Equation (1).

$$T = T_{co} + T_{lut} + T_{rot} + T_{su} \quad (1)$$

In this equation, T_{co} is clock to register (or memory) output delay, T_{lut} is the delay of LUT cells (multiplexer logic), T_{rot} is the delay due to programmable wire routing and T_{su} is the setup time of the device. Detailed Static Timing Analysis (STA) indicates that an LUT delay is around 0.2 ns on a 65-nm FPGA device (Stratix III family from ALTERA). Typically an FPGA operates at 100 MHz - 200 MHz, giving a 5 ns - 10 ns timing budget on the path. Two LUTs contribute roughly 4%-8% to the total delay due to the very simple multiplexer logic, leaving most of the timing budget on the path for the programmable wire routing. Therefore, separating the multiplexer logic delay and the link wire routing delay into two pipeline stages, as is done in ASIC based approaches, will increase the network delay but cannot improve speed performance. We therefore combine the crossbar traverse and link traverse together within the same stage.

Figure 4 shows the pipeline diagram of the proposed router architecture and contrasts this with a corresponding diagram for typical ASIC based, low-latency router architecture [23].

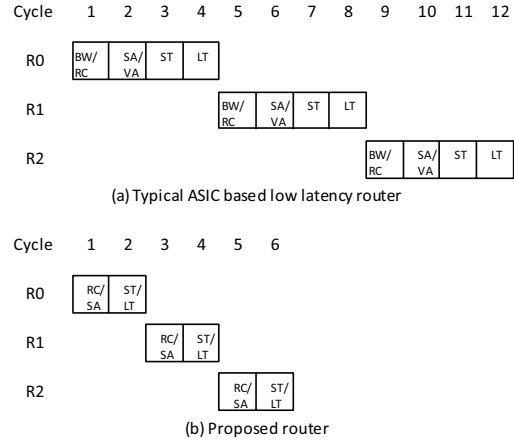


Fig. 4. Pipeline diagram of (a) a typical ASIC based low latency router, and (b) the proposed router for FPGA

The pipeline stages of ASIC router are utilized as (a) buffer write and route computing (BW/RC), (b) switch allocation and VC allocation (SA/VA), (c) switch traverse (ST), and (d) link traverse (LT). Due to the modest clock rate that an FPGA operates at, the buffer read and write can be performed in parallel with a control setting and no separate pipeline stage are required for dedicated buffer write or read as used in an ASIC approach. Here the route computation (RC) and switch arbitration (SA) are computed in parallel within the first cycle. The switch traverse (ST) time and link traverse (LT) time combined within the second cycle as implied above. With this approach a two cycle hop delay, including router delay and wire delay, is achieved. Detailed timing analysis results for the critical path with using STA are presented in the following section.

C. Radix and flow control

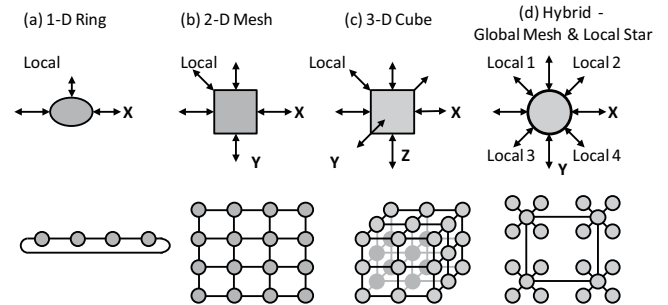


Fig. 5. Typical NoC topologies and routers radices

Four different and typical categories of network topologies have been investigated as shown in Figure 5. These are classified by their radix (or degree) with this defining the number of the input and output ports used in the router, e.g., a radix 5 router uses 5 input ports and 5 output ports [24]. As can be seen from the figure, the 1-D Ring topology uses a router with a radix of 3. In this two ports connect to

two neighbouring routers, whilst the third one connects to its local Processing Element (PE). The 2-D Mesh and 3-D Cube topologies use radix 5 and radix 7 routers respectively. As before port 1 connects to the local PE with the rest connecting to neighbouring routers. The hybrid (global mesh and local star) topology uses a radix 8 router with 4 ports connected to neighbouring routers and 4 ports connected to local PEs. The attraction of high-radix routers with high levels of connectivity is that hop numbers can be significantly reduced with the network diameter. For example, to forward a packet from the top left corner to the bottom right corner in a 4×4 2-D mesh network requires 7 hops but only 3 hops in the hybrid global mesh and local star network. Of course, the performance of a network is also affected by a number of other factors, as is discussed.

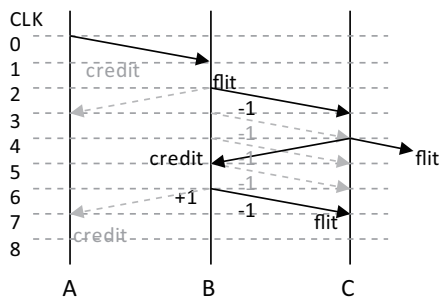


Fig. 6. Credit round-trip

As discussed above a credit based flow control is used in the proposed router. With this a credit is returned to the upstream router once a flit leaves the local buffer. Full utilization of the link bandwidth requires sufficient flit buffer space to be allocated [25]. Figure 6 shows the credit round-trip sequence used in the proposed router. Here a flit leaves the buffer of the router A at CLK 0 and arrives at router B on CLK 1. After travelling through router B and the link between routers B and C, it arrives at the router C on CLK 3 whilst on CLK 4 it leaves the buffer of the router C. Meanwhile, a credit is sent back to the router B, arriving then at CLK 5. After one cycle of processing time the credit register within router B increments by one. If the flit buffer only has sufficient space to store a single flit, then no transaction will be permitted between CLK 2 and CLK 6 as the credit value is zero during these cycles. Thus, the link bandwidth will not be utilized. However, this issue can be overcome and full utilization achieved by setting the minimal initial value of the credit register to five. Thus, the buffer depth of the proposed router must be set to four flits to ensure effective use of the link bandwidth.

IV. DESIGN STUDIES AND ANALYSIS

The generic architecture described has been captured at RTL level using the VERILOG Hardware Description Language and synthesized this using commercial EDA tools. An overview of the experimental flow and the design tools used is summarized below, Figure 7.

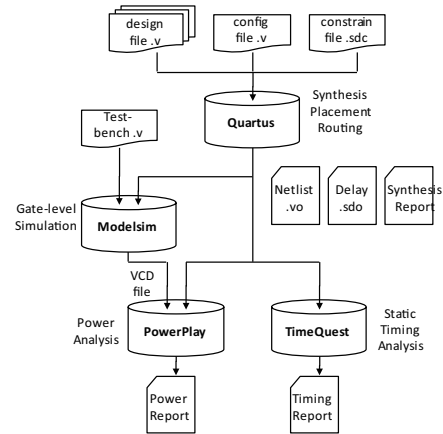


Fig. 7. Experiment flow with commercial CAD tools

Each investigated variance of the proposed wormhole router architecture has been implemented via the configuration files specifying radix used, flit format, data-path width, flit buffer depth, etc. Synopsys Design Constraints (SDC) have been used in each of the designs undertaken. Front end and back end design flow have been implemented using Quartus, creating the netlist and delay files. MODELSIM has then been used for post synthesis and layout simulation. The test-bench is used to generate the packets propagated to each router and the Value Change Dump (VCD) file used to record all the internal netlist signals. The VCD file was then used to undertake power analysis using the PowerPlay tool. The TimeQuest timing analyzer was used for Static Timing Analysis (STA) and used to determine the value of each critical path.

A. Resource utilization

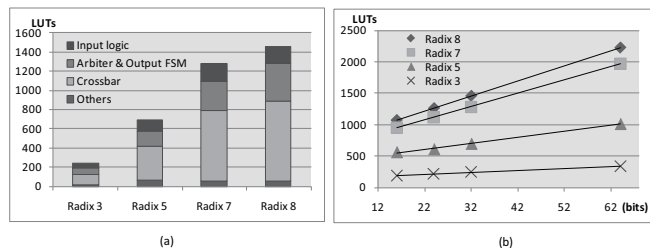


Fig. 8. (a) Logic cost vs. Router radix (32 bits data-path width), and (b) logic cost vs. Data-path width

Logic cost vs. radix. With an Altera Stratix III device chosen as the target technology, Figure 8 (a) shows how logic (LUTs) cost of various routers varies with router radix in this section. As illustrated, the different gray scales used indicate the relative logic requirements of the various router components. The results obtained show that the total logic cost of a router increase significantly with the radix number. These increased costs reflect three major factors. Firstly, the crossbar logic dominates router cost, typically representing between 45% - 60% of the overall router logic, with crossbar complexity

increasing exponentially with the radix, making efficient LUT mapping challenging. Secondly, as the number of the input and output ports increases with the radix, the corresponding control logic needed for these increases correspondingly. Thirdly, high radix routers require larger size arbiters with this then increasing the complexity and cost of switch arbitration.

Logic cost vs. data-path width. Increase the channel bandwidth can reduce the average latency due to the reduced packet serialization delay. A detailed analysis was also undertaken of variations in logic cost with data-path width and results obtained to for 16, 24, 32, and 64 bit wide data-paths. These are summarized in Figure 8 (b). It should be noted that 16 and 24 bits data widths tend to be widely adopted in DSP based systems, while 32 and 64 bits data widths are typically preferred in network processing and general purpose computing systems. The result imply that, in the main, logic cost grows linearly with the bit count of the data-path width, as might be expected, since increases in data-path width act only to increase the buffer memory and crossbar size, with the remaining circuitry remaining the same.

B. Timing

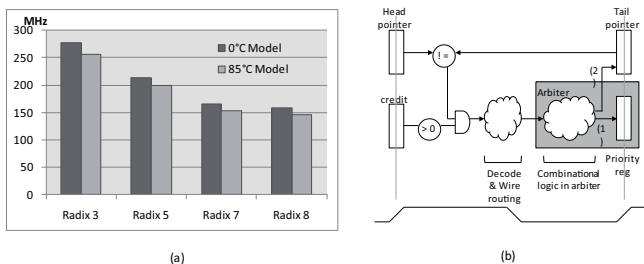


Fig. 9. (a) Maximum clock rate vs. Router radix, and (b) two important critical paths within a router

Detailed analyses have also been undertaken using Static Timing Analysis in order to investigate how maximum clock rates and critical paths vary with design choices. The results obtained are summarized in Figure 9 (a). These results also show variations in operating temperature impact delay.

The results clearly show that the maximum clock rate achievable drops by around 44% as the radix increases from 3 to 8 (in this case from 278 MHz for radix 3 and 158 MHz for radix 8 at 0°C and 256 MHz for radix 3 and 147 MHz for radix 8 at 85°C). Exact figures for any specific design obviously depend on the details of placement and routing. However, these wider variations can be understood by considering the two important critical paths highlighted in Figure 9 (b).

The first of these is from the output of the head pointer register to the input of the priority register within the arbiter. The second is from the output of the head pointer to the enable port of the tail pointer register. The combinational logic delay in both these paths is strongly impacted by the combinational logic delay of the arbiter. As discussed previously, higher radix routers employ larger size arbiters with longer combinational logic delays. Another key factor is the delay due to decoding

and wire routing. The higher the radix the more complex the wire routing and thus it is inevitable that these two factors combine leading to the trends highlighted in Figure 9 (b).

C. Power

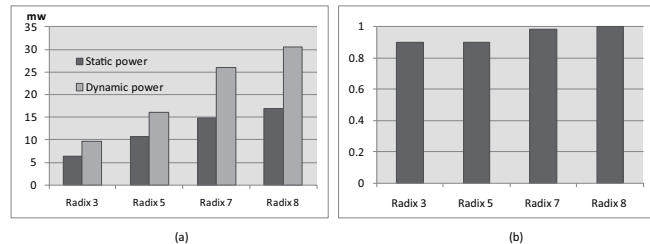


Fig. 10. Power of the proposed router. (a) Static and dynamic power (mw) vs. Router radix, (b) normalized per-packet power consumption vs. radix

Power dissipation is increasingly a key issue in complex chip designs. We have therefore also undertaken a detailed analysis of the power characteristics of the various routers investigated. These are based on the post synthesis and layout simulation results rather than simply using default toggle rates. For the purposes of this work the clock rate was set to 100 MHz for all routers and for simulations each port was injected with 32 bit wide randomly addressed packet data, 4 flits in length, with an injecting rate of 10% and with the VCD file recording the toggle rate of all internal signals.

Static and dynamic power. Figure 10 (a) show the results obtained for static power and the dynamic power for each router versus router radix. As can be observed, static power increases noticeably with radix for example from around 6.4 mw for the radix 3 based router to around 16 mw for the radix 8. Dynamic power also grows, for example from around 9.6 mw for a radix 3 router to around 30.6 mw for a radix 8 router. This increased power is due to the larger resource requirements and utilization of higher radix routers.

Power efficiency. The advantage of higher radix routers is that they are able to processes more packets than low radix ones within the same time frame. Thus to provide a fairer comparison we have also derived normalized values to ascertain the power consumption per packet for each of the router configurations. These results are shown in Figure 10 (b) where the values shown comprise combined static power and dynamic power.

The results obtained suggest that low radix routers are more power efficient than high radix router designs, mainly due to the simplicity their implementation. However, the variations across all of these configurations is not significant being less than 10% between the best (radix 3) and the worst cases (radix 8).

V. NETWORK IMPLEMENTATION

In this section, we verify the feasibility of the proposed router in an actual on-board design environment. This has been done by implementing the NoC network configurations as circuits on an FPGA evaluation board (DE-3 EP3SE340

TABLE I
RESOURCE UTILIZATION OF DIFFERENT NETWORK CONFIGURATIONS

Size	Node count	Topology	LUTs	Registers	Memories	Overall cost
Small	< 16	1-D ring or 2-D mesh	< 4%	< 2%	< 1%	Very low
Medium	64	2-D mesh	16%	7%	< 1%	Low
Medium	64	3-D cube (4-layers)	24%	9%	< 1%	Medium
Large	128	3-D cube (2-layers)	47%	18%	1%	High

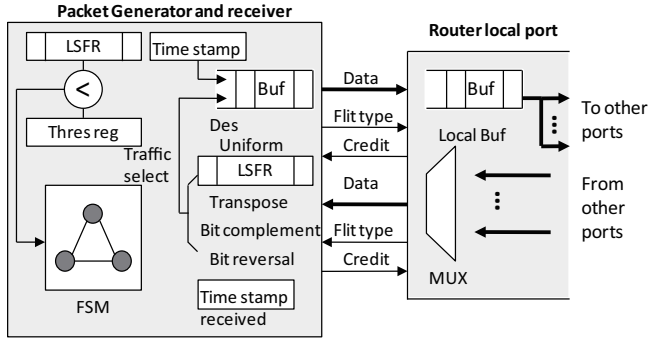


Fig. 11. Packet generator and receiver.

from Altera). Due to pin limitations, packets must be generated using on chip logic within the FPGA rather than external sources. Each node of the NoC system is attached with a packet generator and receiver.

The packet generator and receiver are connected to the local port of the router with three signals, namely data, flit type and credit for each direction (admission and ejection), as shown in Figure 11. In these studies the packet generation process was controlled by a Finite State Machine (FSM) and a pseudo random number generator implemented using maximum length Linear Feedback Shift Registers (LFSR). The FSM maintains the state of the local buffer and decides whether a packet from the queue can be admitted to the network. Also, the FSM acts to interfere with the packet injecting process. For example, if the local buffer does not have enough credit for new flit, because of network contention, the FSM will stop the packet injection process until the buffer space has been evacuated. The destination address is selected from the four traffic pattern candidates, namely uniform, transpose, bit complement, and bit reversal.

The packet receiver used is a simple logic circuit that returns credit back to the router after a flit has been received. The receiver is also responsible for collecting the time stamp carried by the received packet and for calculating the packet delay by comparing the time stamp to the current time.

Table I lists the different resource utilization (LUT, register, and memory) as a percentage of each network configuration and Figure 12 further illustrates this graphically (total available resource on EP3SE340 ALUTs: 270400, Registers: 270400, Memory bits: 16662528). This also includes a modest cost for the packet generator and receiver attached on each node.

This table and the figure indicates that the cost of implementation on a small sized NoC is very low and that less than

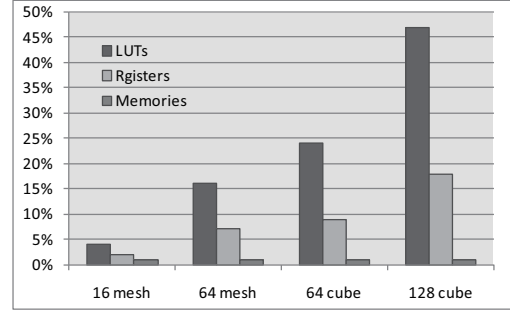


Fig. 12. Resource utilization of different network configurations

4% LUTs, 2% Registers, and 1% memories on this 65-nm FPGA are required. NoCs at this scale can therefore be simply be deemed to be glue logic since they will not significantly impact the total system resource requirements.

As the size of a network increases, the cost increases significantly. An NoC comprising 64 nodes consumes around 16%-24% of LUTs and 7%-9% of registers but less than 1% of on-chip FPGA memory with actual values dependent on the network topology used. With the same network size (64 nodes) a 4 layer 3-D cube requires an approximate increase of around 33% in terms of LUTs required and a 22% increase in the number of registers compared with a 2-D mesh topology. This increased overhead associated with the high radix 3D router is a consequence of the crossbar and arbitration logic exhibit an increasing as $O(n^2)$. However, when measured against total FPGA resources available on the FPGA device investigated the overall overhead introduced by using a 3-D topology still remains very small (8% of LUTs and 2% of registers).

When we increase the size of the NoC to 128 nodes with a 2-layer 3-D topology, the cost increases to near a half (47%) of total LUTs available. FPGA vendors have recently announced their 28-nm products while this work is based on an off-the-shelf evaluation board based in with 65-nm technology. Thus with such designs it is expected that this overhead will be reduced to about one quarter of that on the device we have used for investigation i.e. the latest Altera FPGA's have over 1,000,000 equivalent LUTs [26] as opposed to the 270,400 LUTs on the FPGA used for these evaluations.

VI. CONCLUSION

The research presented in this paper has resulted in a novel low-latency NoC router design tailored for FPGA technology. This has been designed to be scalable at the system level and to fully exploit the characteristics and constraints of FPGA based

systems, rather than custom ASIC solutions, where much of the research undertaken to date has been focused.

The proposed architecture exhibits the main attractions of packet-switched NoC systems, while addressing the problem of hop-by-hop propagation latency. Each pipeline stage is optimized as such that the zero-load packet propagation latency of the proposed NoC is only two cycles per hop including the router pipeline and link traversal. This we believe represents a significant enhancement over state-of-the-art FPGA designs.

Key contributions include (a) the definition of a highly scalable router architecture capable of supporting various network topologies on FPGA (1D, 2D and 3D) (b) the architectural optimization of the router such that two cycles per hop can be achieved, (c) a detailed analysis of the proposed architecture in terms of scalability, hardware cost (area), operation speed (critical path) and power dissipation and (d) demonstrating the feasibility of the proposed router in an real-world on-board design environment. By taking advantage of the abundant reconfigurable logic and routing resources on modern FPGA we have been able to derive a scalable on-chip router architecture suitable for FPGA that exhibits high dimensionality and high connectivity that enables flexible, robust and cost-effective NoC solutions to be rapidly designed that are well suited to high-performance FPGA computing systems.

REFERENCES

- [1] M. Lanzagorta, S. Bique, and R. Rosenberg, *Introduction to Reconfigurable Supercomputing*. Morgan & Claypool Publishers, Jan. 2009, vol. 4, no. 1.
- [2] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Computing Surveys (CSUR)*, vol. 38, no. 1, pp. 1–51, Jun. 2006.
- [3] R. Marculescu, U. Ogras, L. Peh, N. Jerger, and Y. Hoskote, "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 1, pp. 3–21, Jan. 2009.
- [4] N. Kapre, N. Mehta, M. DeLorimier, R. Rubin, H. Barnor, M. Wilson, M. Wrighton, and A. DeHon, "Packet Switched vs. Time Multiplexed FPGA Overlay Networks," in *2006 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*. Ieee, Apr. 2006, pp. 205–216.
- [5] J. Lee and L. Shannon, "Predicting the performance of application-specific NoCs implemented on FPGAs," in *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays - FPGA '10*. New York, New York, USA: ACM Press, 2010, p. 23.
- [6] M. Saldana, L. Shannon, J. Craig, and P. Chow, "Routability of Network Topologies in FPGAs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 8, pp. 948–951, Aug. 2007.
- [7] M. Saldaña, L. Shannon, and P. Chow, "The routability of multiprocessor network topologies in FPGAs," in *Proceedings of the 2006 international workshop on System-level interconnect prediction*. New York, New York, USA: ACM, 2006, pp. 49–56.
- [8] N. Mehta, "Time-multiplexed FPGA overlay networks on chip," Ph.D. dissertation, 2006.
- [9] N. Kapre, "Packet-switched on-chip FPGA overlay networks," Ph.D. dissertation, 2006.
- [10] C. Hilton and B. Nelson, "PNOC: a flexible circuit-switched NoC for FPGA-based systems," *Computers and Digital Techniques, IEE Proceedings-*, vol. 153, no. 3, pp. 181–188, 2006.
- [11] L. Moller, P. Fischer, F. Moraes, L. S. Indrusiak, and M. Glesner, "Improving QoS of Multi-layer Networks-on-Chip with Partial and Dynamic Reconfiguration of Routers," in *2010 International Conference on Field Programmable Logic and Applications*. Ieee, Aug. 2010, pp. 229–233.
- [12] H. Giefers and M. Platzner, "A Triple Hybrid Interconnect for Many-Cores: Reconfigurable Mesh, NoC and Barrier," in *2010 International Conference on Field Programmable Logic and Applications*. Ieee, Aug. 2010, pp. 223–228.
- [13] J. Duato, P. Lopez, F. Silla, and S. Yalamanchili, "A high performance router architecture for interconnection networks," *Proceedings of the 1996 ICCP Workshop on Challenges for Parallel Processing*, pp. 61–68, 1996.
- [14] B. Sethuraman, P. Bhattacharya, J. Khan, and R. Vemuri, "LiPaR: A light-weight parallel router for FPGA-based networks-on-chip," in *Proceedings of the 15th ACM Great Lakes symposium on VLSI*. ACM, 2005, pp. 452–457.
- [15] B. Sethuraman and R. Vemuri, "Multi2 Router: A Novel Multi Local Port Router Architecture With Broadcast Facility For FPGA-Based Networks-On-Chip," in *Field Programmable Logic and Applications, 2006. FPL'06. International Conference on*. IEEE, 2006, pp. 1–4.
- [16] —, "A Force-directed Approach for Fast Generation of Efficient Multi-Port NoC Architectures," in *20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems (VLSID'07)*. Ieee, Jan. 2007, pp. 419–426.
- [17] A. Janarthanan, V. Swaminathan, and K. Tomko, "MoCRoS: an Area-Efficient Multi-Clock On-Chip Network for Reconfigurable Systems," *IEEE Computer Society Annual Symposium on VLSI*, pp. 0–1, 2007.
- [18] N. Genko, D. Atienza, G. De Micheli, J. Mendias, R. Hermida, and F. Catthoor, "A Complete Network-On-Chip Emulation Framework," in *Design, Automation and Test in Europe*. Ieee, 2005, pp. 246–251.
- [19] H. G. Lee and D. Marculescu, "Challenges and promising results in NoC prototyping using FPGAs," *Ieee Micro*, pp. 86–95, 2007.
- [20] N. Genko, D. Atienza, G. De Micheli, and L. Benini, "Feature-NoC emulation: a tool and design flow for MPSoC," *Circuits and Systems Magazine, IEEE*, vol. 7, no. 4, pp. 42–51, 2007.
- [21] P. Gupta and N. McKeown, "Designing and implementing a fast crossbar scheduler," *IEEE Micro*, vol. 19, no. 1, pp. 20–28, 1999.
- [22] L.-S. Peh and W. Dally, "A delay model and speculative architecture for pipelined routers," in *Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture*. IEEE Comput. Soc, 2001, pp. 255–266.
- [23] A. Kumar, L.-S. Peh, and N. K. Jha, "Token flow control," *2008 41st IEEE/ACM International Symposium on Microarchitecture*, pp. 342–353, Nov. 2008.
- [24] J. Kim, W. J. Dally, B. Towles, and A. K. Gupta, "Microarchitecture of a High-Radix Router," in *32nd annual international symposium on Computer Architecture*, vol. 33, no. 2. IEEE Computer Society, May 2005, pp. 420–431.
- [25] W. Dally and B. Towles, *Principles and practices of interconnection networks*. Morgan Kaufmann, 2004.
- [26] Altera, *Stratix V Device Family Overview*, 2010.