I J C T A, 9(13) 2016, pp. 6267-6276
© International Science Press

Design of High Speed Data Efficiency Programmable on-chip Permutation Network for MPSOC

Ratnakaram Jyothi and V. Sarada

ABSTRACT

Multiprocessor system on chip which is a most developed architecture with different features. One of the most important is networking. Networking which required high speed for the traffic permutation in which each input sends traffic to each output by using pipelined circuit switching network which is better compared with other on-chip network. The pipelined circuit switching offers guarantee of permuted data with dynamic path setup scheme for selecting arbitrary schemes. Another most important aspect is Low Power Analysis which mainly reduced the leakage and optimists the dynamic power.

Index Terms: Pipelined circuit switching, traffic permu tetion, dynamic path setup scheme, arbitrary scheme, dynamic power consumption.

I. INTRODUCTION

A trend of multiprocessor system-on-chip (MPSoC) design being interconnected with onchip networks is currently emerging for applications of parallel processing, scientific computing, and so on. Permutation traffic, a traffic pattern in which each input sends traffic to exactly one output and each output receives traffic from exactly one input, is one of the important traffic classes exhibite from on-chip multiproces - sing applications. Many of the MPSoC applications compute in real-time, therefore, guaranteeing throughput is critical for such permutation traffics. Most on-chip networks in practice are general-purpose and use can be implemented as source routing or distributed routing. However, such application -aware routings cannot efficiently handle the dynamic changes of a permutation pattern, which is exhibited in many of the application phases. The difficulty lies in the design effort to compute the routing to support the permutation changes in runtime, as well as to guarantee the permutated traffics. This becomes a great challenge when these permutation networks need to be implemented under very limited on-chip power and area overhead. Reviewing on-chip permutation networks (supporting either full or partial permutation) with regard to their implementation shows that most the networks employ a packet-switching mechanism to deal with the conflict of permuted data. Their implementations either use first-input first-output (FIFO) queues for the conflicting data or time-slot allocation in the overall system with the cost of more routing stages, or a routing algorithms such as dimension-ordered routing and minimal adaptive routing. This paper presents a novel silicon-proven design of an onchip permutation network to support guaranteed throughput of permutated traffics under arbitrary permutation.

Most on-chip networks in practice are general purpose and use routing algorithms such as dimension-ordered routing and minimal adaptive routing. To support permutation traffic patterns, on-chip permutation networks using application aware routings are needed to achieve better performance compared to the general-purpose networks. These application-aware routings are configured before running the applications and

* Dept. of Electronics and Communication Engineering, SRM University, Chennai, INDIA, *E-mail: Ashajyothi.993@gmail.com@gmail.com; Sarada.v@ktr.srmuniv.ac.in*

complex routing with a deflection technique that avoids buffering of the conflicting data. The choices of network design factors, i.e., topology, switching technique and the routing algorithm, have different impacts on the on-chip implementation.

System on- Chip (SoC), composed of heterogeneous cores on a single chip, has entered billion-transistor era. As the microprocessor industry is moving from single-core to multicore and eventually to many-core architectures, containing tens to hundreds of identical cores arranged as chip multiprocessors, which also require efficient communications among processors. Both SoC and microprocessor call for a high-performance, flexible, scalable, and design-friendly interconnection. How to provide efficient communication poses a challenge to researchers. Before the advent of network-onchip, interconnection architectures are usually based on dedicated wires or shared buses.

Dedicated wires provide point-to-point connection between every pair of nodes, effective for small systems of a few cores. But as the number of cores increases, the number of wires in the point-to-point architecture grows quadratically, making it unable to scale. Compared to dedicated wires, a shared bus which is a set of wires shared by multiple cores is more scalable and reusable.

However, due to the inherent disadvantage of buses, only one communication transaction is allowed at a time, blocking communication for all other cores. The disadvantages of shared bus architectures include long data delay, high energy consumption, increasing complexity in decoding/ arbitration, low bandwidth. It would be daunting inefficient if hundreds of nodes are connected by shared buses. Thus, the usage of shared buses is limited to a few dozens of IP cores. To deal with the problems in shared buses, a hierarchical architecture, which segments bus into shorter ones, is introduced. Hierarchical bus architectures may relax some of constraints faced by dedicated wires and shared buses, since different buses may account for different bandwidth needs, protocols and also increase communication parallelism. Nonetheless, scalability remains a problem for hierarchical bus architectures.

In order to meet the communication requirements, accelerate Time-to-market and cut down the communication energy consumption of large scale SoCs, there is a great need to find a new design alternative to the conventional point-topoint and bus based computation architectures.

II. EXISTING METHODOLOGY

Regarding the switching technique, packet switching requires an excessive amount of onchip power and area for the queuing buffers (FIFOs) with pre-computed queuing depth at the switching nodes and/or network interfaces. Regarding the routing algorithm, the deflection routing is not energy-efficient due to the extra hops needed for deflected data transfer, compared to a minimal routing. Moreover, the deflection makes packet latency less predictable; hence, it is hard to guarantee the latency and the in-order delivery of data. This paper presents a novel silicon-proven design of an on-chip permutation network to support guaranteed throughput of permutated traffics under arbitrary permutation.

Unlike conventional packet-switching approaches, our on-chip network employs a circuit switching mechanism with a dynamic path-setup scheme under a multistage network topology. The dynamic path setup tackles the challenge of runtime path arrangement for conflict-free permuted data. The pre-configured data paths enable a throughput guarantee. By removing the excessive overhead of queuing buffers, a compact implementation is achieved and stacking multiple networks to support concurrent permutations in runtime is feasible.

In our synthesis approach, we use accurate delay and power models for the network components (switches and links) that are obtained from layouts of the components using industry standard tools. The synthesis approach utilizes the floor plan knowledge of the NoC to detect timing violations on the NoC links early in the design cycle.

This leads to a faster design cycle and quicker design convergence across the high-level synthesis approach and the physical implementation of the design. We validate the design flow predictability of our proposed approach by performing a layout of the NoC synthesized for a 25-core CMP. Our approach maintains the regular and predictable structure of the NoC and is applicable in practice to existing NoC architectures.

(A) The Switch And Network Taxonomy

The switch is the other important component in IIP and has a central function in NoC. Responsible for routing data packets, it implements the network (sending resource-to-receiving resource routing) and link layer (switch-to-switch routing) when receiving a data packet, the switch extracts the header information, makes routing decision based on the header information and current traffic load (to avoid congestion) and performs appropriate action (put the packet onto a link, delay the packet, drop the packet, etc.). So far, the NoC has been described as a communication network based on data packets and the high-level logic function of the switch is routing the packets.

For different network cores, different approaches may be used for data packet routing. In the following text, the traditional telecommunications network taxonomy (also apply on NoC), which determines the low-level architecture and Implementation of the switch will be studied.

A traditional telecommunications network either employs circuit or packet switching. A link in a circuit switched network can use either frequency-division multiplexing (FDM) or timedivision multiplexing (TDM) while packet switched networks are either virtual circuit (VC) networks or datagram networks. This classification can be generalized and apply on any network core, including NoC.

(B) Packet Switching

Depending on the routing method, packet switched networks are divided into virtual circuit networks and datagram networks. The virtual circuit approach is connection-oriented and resembles the circuit switching. Both packet switched VC network and circuit switched network are suitable for uniform data traffic with long lifetime. For other bursty traffic, the connection management will tend to be computationally demanding and occupy a large portion of the bandwidth. They also require that the switches maintain the state information, resulting in more complex switch architecture and signaling scheme between switches.

To reduce the switch complexity and therefore also the area overhead, datagram switching can be used. The datagram based switch is stateand memory less, each packet is treated independently, with no reference to preceding packets. This approach more easily adapts to changes in the network such as congestion and dead links. However, it does not guarantee that packets with same source and destination will follow the same route. Consequently, the delay of packets with same source and destination may vary and packets may also arrive out of order, requiring buffering element at the receiving end. A datagram based switch implementation is described in.

(C) Circuit Switching

A circuit switched network requires a dedicated end-to-end circuit (with a guaranteed constant bandwidth) between the transmitting and the receiving end. As the “circuit” is an abstract concept, most of the time, it is not a physical end to end wire, but can span over many links. In a telecommunications network, the circuit is typically implemented with either frequency division multiplexing or time-division multiplexing in each link. With FDM, the frequency spectrum of a link is shared among the connections across the link. For obvious reasons, the FDM is not suitable for NoC. For TDM on the other hand, time is divided into frames of fixed duration, and each frame is divided into a fixed number of time slots as shown in Figure 10. When the network establishes a connection (or circuit) across a TDM link, the network dedicates a certain number of time slots in every frame to the connection. These slots are dedicated for the sole use of that connection, with some time slots available for use (in every frame) to transmit the connections data. The Ethereal Network on Chip developed at Philips Research is based on the time-division multiplexed circuit switching approach described above.

III. PROPOSED ON-CHIP NETWORK DESIGN

To meet the growing computation-intensive applications and the needs of low-power, high performance systems, the number of computing resources in single-chip has enormously increased, because current VLSI technology can support such an extensive integration of transistors. By adding many computing resources such as CPU, DSP, specific IPs, etc to build a system in System-on-Chip, its interconnection between each other becomes another challenging issue. In most System-on-Chip applications, a shared bus interconnection which needs arbitration logic to serialize several bus access requests, is adopted to communicate with each integrated processing unit because of its low-cost and simple control characteristics. However, such shared bus interconnection has some limitation in its scalability because only one master at a time can utilize the bus which means all the bus accesses should be serialized by the arbitrator. Therefore, in such an environment where the number of bus requesters is large and their required bandwidth for interconnection is more than the current bus, some other interconnection methods should be considered.

This network has a rearrange able property that can realize all possible permutations between its input and outputs. The choice of the three stage Clos network with a modest number of middle-stage switches is to minimize implementation cost, whereas it still enables a rearrange able property for the network.

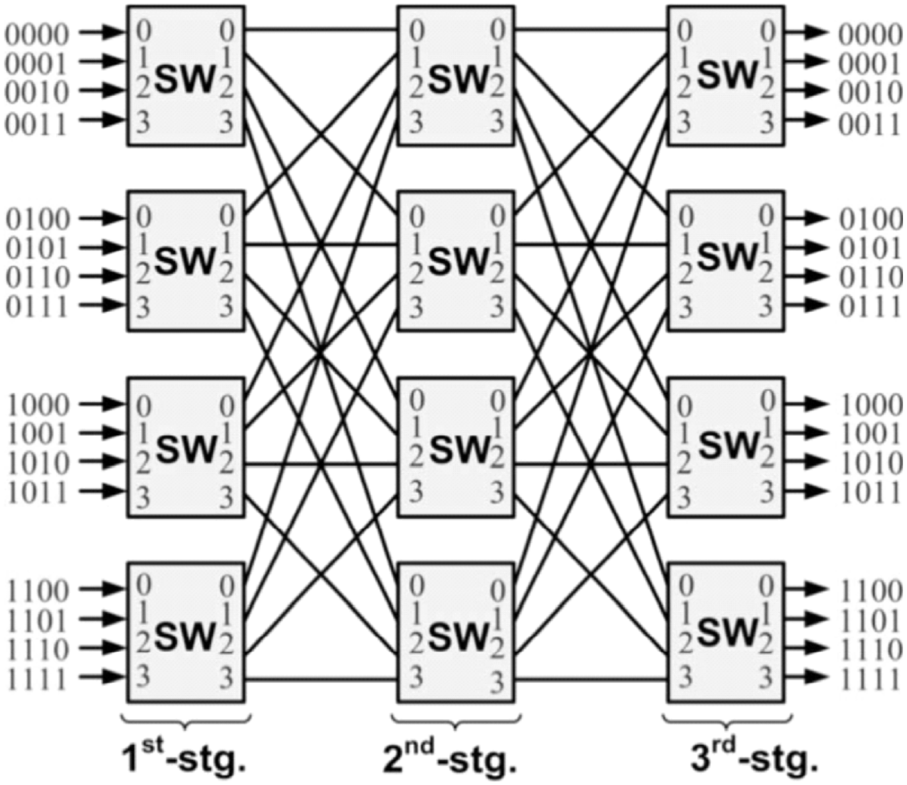
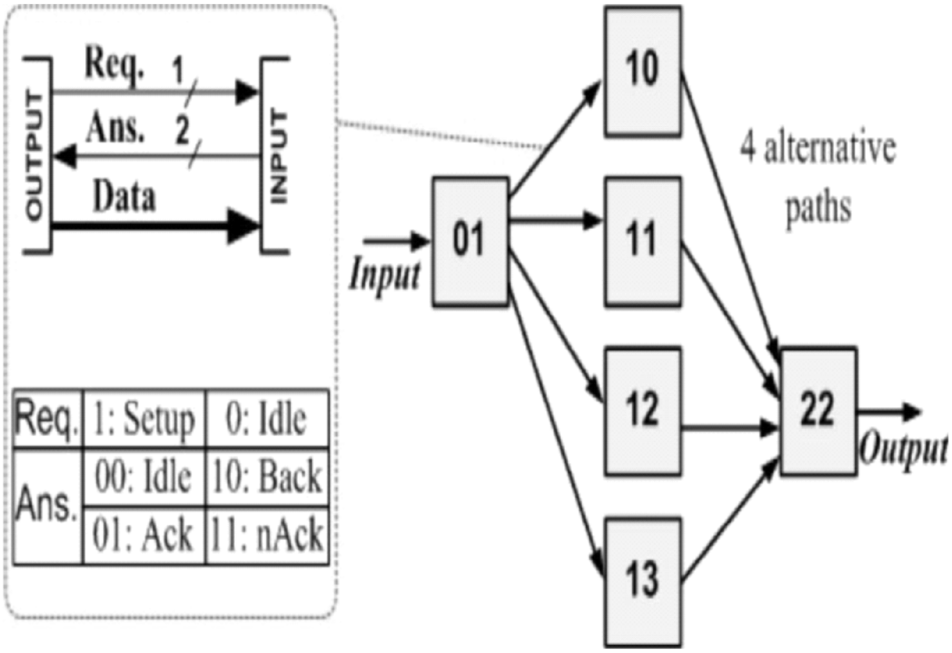


Figure 1: Proposed on-chip Network Topology with Port Addressing Scheme

Unlike conventional packet-switching approaches, our on-chip network employs a circuit-switching mechanism with a dynamic path setup scheme under a multistage network topology.

The dynamic path setup tackles the challenge of runtime path arrangement for conflict-free permuted data. The pre-configured data paths enable a throughput guarantee. By removing the excessive overhead of



Req.	1: Setup	0: Idle
Ans.	00: Idle	10: Back
	01: Ack	11: nAck

Figure 2: Switch-by-Switch Interconnection and Path-Diversity Capacity

queuing buffers, a compact implementation is achieved and stacking multiple networks to support concurrent permutations in runtime is feasible.

Figure 2. The bit format of the handshake includes a 1-bit *Request (Req)* and a 2-bit *Answer (Ans)*. Req==1 is used when a switch requests an idle link leading to the corresponding downstream switch in the setup phase. The Req==1 is also kept during data transfer along the set up path. A Req==0 denotes that the switch releases the occupied link. This code is also used in both the setup and the release phases. Ans Ans == 01(ack) means that the destination is ready to receive data from the source. When the Ans == 01propagates back to the source, it denotes that the path is set up, and then a data transfer can be started Immediately.

An Ans == 11(n ack) is reserved for end-toend flow control when the receiving circuit is not ready to receive data due to being busy with other tasks, or overflow at the receiving buffer, etc. An Ans == 10 (Back) means that the link is blocked. This Back code is used for a backpressure flow control of the dynamic path-setup scheme, which is discussed in the following subsection. Module of network interconnection like switching logic, routing algorithm and its packet definition should be light-weighted to result in easily implemental solutions.

The circuit-switching approach offers a guarantee of permuted data and its compact overhead enables the benefit of stacking multiple networks.

The circuit-switching approach combined with dynamic path-setup scheme under a Clos network topology, the proposed design offers arbitrary traffic permutation in runtime with compact implementation overhead.

(A) Dynamic Path Setup to Support Path Arrangement

A dynamic path-setup scheme is the key point of the proposed design to support a runtime path arrangement when the permutation is changed. Each path setup, which starts from an input to find a path leading to its corresponding output, is based on a dynamic probing mechanism. The concept of probing is introduced in works, in which a probe (or setup flit) is dynamically sent under a routing algorithm in order to establish a path towards the destination. Exhausted profitable backtracking (EPB) is proposed to use to route the probe in the network work. A path arrangement with full permutation consists of sixteen path setups, whereas a path arrangement with partial permutation may consist of a subset of sixteen path setups.

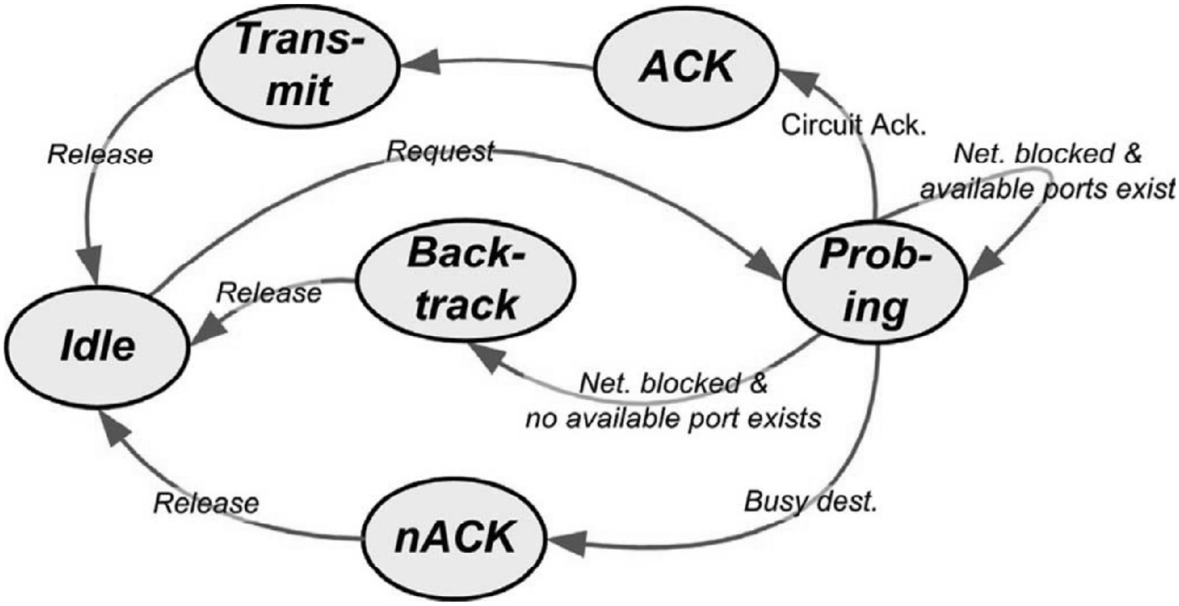


Figure 3: Switch-by-Switch interconnection and path diversity capacity

The path acquisition procedure requires the path-setup packet to travel a number of electronic routers and undergo some processing in each hop. Contention may cause the packet to be blocked, leading to path-setup latency on the order of tens of nanoseconds. Once a path is acquired, the transmission latency of the optical data is very short, depending only on the group velocity of light in a silicon waveguide: approximately 6:6_107 m/s, or 300 ps for a 2cm path crossing a chip. This latency mismatch is fundamental to intrachip optical communications: the network control and arbitration latency, determined by the electronic propagation velocity and processing speed, can impede the full exploitation of the latency advantages of optical transmission. This is independent of whether the control is performed in a centralized or distributed fashion. The path setup procedure is, therefore, a key issue in determining the performance of the photonic NoC. Reductions in path-setup latency will directly translate to improved efficiency of the network interfaces, to higher average bandwidth, and to better exploitation of the optical medium.

For a given source-destination pair, the setup latency can be expressed as D = (H 1)_tp+tq, where H is the number of hops in the packet's path, tp is the processing latency in each router and tq is the total additional latency due to contentions. Contentions in the path-setup phase are handled by queuing the path-setup packet until the message blocking its path is torn down and the path is cleared. Simulations show that tq is a major contributor to the overall setup latency, especially when the network is heavily loaded. To reduce the contention-based setup latency, tq, a new method of handling congestion is suggested. The new method relies on the fact that the actual processing latency in the pathsetup phase, (H 1)_tp, is typically much lower than the contention-based latency. In the suggested methods, the buffering depth in the electronic router is reduced to zero. This means that when a path-setup packet is blocked, it is immediately dropped, and a packet-dropped packet is sent, on the control network, in the opposite direction to notify the sender. The sender can immediately attempt to set up an alternative path, exploiting the network's path multiplicity.

With an adequate level of path-multiplicity, it is reasonable to assume that an alternative path can be found faster than it would take for the message obstructing the original path to be torn down.

Using the OMNET++ based POINTS simulator, a 36-core system with a photonic NoC and a path-multiplicity factor of _2 is simulated. The latency components in POINTS are based on predicted individual latencies of electronic and silicon-photonic components in a future 22nm process, and the optical message size is 16 Kbyte.

(B) Pipelined Circuit Switching

Pipelined circuit switching (PCS) has been suggested as an efficient switching method for supporting inters processor communication in multicomputer networks due to its ability to preserve both communication performance and fault-tolerant demands in such networks. The torus has been the underlying topology for a number of practical multicomputers. Analytical models of fully adaptive routing have recently been proposed for PCS in the torus under the uniform traffic pattern. However, there has not been any similar analytical model of PCS under the non-uniform traffic pattern, such as that generated by the presence of hot spots in the network. This paper proposes a new analytical model of PCS in the torus operating under the hot spot traffic pattern. Results from simulation experiments show close agreement with those predicted by the analytical model.

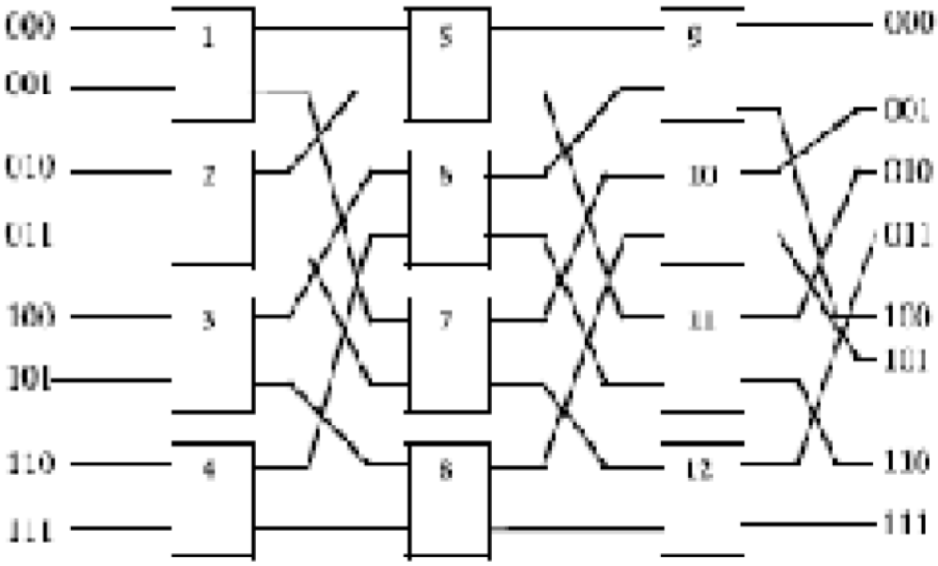


Figure 4: Multistage Interconnection Network and an Example 8*8 Shuffle Exchange Network (SEN)

In circuit-switching networks, the path between the source and destination is first determined, all links along that path are reserved, and no buffers are needed in each node. After data transfer, reserved links are released for use by other messages.

An important characteristic of the circuitswitching technique is that the source and destination are guaranteed a certain bandwidth and maximum latency when communication is established between them. This static bandwidth allocation regardless of the actual use is the main drawback of the circuit-switching approach.

However, static bandwidth allocation leads to a simple buffering strategy. In addition, circuitswitching networks are characterized by having the smallest amount of delay. This is because message routing overhead is only needed when the circuit is set up; subsequent messages suffer no, or minimal, additional delay. Therefore, circuitswitching networks can be advantageously used in the case of a large number of message transfers.

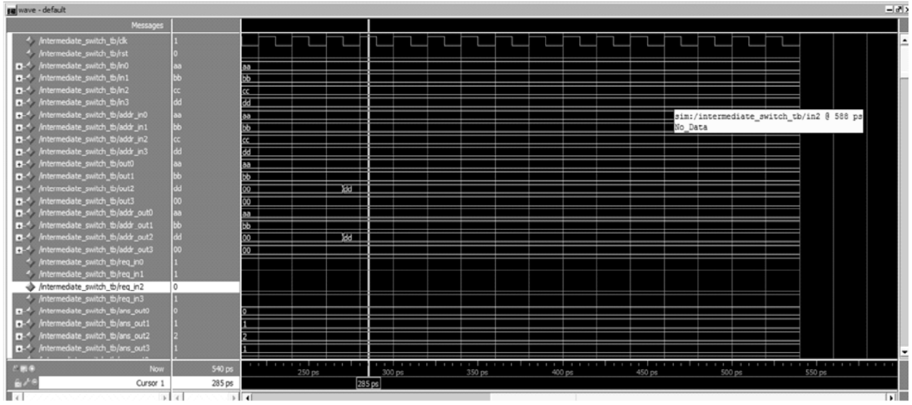
The store-and-forward switching mechanism provides an alternate data transfer scheme. The main idea is to offer dynamic bandwidth allocation to messages as they flow through the network, thus avoiding the main drawback of the circuitswitching mechanism.

Two main types of store-and-forward networks are common. These are packet-switched and virtual cut-through networks.

(C) Guaranteed Throughput

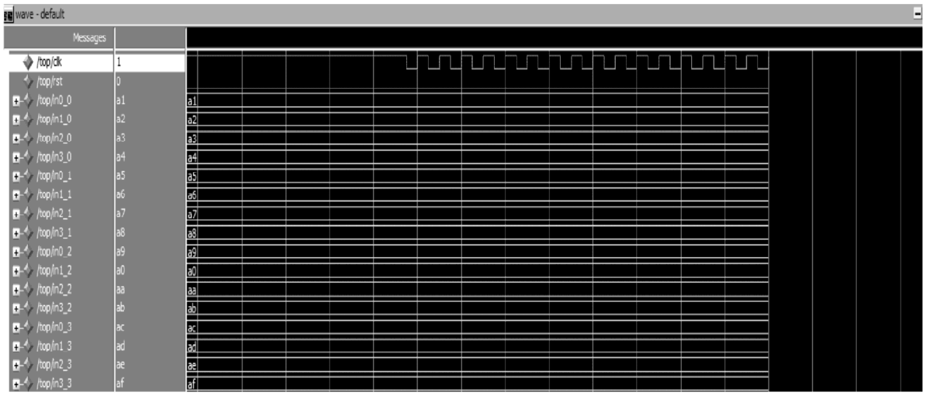
The possibilities of providing throughput guarantees in a network-on-chip by appropriate traffic routing. A source routing function is used to find routes with specified throughput for the data streams in a streaming multiprocessor systemon-chip. The influence of the routing algorithm, network topology and communication locality on the routing performance are studied. The results show that our method for providing throughput guarantees to streaming traffic is feasible. The communication locality has the strongest influence on the routing performance while the routing algorithm has weakest influence. Therefore, the mapping algorithm is of greater importance for the system performance than the routing algorithm and it is profitable to use a more complex mapping algorithm that preserves the communication locality together with a simple routing algorithm.

IV. IMPLEMENTATION RESULTS



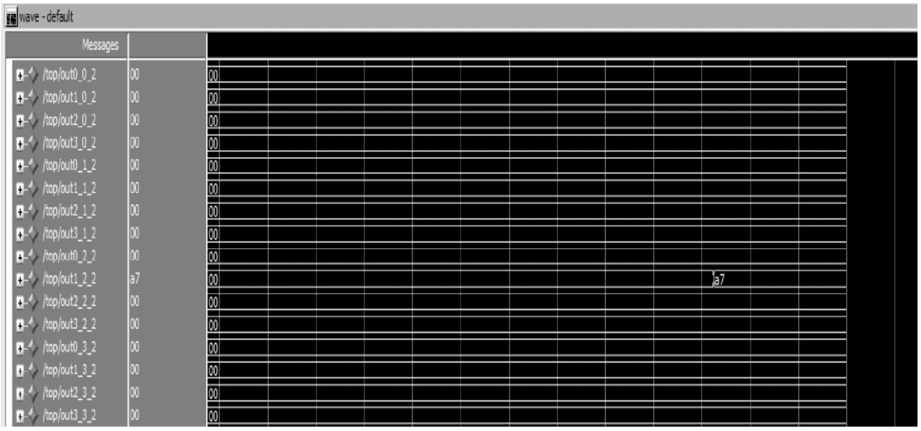
```
wave - default
Messages
/intermediate_switch_tb/clk        1
/intermediate_switch_tb/rst        0
/intermediate_switch_tb/in0        aa
/intermediate_switch_tb/in1        bb
/intermediate_switch_tb/in2        cc
/intermediate_switch_tb/in3        dd
/intermediate_switch_tb/addr_in0   aa
/intermediate_switch_tb/addr_in1   bb
/intermediate_switch_tb/addr_in2   cc
/intermediate_switch_tb/addr_in3   dd
/intermediate_switch_tb/out0       aa
/intermediate_switch_tb/out1       bb
/intermediate_switch_tb/out2       dd
/intermediate_switch_tb/out3       00
/intermediate_switch_tb/addr_out0  aa
/intermediate_switch_tb/addr_out1  bb
/intermediate_switch_tb/addr_out2  dd
/intermediate_switch_tb/addr_out3  00
/intermediate_switch_tb/req_in0    1
/intermediate_switch_tb/req_in1    1
/intermediate_switch_tb/req_in2    0
/intermediate_switch_tb/req_in3    1
/intermediate_switch_tb/ans_out0   0
/intermediate_switch_tb/ans_out1   1
/intermediate_switch_tb/ans_out2   2
/intermediate_switch_tb/ans_out3   1
Now       540 ps
Cursor 1  285 ps

sim:/intermediate_switch_tb/in2 @ 588 ps
No_Data
```

Figure 5: Simulation of Data Part for Single Switch



```
wave - default
Messages
/top/clk    1
/top/rst    0
/top/in0_0  a1
/top/in1_0  a2
/top/in2_0  a3
/top/in3_0  a4
/top/in0_1  a5
/top/in1_1  a6
/top/in2_1  a7
/top/in3_1  a8
/top/in0_2  a9
/top/in1_2  a0
/top/in2_2  aa
/top/in3_2  ab
/top/in0_3  ac
/top/in1_3  ad
/top/in2_3  ae
/top/in3_3  af
```

Figure 6: Data at the Input Ports



```
wave - default
Messages
/top/out0_0_2  00
/top/out1_0_2  00
/top/out2_0_2  00
/top/out3_0_2  00
/top/out0_1_2  00
/top/out1_1_2  00
/top/out2_1_2  00
/top/out3_1_2  00
/top/out0_2_2  00
/top/out1_2_2  a7
/top/out2_2_2  00
/top/out3_2_2  00
/top/out0_3_2  00
/top/out1_3_2  00
/top/out2_3_2  00
/top/out3_3_2  00
```

Figure 7: Data at the Output Ports

V. LOW POWER ANALYSIS

Low power has emerged as a principal theme in today's electronics industry. The need for low power has caused a major paradigm shift where power dissipation has become as important a consideration as performance and area. In the past, the major concerns of the VLSI designer were area, performance, cost and reliability; power consideration was mostly of only secondary importance. Power dissipation in CMOS circuits is caused by three sources: 1) the leakage current which is primarily determined by the fabrication technology, consists of reverse bias current in the parasitic diodes formed between source and drain diffusions and the bulk region in a MOS transistor as well as the sub threshold current that arises from the inversion charge that exists at the gate voltages below the threshold voltage, 2) the short-circuit (rush-through) current which is due to the DC path between the supply rails during output transitions and 3) the charging and discharging of capacitive loads during logic changes.

The short circuit and leakage currents in CMOS circuits can be made small with proper circuit and device design techniques. The dominant source of power dissipation is thus the charging and discharging of the node capacitances (also referred to as the dynamic power dissipation) and is given by:

$$P = 0.5CV_{dd}^{2}E\ sw\ f_{clk}$$

Where C is the physical capacitance of the circuit, V_{dd} is the supply voltage, $E(sw)$ (referred as the *switching activity*) is the average number of transitions in the circuit per $1/f_{clk}$ time, and f_{clk} *is* the clock frequency.

VI. CONCLUSION AND FUTURE WORK

This paper proposes an on-chip multistage interconnection network with the least possible number of hardware, the minimum amount of wiring between stages and the minimum wire lengths. It can be used for high-performance inter processor communication in real-time applications. Although logN -MINs have been already researched and used in parallel super computers, they can be adapted also for networkon-chips as well. High bandwidth and low latency are combined with a deterministic behavior of interprocessor communication in the proposed NoC. The objective is to use MPSoCs in high performance embedded systems with hard real time constraints that can be found in electronic control units for cars or for production machinery. On-chip network achieves reduction of silicon overhead compared to other design approaches. Test-chip validates the feasibility and efficiency of the proposed design. A silicon-proven test-chip validates the proposed design supporting traffic permutation in future MPSoC researches.

In future work we want to discuss the pros and cons of blocking and non-blocking MINs for real-time computing and their implementation in FPGA hardware. Blocking networks such as the Baseline network minimize the costs in hardware but they require a suitable scheduling strategy because not all permutations of sender/receiver pairs can be realized. Therefore further scheduling algorithms such as priority scheduling or earliest-deadline-first have to be considered for hardware implementation. In comparison with blocking networks, non-blocking networks as the Benes network can be operated without complex scheduling strategies since messages can be sent to each free receiver port at any time. On the other hand non-blocking networks exhibit extra costs in hardware plus they require complex routing algorithms due to the rearrangement of alternative connection paths.

VII. AKNOWLEDGEMENT

The Author expressing its gratitude to SRM Universities Research Lab for supporting this work.

REFERENCES

[1] S. Borkar, "Thousand core chips—A technology perspective," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2007, pp. 746–749.

[2] P.-H. Pham, P. Mau, and C. Kim, “A 64-PE folded-torus intra-chip communication fabric for guaranteed throughput in network-on-chip based applications,” in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, 2009, pp. 645–648.

[3] C. Neeb, M. J. Thul, and N. Wehn, “Network-on-chip-centric approach to interleaving in high throughput channeldecoders,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2005, pp. 1766–1769.

[4] H. Moussa, A. Baghdadi, and M. Jezequel, “Binary de Bruijn on-chip network for a flexible multiprocessor LDPC decoder,” in *Proc. ACM/ IEEE Design Autom. Conf. (DAC)*, 2008, pp. 429–434.

[5] H. Moussa, O. Muller, A. Baghdadi, and M. Jezequel, “Butterfly and Benes-based on-chip communication networks for multiprocessor turbo decoding,” in *Proc. Design, Autom. Test in Euro. (DATE)*, 2007, pp. 654–659.

[6] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, “An 80-tile sub-100-w TeraFLOPS processor in 65-nm CMOS,” *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, Jan. 2008.

[7] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks:*. San Francisco, CA: Morgan Kaufmann, 2004.

[8] N. Michael, M. Nikolov, A. Tang, G. E. Suh, and C. Batten, “Analysis of application-aware on-chip routing under traffic uncertainty,” in *Proc. IEEE/ACM Int. Symp. Netw. Chip (NoCS)*, 2011, pp. 9–16.

[9] P.-H. Pham, J. Park, P. Mau, and C. Kim, “Design and implementation of backtrackingwave-pipelineswitchto supportguaranteedthroughput innetwork-on-chip,”*IEEETrans.Very LargeScaleIntegr.(VLSI)Syst.*, 10.1109/ TVLSI.2010.2096520.

[10] D. Ludovici, F. Gilabert, S. Medardoni, C. Gomez, M. E. Gomez, P. Lopez, G. N. Gaydadjiev, and D. Bertozzi, “Assessing fat-tree topologies for regular network-on-chip design under nanoscale technology constraints,” in *Proc. Design, Autom. Test Euro. Conf. Exhib. (DATE)*, 2009, pp. 562–565.

[11] Y. Yang and J. Wang, “A fault-tolerant rearrangeable permutation network,” *IEEE Trans. Comput.*, vol. 53, no. 4, pp. 414–426, Apr. 2004.

[12] P. T. Gaughan and S. Yalamanchili, “A family of fault-tolerant routing protocols for direct multiprocessor networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 6, no. 5, pp. 482–497, May 1995.

[13] P. K. Meher, “Systolic and non-systolic scalable modular designs of finite field multipliers for Reed-Solomon Codec,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 6, pp. 747–757, Jun. 2009.