2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

Power and Performance Efficient Partial Circuits in Packet-Switched Networks-on-Chip

Nasibeh Teimouri[1], Mehdi Modarressi[2], Hamid Sarbazi-azad[1,3]
[1]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[2]Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[3]Schoolof Computer Science, Institute for Research in Fundamental sciences (IPM), Tehran, Iran
Email: n_teimouri@ce.sharif.edu, modarressi@ece.ut.ac.ir, azad@{sharif.edu, ipm.ir}

Abstract— **In this paper, we propose a hybrid packet-circuit switching for networks-on-chip to benefit from the advantages of both switching mechanisms. Integrating circuit and packet switching into a single NoC is achieved by partitioning the link bandwidth and router data-path and control-path elements into two parts and allocating each part to one of the switching methods. In this NoC, during injection in the source node, packets are initially forwarded on the packet-switched sub-network, but keep requesting a circuit towards the destination node. The circuit-switched part, at each cycle, collects the circuit construction requests, performs arbitration among the conflicting requests, and constructs circuits over the unallocated circuit-switched sub-network links. Unlike traditional circuit-switching, the circuit end point in this NoC is not necessarily the packet destination; rather the circuits can be terminated in any intermediate node between the packet source and destination nodes. At that node, the packet may either travel over another circuit (in case of successful circuit request) or continue its path over the packet-switched part. Therefore, packets may switch between the two sub-networks several times during their life-time in the network. Circuit construction is handled by a low-latency and low-cost setup network. To keep the complexity of the circuit construction low, the circuits are restricted to span within a neighborhood of d hops of the requesting node. The experimental results show considerable improvement in energy and latency over a traditional packet-switched NoC.**

Keywords: Network-on-chip; circuit-switching; packet-switching; power consumption; performance.

I. INTRODUCTION

In recent years, the size and complexity of application-specific multiprocessor system-on-chips (SoCs) and general-purpose chip multiprocessors (CMPs) has grown very fast. By moving from single-core to multi-core and then to many-core architectures, handling the on-chip communication becomes a more important and challenging issue [1]. Although packet-switching is the dominant switching method in today NoCs, some NoC architectures use circuit-switching, mainly to reduce on-chip communication power consumption. However, circuit-switching suffers from some problems including the long setup latency due to contention among circuit paths, and low resource utilization [2]. Combining packet switching with some other switching methods (often circuit-switching) has been also considered in some papers to reduce NoC power consumption [4], [5] and [6]. In [4], a hybrid circuit/packet-switched NoC is proposed. This NoC is potentially capable to benefit from the nice properties of the two popular switching methods: the scalability and high resource utilization of packet switching and power-efficacy of circuit switching. The authors just showed the viability of this approach by developing an initial dynamic circuit construction algorithm. This algorithm is very complex and requires a costly setup network which consumes much power.

In this paper, a similar NoC architecture is presented which applies a very efficient circuit construction algorithm to improve the setup network area overhead, as well as the efficiency and speed of the circuit construction algorithm.

In this work, a packet-switched and a circuit-switched sub-network work together to mitigate the shortcomings of each other. Although the two sub-networks can use separate links and routers, we use the Spatial-Division-Multiplexing (SDM) scheme to partition an n-bit NoC into two n/2-bit sub-networks each implementing one of the switching methods. SDM has been used in several previous works, for example to construct short-cut paths [8], provide certain QoS levels [9], and implement low-power hybrid packet-circuit switched [4].

A packet starts traveling on the packet-switched sub-network, but it always tries to move to the circuit-switched part. Every few cycles, at each node in the packet-switched part, the circuit-switched part is requested to construct a circuit towards its destination node. The circuit-switched part receives the circuit construction requests and returns a circuit for some of them in such a way that the cumulative length of the constructed circuits is maximized. Circuits can be set up over the links that are not currently used by existing circuits. Arbitration is needed if two packets request for circuits with overlapping links. These tasks are done by a separate and very light-weight and low-latency setup network.

Each node requests for a circuit by sending a probe message on the setup network towards the destination node. A probe message is forwarded by very simple nodes of the setup network until it reaches its destination or is blocked by some other conflicting probe messages. During its travel, it may also block some other probes. The node at which the probe message stops may be the circuit end node. At the other side, the nodes with free circuit-switched ports send out a message to their neighboring nodes to notify them about their free ports that can be used by the new circuits. These messages, named freedom tokens, can be directed by the probe messages to the paths with more probability of successful circuit construction.

1066-6192/12 $26.00 © 2012 IEEE
DOI 10.1109/PDP.2013.82

The idea of sending probe messages and freedom tokens to neighboring nodes in order to inform them respectively about circuit requests and the free nodes was first proposed in the Token Flow Control (TFC) method in [6]. The probe messages in our work are used to direct the freedom tokens to non-conflicting paths.

As the packet should decide whether to move to circuit-switched part or continue with packet-switching, the setup network must reply the circuit construction requests in a single cycle in order to avoid additional latency for the packets. Like the work in [4], we used low-latency on-chip wires for the setup network to have it carry the probe and freedom token messages very fast and reply the requests (by performing the circuit construction procedure) within a single cycle of the data network.

In on-chip networks, some parameters including wire width, wire spacing, and repeater size and spacing determine the latency, energy, and bandwidth characteristics of the links [7]. By increasing the wire width, spacing, and number of repeaters we can increase the wire speed, but the area and power consumption of the links are also increased. The method in [4] put no limitation on the circuit length, so the setup network must be fast enough to construct circuits as long as the network diameter. This limits the scalability of that method. To address this problem, we limit the circuit length: they are restricted to span within a local neighborhood of d hops from the requesting node. As a result, unlike traditional circuit-switching, the circuits are not restricted to end to the packet destination, but they can end at some node along the path from the current node to the destination node. The capability to construct partial circuits greatly increases the resource utilization of the circuit-switched part. In case of unsuccessful circuit request, the packet goes to the next node and requests for a circuit again. On successful request, packets move to the circuit-switched part until reaching the circuit end node, where they will request for a circuit again. This may lead to several switching between the packet and circuit-switched parts.

II. PROPOSED ARCHITECTURE

A. The NoC Architecture

The base of the proposed architecture is the conventional architecture presented in [10]. Fig. 1 depicts this architecture, where the network is divided into two n/2-bit parallel sub-networks using SDM. One of the n/2-bit sub-networks directs the packets according to the traditional packet-switching scheme (PS sub-network) and the other sub-network is used to build partial circuits (PC) to shorten the packet's path to its destination (CS sub-network). The switch allocator is slightly modified, compared to the conventional arbiter. It is divided into two parts to handle the CS and PS flits separately. CS allocator is a register indicating whether the corresponding output port is a part of a PC. The value of this register is set when PC is built by setup network.

The pipeline stages in our router are similar to the pipeline stages in [4] and similar to this work, our setup network configures input port and crossbar connection of every router along the PC. As displayed in Fig. 1, received packet may continue its path in its current sub-network or switch to other sub-network.

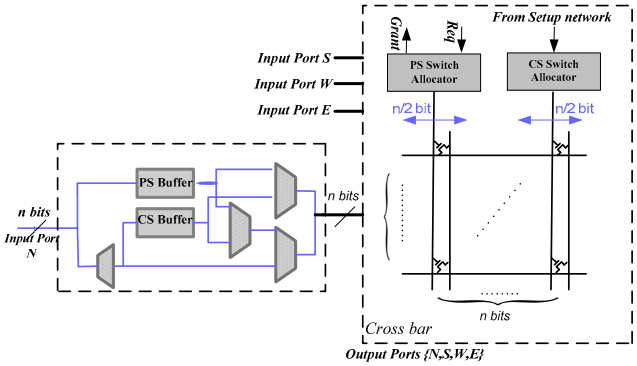


Figure 1. The router architecture and the details of one of the input ports

As shown in the Fig. 1, there are 3 extra multiplexers and 1 de-multiplexer in the router input ports in our proposal data network, compared to the conventional router in [10]. Our evaluation shows that the proposed architecture imposes about 2% area overhead over the conventional NoC architecture (for example, for a 64-bit 6×6 NoC with 8-flit buffers). We modeled the area of a conventional and the proposed router using the analytical NoC area models presented in [11]. We also used the area values in [14] to set the parameters of the model.

III. PC CONSTRUCTION

A. Setup network structure

In our proposed architecture, we use the setup network which is designed based on the reconfigurable mesh presented in [13] to provide ultra-low latency and low-power communications for propagating control messages (probe and freedom tokens) through the network.

In on-chip networks, by tuning wire width, wire spacing, and repeater size and spacing, we can design wires with varying latency, energy, and bandwidth characteristics in comparison to the traditional off-chip interconnections [5], [7]. We employ wire properties to design an ultra-low latency setup network to carry messages needed to perform the PC construction steps. Having a small bit-width and a simple structure, the area of the setup network is negligible in comparison to the main data-network.

As mentioned, our setup network is responsible to transfer the information about the free router ports (freedom tokens) and PC requests (probe) throughout the network required by our algorithm.

The probe messages are created by the header flits which are allowed to request a PC. The copies of probe messages are forwarded in the network toward their destinations based on XY routing. One of our reasons in order to apply this limitation on probe path and not sending the probe messages in all direction is to avoid congestion in the setup network.

In this work, we propose an algorithm according to this architecture where at every few cycles, a set of circuit

requests (probe messages) from the packets in some nodes is received and after that the longest possible partial circuit (PC) for some requesting nodes is found over the free CS sub-network links.

Setting up the PCs on CS sub-network includes appropriately configuring the input ports and crossbar switches by setup network. Establishing PCs with maximum length of d hops needs d steps of propagating probe and token messages in the setup network and configuring PCs, so the Setup network clock cycle must be set in such a way that during each cycle of the data network, these tasks can execute. As mentioned before, in order to avoid frequent PC construction overheads in setup network, this procedure is called every few cycles of date network clock cycles.

IV. PC CONSTRUCTION ALGORITHM

A fast algorithm for constructing and tearing down a PC between two nodes is implemented on the setup network nodes. In this section, the algorithm of constructing PCs based on packet requests is described.

There are two procedures in our proposed algorithm which are called at each step of the algorithm: Freedom Advertisement and Probe Propagation which are performed in parallel.

Freedom Advertisement, Each node sends the list of its free ports to the nodes at a given distance. The messages that contain this information are called "Freedom Tokens". Freedom tokens sent by a node are either the information of its own ports or the information of the other nodes in the network which have been received by the node in previous steps.

Probe Propagation, Each node that hosts a requesting packet sends a probe message in order to advertise the destination node of the packet. They also send the probes received in previous steps. The propagation of probes is useful when our first procedure needs to select one node among the possible nodes to send freedom tokens to it. The selected node is the one which hosts a probe indicating the necessity of receiving the freedom token for constructing a PC.

In the first step, each node only sends its own tokens and probes (if any) to the network. In the next steps, the nodes forward the tokens and the probes received at the previous steps from neighboring nodes over the setup network.

Assuming d= 3, we describe the setup-network steps of our algorithm in order to construct a PC. The propagation procedure is simple and deterministic; at each step it forwards the copies of probes at each node toward their destinations using XY routing scheme. In fact, since probe broadcasting in all directions (and not only XY) can cause loss of freedom tokens due to using several freedom tokens for a requesting packet over multiple paths, the procedure uses the XY path and do not broadcast probes to the network.

In our method, at step i (i >1), Freedom tokens originated from each node reach only one node at distance i from the source node. This certain node is selected based to existent probes in this node (if any), which will use the PC which will be constructed. In case there are more than one node in each step that meet this condition, one of them is selected randomly.

At the end of these d steps, each node in the network tries to make a PC based on received information, for the packet whose header flit is buffered in one of its input ports. The constructed PCs are remained till the tail flit of the packet passes over it and reaches the end node of PC; at this time the PC is released.

The nodes in network use received information for creating PCs. In this algorithm, E(A), W(A), N(A) and S(A) refer to the neighbors of node A in the east, west, north, and south directions, respectively. A hierarchical scheme is also used; for example the eastern neighbor of the western neighbor of A is represented by E(W(A)).

In the following paragraphs, the Advertisement steps of the proposed algorithm are presented.

Step1 - Advertisement: The purpose of this step is making network nodes aware about the available input ports at the nodes within the neighborhood of 1 hop. In this step, each node advertises its available unallocated ports that can be used to construct a PC to its adjacent nodes as displayed in Fig. 2 (left part). At the end of this step, each node knows its adjacent nodes' available ports.

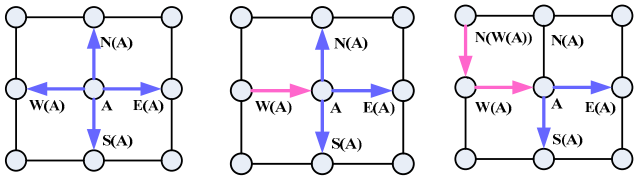


Figure 2. Examples of advertising freedom information in step 1(left part), step 2 (middle part), step 3 (right part)

Step2 - Advertisement: In this step, the messages continue traveling to reach to the nodes at the neighborhood of 2 hops. Sending out freedom information is categorized in 4 cases, which one of them is sending out the received information from the western neighbor of node A, W(A), to one of its eastern, northern and southern adjacent nodes, i.e. E(A), N(A), and S(A), respectively. This case is shown in the middle part of Fig. 2.

Selecting the token receiver node between the three neighboring nodes is based on existent probes which are hosted by them, in the similar state, and one of them is selected randomly.

Step3 - Advertisement: In this step, the messages reach to the nodes at distance 3 from the sender node. In this step, each node at the neighborhood of two hops from the sender sends a vector of the messages received from adjacent nodes to the next nodes that have a distance of 3 hops from the sender. One case out of twelve possible cases is sending out the received information from the western neighbor at the first level and the northern neighbor of the western neighbor, N(W(A)), to one of the A's southern or eastern neighbors,

i.e., S(A) or E(A), respectively. The right part of Fig. 2 shows this case.

As mentioned before, freedom tokens for more than 1 hop are sent out to only one node in order to avoid conflict between more than one requesting packets from different nodes during constructing PCs.

After finishing advertisement and propagation steps, PCs should be reserved by the setup network according to the probes and freedom tokens transferred over the network. PC reservation should be done based on the length of possible PCs in descending order.

Fig. 3 shows the case when E(A) has a header flit destined to W(N(W(A))) and S(A) has header flit to send to W(S(W(A))). In this case, E(A) is selected as the information receiver because its flits destination, W(N(W(A))), is closer to N(W(A)).

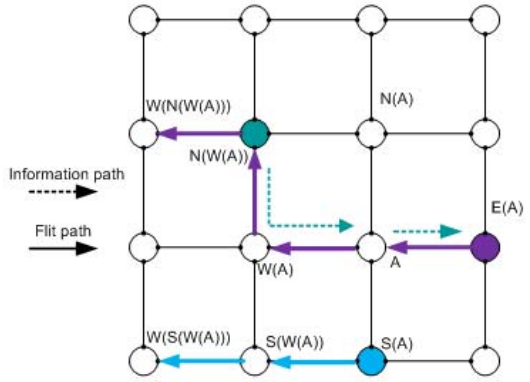


Figure 3. Selecting one node as receiever of freedom information about neighbourhood of 3 distance units

V. EXPERIMENTAL RESULTS

In this section, we evaluate the impact of the proposed hybrid switched NoC architecture on the NoC power consumption and latency under some realistic and synthetic benchmarks. The results are compared against a conventional mesh NoC with speculative 6-stage pipelined routers. In our proposed NoC, the PCs are built dynamically per packets.

Please note that for the two NoCs (proposed and conventional) simulations are performed for a 64-bit system with speculative 6-stage pipelined routers, 2 virtual channels per port, and 8-flit buffer. Each port has two virtual channels. When constructing PCs, following XY routing, only 2 turns are allowed to guarantee deadlock freedom and avoid increasing area overhead of the network. In all simulation experiments, PC construction procedure is not performed at every data network clock cycle and it is performed at every 5 data network cycles once.

We have evaluated the proposed NoC architecture using Booksim, a fully parameterized cycle accurate simulator for interconnection networks [14]. We have integrated the Orion power library [12] to Booksim to calculate the power consumption of the NoCs. The power results reported by Orion are based on a NoC implemented in 65 nm technology and the working frequency of the NoC is set to 2 GHz.

A. Synthetic Traffic Results

Since a high degree of temporal and spatial communication locality has been observed in most multi-core SoC and scientific CMP workloads [3][16], we use a 2-hot flow traffic pattern (n=2). Each packet is 8 flits long. Each simulation runs for 5,000,000 cycles.

Fig. 4 shows the PDP (power-delay product) of the proposed and conventional NoCs as a function of the injection rate (packet/node/cycle) under 2-hot flow traffic in a 6×6 mesh NoC. As shown in Fig. 4, the proposed NoC architecture improves the PDP over the other considered NoC design.

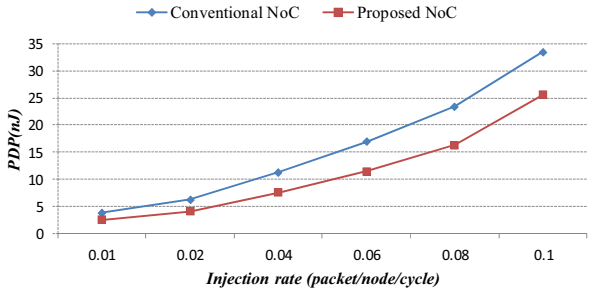


Figure 4. The Power Delay Product (PDP) of the proposed and a conventional 6×6 NoC under 2-hot flow traffic for different injection rates

B. SPLASH-2 Results

We then evaluate the effectiveness of the proposed switching mechanism under the traffic traces generated from SPLASH-2 benchmarks [15]. Fig. 6 shows power consumption and latency of the proposed 7×7 NoC in comparison with the conventional 7×7 NoC under SPLASH-2 traffic traces. We set simulation clock frequency to 2 GHz, while each simulation runs for 5,000,000 cycles.

As Fig. 6 shows, on average, our proposed NoC outperforms the conventional NoC by 25% and 37% when considering the dynamic power consumption and average packet latency, respectively.

C. Overhead analysis

As mentioned above, the proposed reconfigurable NoC structure relies on a setup network to initialize partial circuits. Our setup and data networks are similar to the conventional NoC structure in many ways. In this work, we assume the setup network is structured as a reconfigurable on-chip network [13] that is placed in parallel to the baseline network. With respect to this structure, setup network overhead depends on the number of ports per each router, the buffer size of the input buffer, and the bit-width of the inter-router links. Using the methods which are presented in [14] and applying values from [11], we have evaluated the area estimation of the proposed setup network as 10% of the

conventional NoCs with the same topology, whereas the proposed data network imposes only about 2% area overhead to the conventional network. Overall, the estimated area overhead using the proposed reconfigurable NoC structure is about 12% against conventional packet-switched mesh-based NoCs.

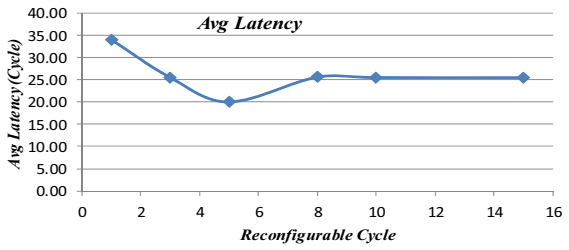
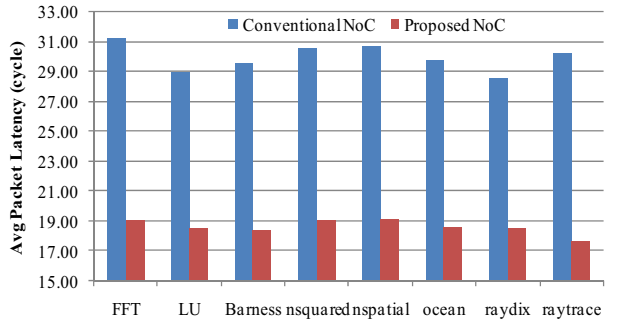
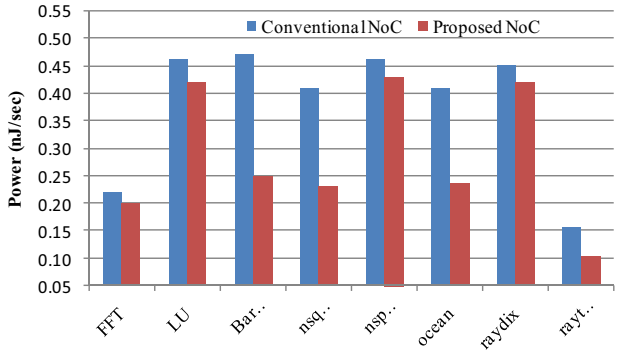


Figure 5. The effect of reconfigurable cycle on performace





The average packet latency (cycles) and dynamic power consumption (nJ/sec) for the SPLASH-2 programs

D. Sensitivity analysis

As mentioned before, the procedure of PC construction and network reconfiguration is performed at every 5 data network clock cycles. Fig. 5 reports the average packet latency when running seven Splash workloads with the proposed 7×7 NoC as a function of reconfiguration cycles. As can be seen in Fig. 5, considering reconfiguration at every 5 cycles results in the best performance as with higher numbers there are many packets which pass over the PS sub-network and with lower numbers frequent PC constructions imposes more overhead and thus reduces the performance.

VI. CONCLUSIONS

In this work, we integrated a packet-switched and a circuit switched network into a single architecture. Besides the packet-switched sub-network which implemented the traditional packet-switching operation, circuit-switched sub-network was used to direct packets over circuits by bypassing some middling nodes. Unlike traditional circuit-switching, the circuit end point in this NoC is not necessarily the packet destination; rather the circuits can be terminated in any intermediate node between the packet source and destination nodes.

In our architecture, a light-weight and low-latency setup-network was used to construct the circuits. The experimental results showed considerable improvements of NoC power and performance compared to a conventional packet-switched NoC of the same cost.

VII. REFERENCES

[1] L. Benini and G. De Micheli, “Networks on Chip: a New Paradigm for Systems on Chip Design,” DATE, 2002, pp. 418-419.

[2] K. Chang, “A Low-Power Crossroad Switch Architecture and Its Core Placement for Network-On-Chip,” ISLPED, 2005, pp.375-380.

[3] M. Modarressi, A. Tavakkol, H. Sarbazi-Azad, ”Virtual Point-to-Point Connections in NoCs,” Computer-Aided Design for Integrated Circuits and Systems, vol. 29, 2010, pp. 855-868.

[4] M. Modarressi, H. Sarbazi-Azad, M. Arjomand, “An SDM-Based Hybrid Packet-Circuit-Switched On-Chip Network,” DATE, 2009, pp. 566-569.

[5] T. Krishna, A. Kumar, P. Chiang, M. Erez, L. Peh, ”NoC with Near-Ideal Express Virtual Channels Using Global-Line Communication,” Symposium on High Performance Interconnects, 2008.

[6] A. Kumar, L. Peh, N. Jha, "Token flow control," IEEE/ACM International Symposium on Microarchitecture, 2008, pp. 342-352.

[7] R. Balasubramonian, “Microarchitectural Wire Management for Performance and Power in Partitioned Architectures,” HPCA, 2005, pp.28-39.

[8] N. Teimouri, M. Modarressi, A. Tavakkol, H. Sarbazi-Azad, “Energy-Optimized On-Chip Networks Using Reconfigurable Shortcut Paths,” ARCS, 2011, pp. 231-242

[9] A. Morgenshtein, A. Kolodny, R. Ginosar, “Link Division Multiplexing (LDM) for Network-on-Chip Link,” IEEE International Symposium on Electrical and Electronics Engineers, 2006,

[10] W. J. Dally, B. Towles, ”Principles and practices of interconnection networks,” Morgan Kaufmann Publishers, 2004.

[11] J. Balfour, W.J. Dally, ”Design tradeoffs for tiled CMP on-chip networks,” The International Conference of Supercomputing, 2006, pp. 178-189.

[12] A. Kahng, B. Li, L. Peh, K. Samadi, “ORION 2.0: A Fast and Accurate NoC Power and Area Model for Early-Stage Design Space Exploration,” DATE, 2009, pp.423-428.

[13] R. Vaidyanathan, J. Trahan, “Dynamic Reconfiguration: Architectures and Algorithms”, Springer, 2004.

[14] Booksim NoC simulator, http://nocs.stanford.edu/booksim.html

[15] SPLASH-2, http://www.flash.stanford.edu/apps/SPLASH

[16] N.E. Jerger, M. Lipasti, L. Peh, ”Circuit-Switched Coherence,” Computer Architecture Letters, Vol. 6, 2007, pp. 5-8.