A Hybrid Packet-Circuit Switched On-Chip Network Based on SDM

Mehdi Modarressi, Hamid Sarbazi-Azad, Mohammad Arjomand
Computer Engineering Department, Sharif University of Technology, Tehran, Iran
IPM School of Computer Science, Tehran, Iran
{modarressi, arjomand}@ce.sharif.edu, azad@sharif.edu

***Abstract*—In this paper, we propose a novel on-chip communication scheme by dividing the resources of a traditional packet-switched network-on-chip between a packet-switched and a circuit-switched sub-network. The former directs packets according to the traditional packet-switching mechanism, while the latter forwards packets over circuits which are directly established between two non-adjacent nodes by bypassing the intermediate routers. A packet may switch between the sub-networks several times to reach its destination. The circuits are set up using a low-latency and low-cost setup-network. The network resources are split between the two sub-networks using Spatial-Division Multiplexing (SDM). The work aims to improve the power and performance metrics of Network-on-Chip (NoC) architectures and benefits from the power and scalability advantage of packet-switched NoCs and superior communication performance of circuit-switching. The evaluation results show a significant reduction in power and latency over a traditional packet-switched NoC.**

Keywords-network-on-chip;circuit-switching;packet-switching.

I. INTRODUCTION

Current multiprocessor system-on-chips have grown in size and complexity in recent years and future MPSoCs will consist of complex integrated components communicating with each other at very high-speed rates. Network-on-chip (NoC) architectures are a promising solution to complex on-chip communication problems [1]. Although NoC addresses some problems, e.g. scalability, the need for complex and multistage pipelined routers, however, results in a high router-to-link energy/delay ratio and increases the delay and energy of communication. Circuit-switching has been used in a number of NoC architectures to reduce on-chip communication latency, when compared to packet-switched networks, since packets need not go through routing and arbitration at each hop once circuits are set up. However, by increasing the network size, the contention among circuits increases. Since there are no buffers to store the packets in the intermediate switches, this contention may result in a long message latency and low resource utilization [2].

In this work, we develop a packet-switched NoC architecture which can also benefit from the low-power and high-performance communication provided by circuit-switching. To this end, the network resources are divided between a circuit-switched and a packet-switched network. The traffic is split between the two networks in such a way that the average packet latency and power consumption of the NoC is reduced. The first sub-network, called Pnet, directs the packets using the traditional packet-switching scheme. The second sub-network, called Cnet, on the other hand, is devoted to establish circuits in order to forward packets to some node along the route towards their destinations (and not necessarily the destination node) by bypassing some intermediate routers. Establishing the circuits for requesting nodes is done by a light-weight setup-network, called Snet. This network receives the requests in each cycle, arbitrates between them, and establishes a circuit for each requesting packet on the Cnet. Using a novel architecture based on reconfigurable mesh [3] and taking advantage of ultra low-latency on-chip interconnects, this network can construct the circuits within a single cycle of the data network.

To keep the cost of the proposed NoC equal to a traditional packet-switched one, the NoC links and resources are divided between the two sub-networks using Spatial-Division Multiplexing (SDM). SDM is an effective alternative for Time-Division Multiplexing (TDM) which is the dominant scheme for sharing network links among several circuits in circuit-switched NoCs. Unlike TDM in which at each time slot, all the wires of a link are dedicated to transmission of data from a single source, the SDM technique allocates a sub-set of the link wires to a given circuit for the whole connection lifetime. Authors in [4] showed that SDM is a more power-efficient and less complex alternative to TDM in circuit-switched networks.

In this paper, we first analyze the effectiveness of SDM for dividing the network resources between the sub-networks and then, propose an NoC architecture that benefits from the services provided by both switching methods.

The rest of the paper is organized as follows. Section 2 introduces the basic concepts and the structure of the proposed NoC. The structure of the setup-network and its design issues is discussed in Section 3. Section 4 evaluates the proposed NoC architecture. Finally, Section 5 concludes this paper.

II. THE PROPOSED NETWORK ARCHITECTURE

A. Basic Concepts

As mentioned in Section I, in this work, we propose to integrate circuit-switching into a packet-switched network by using an SDM strategy in order to benefit from both switching methods. Enhancing the power and performance metrics of packet-switched NoCs by integrating a second switching

978-3-9810801-5-5/DATE09 © 2009 EDAA

mechanism has been addressed in several previous work, such as express virtual channels [5] and long range links [6].

Applying SDM in a packet-switched network allows having several links in parallel in the same direction and therefore, increasing the number of possible paths in that direction. This increase in the path diversity reduces the *Head of Line (HOL) Blocking* and can improve the average latency and throughput of the network. On the other hand, splitting the links into some sub-links will increase the average packet latency as the number of flits of a packet is increased.

An in-depth analysis of the effect of splitting the wide links into smaller parallel links on the NoC latency and throughput is presented in [7], where it has been shown that splitting the links into two sub-links increases the throughput by 50-60% with a negligible effect on the average message latency.

As a result, SDM is a suitable division scheme for our proposed NoC, since in addition to throughput, by applying circuit-switching in one of the sub-networks, we can also improve the power and latency of the NoC.

The circuit-switched network (Cnet) consists of a switch which can connect any input port to any output in order to bypass some intermediate nodes and produce a shorter distance to the destination. Cnet circuits are constructed using a light-weight and high-speed setup-network. At each node, the routers split the traffic between these two sub-networks in such a way that the power and performance metrics of the NoC is improved.

B. The Router Architecture

In this section, we introduce the architecture of the router in our design. This architecture is shown in Fig. 1, where every *n*-bit link of the original NoC has been split into two parallel sub-links of width *n/2*. The first sub-links (the upper wires in this figure) are allocated to Pnet and the other part is devoted to Cnet. The number of buffers has been doubled, but the total memory in the router remains the same as the size of these buffers is half of the original system. The crossbar switch structure is the same as in a conventional packet-switched router. However, the switch allocator (arbiter) is divided into two separate allocators. The first one which acts as a conventional arbiter handles the sub-links belonging to Pnet. The other allocator handles the sub-links of Cnet and connects one of the Cnet inputs to the output port. This unit is implemented by a register indicating whether the corresponding output port is a part of a circuit and which input port should be connected to it. The value of this register is set by Snet during the circuit establishment process. We have modified some stages of a conventional packet-switched router with respect to the new features of this architecture [8], as follows.

A header flit, on arriving at an input port, is first buffered in the buffer write (BW) pipeline stage. The flit is assigned a buffer slot at P_Buff in Fig. 1 if it is arrived from Pnet, and at C_Buff if it is arrived from Cnet, and the current node is the end node of its circuit. In this case, the demultiplexer at the input port is configured to direct the flit to C_Buff.

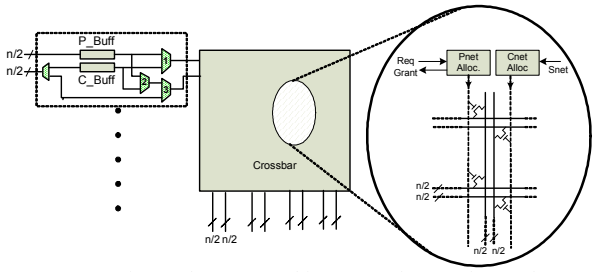

Fig. 1. The router architecture and a sample crossbar output

In the next stage, the routing logic performs route computation (RC) to determine the output port for the packet. The header then arbitrates for both Pnet partition and Cnet partition of the determined output port in the next stage (equivalent to the VC allocation (VA) stage in conventional packet-switched routers). At this stage, the arbiter associated with each output port allocates the Cnet part of the corresponding output port to one of the requesting flits having the longest distance to its destination. This involves calculating the distance between the current node and the destination of each flit at the RC stage. Cnet is allocated if it is free, i.e. it is not a part of a currently established circuit. At the same time, the Pnet part of the output port is allocated to one of the other flits according to a round-robin policy.

Upon successful allocation of Pnet, the next stages are followed just like in the conventional packet-switched routers. The header flit proceeds to the switch allocation (SA), switch traversal (ST), and link traversal (LT) to travel to the next node. Body and tail flits follow a similar pipeline except that they do not go through RC and VA stages. The tail flit, on leaving the router, releases the switch output reserved by the header.

However, if the header flit is decided to be transmitted over the Cnet, the allocator appropriately configures multiplexers 2 and 3 in Fig. 1 in order to connect the buffer to Cnet. At the same time, the header flit sends a request along with its destination address to Snet. Using the structure described in the next section, Snet finds and sets a circuit for the packet during this stage. To this end, Snet configures input port (by appropriately setting the control lines of the demultiplexer and multiplexer 3 in Fig. 1) and crossbar connection (by setting the register of the designated output port in Fig. 1) of every router along the circuit, in such a way that a direct link is established between the current node and the circuit destination node over the crossbars and the links of Cnet. The destination of this circuit may not be the packet destination, but an intermediate node along the path towards the destination. We embed a 1-flit buffer at each input port of Cnet. The buffers provide pipelining over the circuits and also act as a repeater. As a result, the header and consequent flits are sent over the established circuit in a pipelined fashion and gets buffered (in C_Buff in Fig. 1) at the circuit end node. At this node, the header flit needs to go through the router pipeline stages again and this process continues until the packet is delivered at the destination node. A packet may switch between Pnet and Cnet several times to reach its destination.

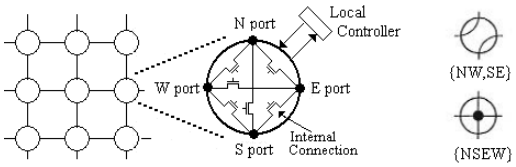

Fig. 2. The internal structure of reconfigurable mesh nodes and two sample node configurations

III. SETUP-NETOWRK STRUCTURE

The setup-network (Snet) is designed based on the reconfigurable mesh presented in [3], where a large number of algorithms are implemented using this interesting structure. Fig. 2 shows a reconfigurable mesh where each node has four ports N, S, E, and W, through which it connects to processors (if any) at North, South, East, and West. Each node is simply composed of some switches that can establish internal connections among its ports. In addition, each node has some simple controller logic. The controller can read and write the ports and configure the switches. Two possible node configurations are displayed in Fig. 2 (at the right side). A node can change its internal connections at each step.

Snet has the same topology and size as the data network. Each node in Snet corresponds to a node in the data-network with the same address. To avoid conflict among the setup process of the circuits spanning in opposite directions in each dimension, each row and column of the reconfigurable mesh has two separate set of links each of which handles requests received from a single direction. Each reconfigurable mesh link is ($log(n)$ + 1)-bit wide in an $n \times n$ NoC. We develop an algorithm based on this architecture that, at each cycle, receives a set of circuit construction requests from Cnet nodes and finds the longest possible circuit for each requesting node through the free Cnet links. The circuits then are set up on Cnet by appropriately configuring the input ports and crossbar switches.

The Snet clock cycle must be set in such a way that, during each cycle, the sent data propagates along each direction and the switch configurations become stable. Unlike traditional off-chip interconnections, 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 properties [9]. Authors in [10], for example, exploited wire properties to design an ultra low-latency global control network in on-chip networks. However, when the network size is increased, the desired clock frequency may not be achieved with an acceptable power and area overhead. To increases the scalability of the proposed NoC architecture, we restrict circuits established by a node to span within its fixed local neighborhood of $d\psi$hops. d is a design parameter which offers a trade-off between the energy consumption and area of Snet and the performance of Cnet. More precisely, a faster Snet is able to construct much longer circuits at the price of more area and power due to wider wires and more repeaters.

Fig. 3 depicts a setup-network with four requests A, B, C, and D. Sx and Dx denote source and destination nodes of request x, where x = A, B, C, D. At the initial state, the Snet nodes form independent buses at each row and column, by connecting the W port to the E port and the N port to the S port. The local controller is also connected to the bus and can read and write data. Each flit wining a Cnet output port sends a request to the same port at the corresponding node in Snet, containing the address of its destination. Based on the output port, Snet selects one of the reconfigurable mesh buses for circuit setup operation.

We describe the circuit setup process for request A to set up a circuit from port E of node S_A towards D_A. Without loss of generality, we assume that d equals the network size. when d is less than the network size, if the distance between the source and destination nodes in dimension i is greater than d, the coordinate of the circuit end point at that dimension is replaced with S_i+d (or S_i-d based on the circuit direction), where S_i is the coordinate of the source node at the ith dimension. The algorithm involves some steps as follows. Due to limited space, we remove some details of the algorithm.

1. S_A disconnects its E and W ports to block the requests received from West (hence, avoid conflict between requests) and writes the y-coordinate of the destination address to its E port. The address is propagated on the bus along the E direction. The node with the same y coordinate detects the address and disconnects its W port and E port to form a bus-segment between S_A and it. However, this bus-segment may be shorter, if one of the nodes between these two nodes initiates circuit setup operation in the same direction and at the same cycle (and disconnects its W and E ports). For example, B is blocked by A at S_A. The Snet ports corresponding to the ports of the currently established circuits in Cnet are also disconnected and block the propagation of address.

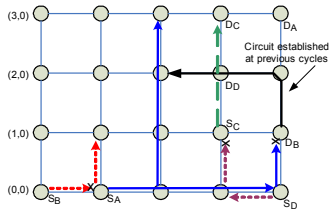

Fig. 3. Snet circuit-setup process for 4 requests.

2. S_A, at this step, broadcasts the x-coordinate of its destination address on its bus-segment. The nodes belonging to the bus-segment receive the address and expand the current bus-segment along the appropriate direction (based on the destination address) just like the previous step. At this step, a conflict may occur between the requests spanning in opposite directions in the same column or row, when they have overlapping bus-segments and try to expand it along the same direction. Based on a simple arbitration policy, on each row, the requests on the E-to-W bus win at the nodes with odd x and the requests on the other bus win at the nodes with even x (e.g., request D is prioritized over A at node (0,3)). The same policy is applied in each column. The bus-segments formed for the requests are depicted in Fig. 3. The solid line represent the bus-segment of request A (the branch along the x dimension is formed at the first step and the branches along the y dimension are added at the second step).
3. Each node at the end of a bus-segment branch sends its address (which denote the branch length) to the branch root (e.g., in request A, node (3,2) sends its y-coordinate to node

(0,2)). Afterwards, the nodes within the row or column where the circuit setup process is initiated find the path with maximum length between the source and one of the bus-segment nodes (e.g. node (3,2) for request A; it may also be the packet destination, e.g. node (3,3) for request C) and configure their switches to form this path. The source, afterwards, sends a signal on the constructed circuit. Each Snet node which receives this signal configures the Cnet resources to construct the same circuit in the Cnet.

This mechanism guarantees that a circuit is constructed for each requesting node, at least between it and its neighboring node at the specific direction. So, at the worst case, Cnet forwards a packet to its neighboring node, as done by a conventional packet-switched NoC. In this case, the control signals of Snet will also travel a short distance, resulting in negligible power overhead.

IV. Evaluation Results

To evaluate the proposed on-chip communication scheme, we have simulated it using Xmulator, a fully parameterized simulator for interconnection networks [11]. The simulator is augmented with the Orion power library [12] to calculate the power consumption of the networks. Simulations experiments are performed for a 5×5, 128-bit wide system with 8-flit packets. Moreover, the process feature size and working frequency of the routers is set to 70nm and 250 MHz, respectively, in the Orion library. Messages are generated according to a Poisson distribution and sent to randomly chosen destinations.

Fig. 4 displays the average message latency and power consumption (static and dynamic) of the proposed NoC and a conventional packet-switched NoC [8]. Our method reduces the power and latency of communication by 22% and 45%, respectively, on average. It also increases the network throughput as a result of SDM, as discussed earlier. Early evaluation results for the area of this architecture using Orion library and also the analysis in [13] shows less than 10% area overhead (mainly due to the setup-network and multiplexers added to input ports).

V. Conclusion and Future Work

In this paper, we proposed an on-chip communication mechanism by integrating a packet-switched and a circuit-switched network into a single architecture. The packet-switched part provides traditional packet-switching while the circuit-switched sub-network directs packets over circuits which bypass some intermediate nodes. The circuits are constructed using a light-weight and low-latency setup-network. Early evaluation result showed a promising power and performance improvement, compared to a conventional packet-switched NoC. For future work, we will implement the setup-network by tuning the wire and repeater sizes and then, we can accurately calculate the power consumption and area of this network and explore the trade-off between the Snet overheads and maximum length of the Cnet circuits. As early evaluation results indicated, since the setup-network has a small bit-width and very simple node structure, implementing it by wider wires will not greatly affect the NoC power consumption and area.

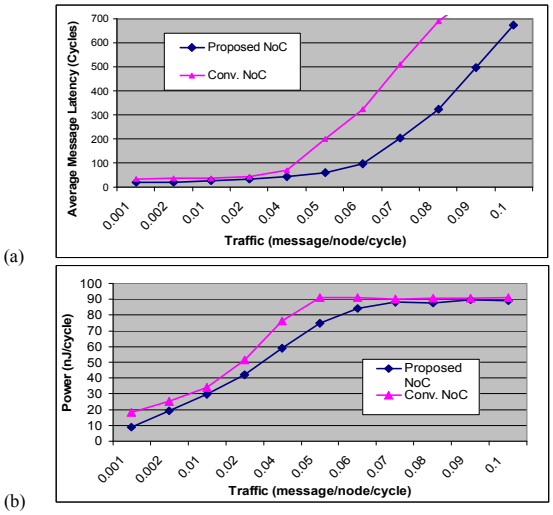

Fig. 4. (a) The latency (cycles) and (b) power (nJ/cycle) for the proposed and a conventional NoC

Out-of-order delivery of packets belonging to a single message is an important issue that must be addressed. Finally, this NoC architecture must be evaluated using other traffic patterns, especially the traffic of realistic applications. This involves developing a mapping algorithm based on the characteristics of this specific NoC architecture. In addition to the common mapping objectives, this algorithm should reduce the conflict among the circuits in the circuit-switched sub-network.

References

[1] L. Benini, and G. De Micheli. "Networks on Chip: a New Paradigm for Systems on Chip Design", in *IEEE Computer,*35(1), 2002.

[2] K. Chang et al., "A Low-Power Crossroad Switch Architecture and Its Core Placement for Network-On-Chip", in *Proc. of ISLPED, 2005.*

[3] R. Vaidyanathan, J. Trahan, *Dynamic Reconfiguration: Architectures and Algorithms*, Springer, 2004.

[4] A. Leroy, et al., "Spatial Division Multiplexing: a Novel Approach for Guaranteed Throughput on NoCs", *in Proc. CODES+ISSS*, 2005.

[5] A. Kumar, et al. "Express Virtual Channels: Towards the Ideal Interconnection Fabric", in *Proc. of the 34th ISCA*, 2007.

[6] U. Ogras, et al., "Application-Specific Network-on- Chip Architecture Customization via Long-Range Link Insertion", *in Proc. of DAC,* 2005.

[7] C. Gomez, et al., "Exploiting Wiring Resources on Interconnection Network: Increasing Path Diversity", *in Proc of the 16th Euromicro PDP*, 2008.

[8] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks,* Morgan Kaufmann Publishers, 2004.

[9] R. Balasubramonian, et al., "Microarchitectural Wire Management for Performance and Power in Partitioned Architectures", in *Proc. of HPCA-11*, 2005.

[10] T. Krishna, et al.," NoC with Near-Ideal Express Virtual Channels Using Global-Line Communication", in *Proc. Symposium on High Performance Interconnects*, 2008.

[11] Xmulator NoC Simulator: www.xmulator.org, 2008.

[12] H. Wang, et al., "Orion: A Power-Performance Simulator for Interconnection Networks", in *Proc. of the 35th. MICRO*, 2002.

[13] J. Balfour, W.J. Dally, "Design Tradeoffs for Tiled CMP On-Chip Networks", in *Proc. of ICS*, 2006.