

An Efficient Way to Increase Performance by Using Low Power Reconfigurable Routers

Noopur Sharma¹, Shivayya Gadag²

¹(Telecommunication Engg, MVJ College of Engg Bangalore/ VTU Belgaum, India)

²(Telecommunication Engg, MVJ College of Engg Bangalore/ VTU Belgaum, India)

Abstract: In this paper, the advantage of the use of an NoC with reconfigurable Routers has been presented instead of homogeneous ones. . In this paper we propose the use of a reconfigurable router, where the buffer slots are dynamically allocated to increase router efficiency in an NoC, even under rather different communication loads. In the proposed architecture, the depth of each buffer word used in the input channels of the routers can be reconfigured at run time. Using reconfiguration, one can dynamically change the buffer depth to each channel, in accordance to the necessity of the application, increasing the power efficiency of the system for the same performance level. The reconfigurable router allows up to 52% power savings, while maintaining the same performance as that of a homogeneous router, but using a 64% smaller buffer size.

Keywords: Buffer, latency, network-on-chip, power consumption, reconfigurable router.

I. Introduction

Network-on-chip (NoC) designs are based on a compromise among latency, power dissipation, or energy, and the balance is usually defined at design time. In NOC technology Cores Communicates With Each Other Using NoC . NoC Consists of Routers (R) and Network Interfaces (NI).One or More Cores Connected to a NI.

MULTIPROCESSOR SYSTEM-ON-CHIPS (MPSoCs) are emerging as one of the technologies providing a way to support the growing design complexity of embedded systems, since they provide processor architectures adapted to selected problem classes, allied to programming flexibility. To ensure flexibility and performance, future MPSoCs will combine several types of processor cores and data memory units of widely different sizes, leading to a very heterogeneous architecture.

The increasing interconnection complexity and the known scalability deficiency of buses require another model of inter- connection. The communication among cores of an MPSoC having reusable and scalable interconnections is being provided by networks-on-chip (NoCs) [1]. NoCs have been proposed to integrate several Intellectual Property (IP) cores, providing high communication bandwidth and parallelism.

Azimi *et al.* [2] affirm that it is necessary to find a way to keep the off-die bandwidth manageable in system architectures with tradeoffs among cost, power, and performance. Moreover, in a hardware context, the system must offer flexibility with high- bandwidth, low-latency, and power-efficiency. Interconnection fabric allows cores to access memory, communicate with each other and with the rest of the system.

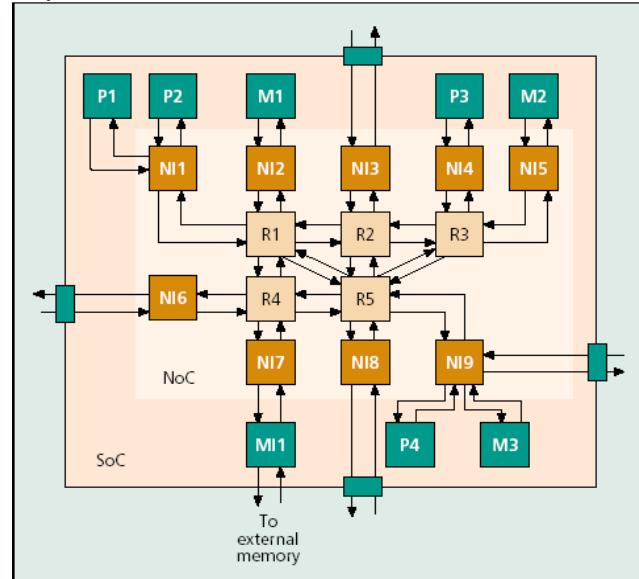


Fig-1 NOC Topology

Manferdelli *et al.* [3] state that to guarantee the increase in performance of general purpose CPUs, one needs to use massive parallel computing. For this, more independent CPUs, bigger caches, and more independent memory controllers have been used, and it is possible to find many applications that use heterogeneous processors with several memory controllers to provide a large memory interface. One can find an example of such architecture on the Xbox360 [4]. Fig. 2(a) shows a system block diagram of the Xbox, a platform with several cores, with each core having a specific throughput and bandwidth.

Another example of mixed communication behaviour requirements is showed in the Fig. 2(b). In accordance with [3], there is a clear difference between traffic among cores in a SoC with out-of-order cores (OoCs) and in-order-cores (IoCs). OoCs are larger and have worse power performance than IoCs. Besides, there is more communication among IoCs than among OoCs, thus the former need to have different interconnection characteristics among them, in order to guarantee a higher communication bandwidth among IoC devices, since their communication with OoCs occurs on a much smaller scale.

In an NoC, several items can vary from design to design, like depth of first-input–first-output (FIFO) buffers, router topology, switch and arbiter [5]. In this manner, decisions such as throughput, latency and bandwidth are defined as a modification of the NoC architecture, most of the times, being

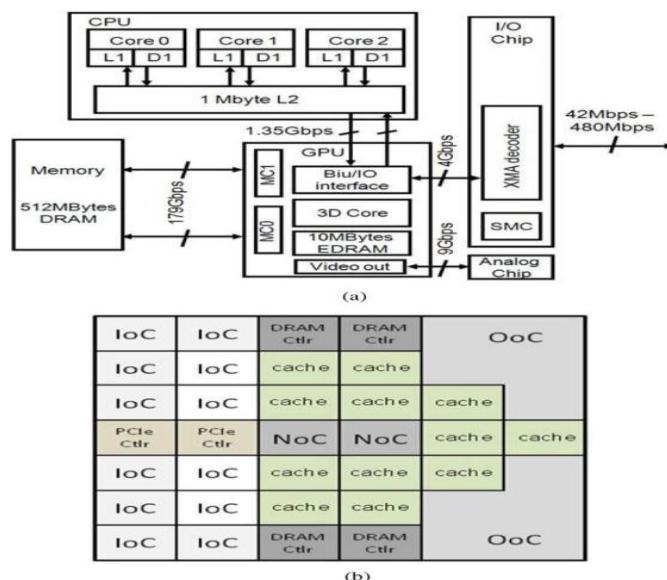


Fig. 2. (a) Xbox 360 block diagram. (b) Example of different cores used in an SoC.

made at design time, trying to guarantee the performance of the system. However, whenever the product needs an update or has to change its functionality, most likely a huge change in the communication pattern will be observed, and hence decisions performed at design time would mean either a loss in performance, or excessive power dissipation.

Considering the NoC components, as crossbars, arbiters, buffers, and links, in the experiments realized by [6] the buffers were the largest leakage power consumers, dissipating approximately 64% of the whole power budget. In this way, the buffers were considered as candidates for leakage power optimization, since even at high loads, there were still 85% of idle buffers [6]. Regarding dynamic power, the buffers' consumption is also high, and it increases rapidly as the packet flow throughput increases [7].

Our particular contribution aims at providing the router with a certain amount of reconfiguration logic, allowing changes in the amount of buffer utilization in each input channel, in conformity with the communication needs. The principle is that each input channel can lend/borrow buffer units to/from neighbouring channels in order to obtain a determined bandwidth. When a channel does not need its entire available buffer, it can lend buffer word slots to neighbouring channels. Results show the inefficiency in the amount of buffers used within a homogeneous router, and the gains that can be achieved using the proposed strategy. We focus on providing a reconfigurable router that can optimize power and improve energy usage while sustaining high performance, even when the application changes the communication pattern. Moreover, experiments compare favourably with other dynamic topologies like virtual channels.

II. Proposed Router Architecture

A. Original Router Architecture

The proposed router architecture was embedded in the SoCIN NoC. SoCIN has a regular 2-D-mesh topology and parametric router architecture. The router architecture used is RaSoC, which is a routing switch

with up to five bi-directional ports (Local, North, South, West, and East), each port with two unidirectional channels and each router connected to four neighbouring routers (North, South, West, and East). This router is a VHDL soft-core, parameterized in three dimensions: communication channels width, input buffers depth, and routing information width.

The architecture uses the wormhole switching approach and a deterministic source-based routing algorithm. The routing algorithm used is XY-routing, capable of supporting deadlock-free data transmission, and the flow control is based on the hand-shake protocol. The wormhole strategy breaks a packet into multiple flow control units called flits, and they are sized as an integral multiple of the channel width. The first flit is a header with destination address followed by a set of payload flits and a tail flit. To indicate this information (header, payload, and tail flits) two bits of each flit are used. There is a round-robin arbiter at each output channel. The buffering is present only at the input channel

B. Reconfigurable Router Architecture

If an NoC's router has a larger FIFO buffer, the throughput will be larger and the latency in the network smaller, since it will have fewer flits stagnant on the network. Nevertheless, there is a limit on the increase of the FIFO depth. Since each communication will have its peculiarities, sizing the FIFO for the worst case communication scenario will compromise not only the routing area, but power as well [6]. However, if the router has a small FIFO depth, the latency will be larger, and quality of service (QoS) can be compromised. The proposed solution is to have a heterogeneous router, in which each channel can have a different buffer size. In this situation, if a channel has a communication rate smaller than its neighbour, it may lend some of its buffer slots that are not being used. In a different communication pattern, the roles may be reversed or changed at run time, without a redesign step.

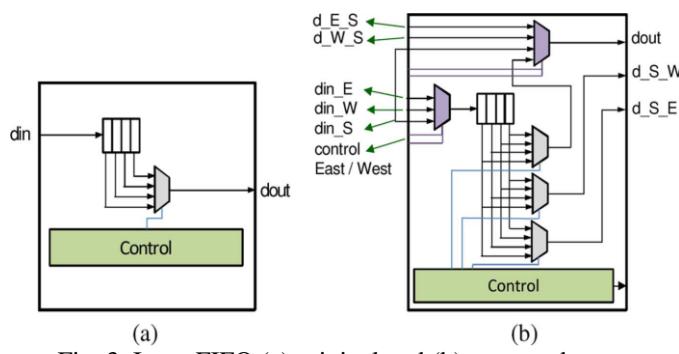


Fig. 3. Input FIFO (a) original and (b) proposed router.

The proposed architecture is able to sustain performance due to the fact that, statistically, not all buffers are used all the time. In our architecture it is possible to dynamically reconfigure different buffer depths for each channel. A channel can lend part or the whole of its buffer slots in accordance with the requirements of the neighbouring buffers. To reduce connection costs, each channel may only use the available buffer slots of its right and left neighbour channels. This way, each channel may have up to three times more buffer slots than its original buffer with the size defined at design time. Fig. 3 shows the original and proposed input FIFO. Comparing the two architectures, the new proposal uses more multiplexers to allow the reconfiguration process. Fig. 3(b) presents the South Channel as an example. In this architecture it is possible to dynamically configure different buffer depths for the channels. In accordance with this figure, each channel has five multiplexers, and two of these multiplexers are responsible to control the input and output of data. These multiplexers present a fixed size, being independent of the buffer size. Other three multiplexers are necessary to control the read and write process of the FIFO. The size of the multiplexers that control the buffer slots increases according to the depth of the buffer. These multiplexers are controlled by the FSM of the FIFO. In order to reduce routing and extra multiplexers, we adopted the strategy of changing the control part of each channel.

Some rules were defined in order to enable the use of buffers from one channel by other adjacent channels. When a channel fills all its FIFO it can borrow more buffer words from its neighbours. First the channel asks for buffer words to the right neighbour, and if it still needs more buffers, it tries to borrow from the left neighbour FIFO. In this manner, some signals of each channel must be sent for the neighbouring channels in order to control its stored flits.

In result, each channel needs to know how many buffer words it uses of its own channel and of the neighbouring channels, and also how much the neighbour channels occupy of its own buffer set. A control block informs this number. Then, based on this information, each channel controls the storage of its flits. These flits can be stored on its buffer this information, each channel controls the storage of its flits. These flits can be

stored on its buffer slots or in the neighbour channel buffer slots. Each input port has a control to store the flits and this control is based in pointers. Each input channel needs six pointers to control the read and writing process: two pointers to control its own buffer slots, two pointers to control the left neighbour buffer slots, and two more pointers to control the right neighbour buffer slots (in each case, one pointer to the read operation and one pointer to write operation).

In this design, we are not considering the possibility of the Local Channel using neighbouring buffers, only the South, North, West, and East Channel of a router can make the use of their adjacent neighbours.

As mentioned before, the loan granularity used in this proposal is a buffer slot. The area results of the reconfigurable router would not present a significant change if loan granularity was increased. This is due to the fact that the control overhead is defined mainly by the FIFO's control circuit. As the buffers are implemented using circular FIFOs, the FIFO pointers are incremented to each new slot, and this control will be the same whatever the used loan granularity. If we increase the loan granularity to more than one slot, then the loss in performance could be large, and the reduction in area or power would be minimal. In addition, we are considering sharing of the buffer slots only among adjacent channels. This decision is based on the costs of interconnections, multiplexers, and logic to control the combination of all loans among all input channels. Consequently, the area and power consumption would be much larger if we consider the last case, and the gains in performance would not be large enough to compensate this extra cost.

Fig. 4 shows the channel of Fig. 3(b) organized to constitute the reconfigurable router. Each channel can receive three data inputs. Let us consider the South Channel as an example, having the following inputs: the own input (din S), the right neighbour input (din E), and the left neighbour input (din W). For illustration purposes, let us assume we are using a router with buffer depth equal to 4, and there is a router that needs to be configured as follows: South Channel with buffer depth equal to 9, East Channel with buffer depth equal to 2, West Channel with buffer depth equal to 1, and North Channel with buffer depth equal to 4. In such case, the South Channel needs to borrow buffer slots from its neighbours. As the East Channel occupies two of its four slots, this channel can lend two slots to its neighbour, but even then, the South Channel still needs more three buffer slots. As the West Channel occupies only one slot, the three missing slots can be lent to the South Channel. When the South Channel has a flit stored in the East Channel, and this flit must be sent to the output, it is passed from the East Channel to the South Channel (d_E_S), and so the flit is directly sent to the output of the South Channel (dout S) by a multiplexer. The South Channel has the following outputs: the own output (dout S) and two more outputs (d_S_E and d_S_W) to send the flits stored in its channel but belonging to neighbour channels.

The choice to resend the flits stored in a neighbour channels to its own channel before sending them to the output was preferred in order to avoid changes in others mechanisms of the architecture. In this manner we did not change the routing algorithm, avoiding the possibility of data deadlock, since the NoC continues using routing, which is intrinsically deadlock free. With this definition, the complexity of the implementation to obtain the correct function of the router was reduced in this aspect. Each flit stored in a neighbour channel returns to the respective channel when it needs to be sent to an output channel. In this case, when an input channel is connected to an output channel, the flits are sent one-by-one, and the pointers are updated as each flit is sent.

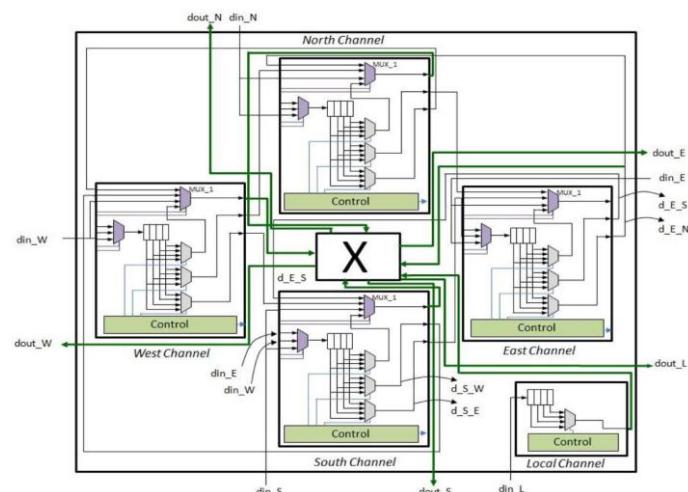


Fig. 4. Proposed router architecture

As each channel knows how many buffer slots it has allocated, when the pointers present an address belonging to a neighbour buffer slot, the control of the first multiplexer of Fig. 5(b) allows the sending of the respective flits to the output of its channel. As we do not change the routing policy, there is no possibility of entering a

deadlock situation. Of course, one could be concerned about one channel asking buffers from another channel which is also asking for buffers. Since only the neighbours are asked about lending/borrowing, no cycle can be made, and hence at the circuit level there is also no possibility of deadlock.

Fig. 5 shows an example of the reconfiguration in a router according to a needed bandwidth in each channel. First, a buffer depth for all channels is decided at design time, in this case, we defined the buffer size equal to 4, as illustrated in Fig. 5(a). After this, the traffic in each channel is verified and a control defines the buffer depth needed in each link to attend to this flow, as shown in Fig. 5(b).

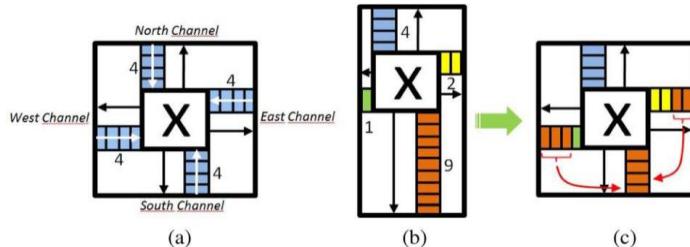


Fig. 5. (a) Router designed with FIFO depth 4; (b) One example of need of configuration of the router; (c) Reconfiguration of the buffers to attend the need.

The distribution of the buffer words among the neighbour channels is realized as shown in Fig. 5(c). Meanwhile, the buffer physical disposition in each channel correspondent the FIFO depth initially defined, as shown in Fig. 5(a), but the allocation of buffer slots among the channels can be changed at run time, as exemplified in Fig. 5(c).

Our proposal consists of reconfiguring the channel according to the availability of buffers in the channels. If a new channel depth is required, the buffer depth is updated slot by slot, and this change is made whenever a buffer slot is free. For the set of benchmarks used in this work, and as reported in many related works, whenever the application is changed, a different bandwidth is required among the channels. The reconfigurable router can change its depth in only few cycles, which means a very small performance overhead. Moreover, as each core sends packets at a different rate, the reconfiguration of the router was implemented considering that in some possible interval among packets there would be a time-slack. As the traffic is composed of packets, the buffers are not used 100% of the time in all parts of the network.

III. Results

1. AREA, POWER AND FREQUENCY RESULTS

The proposed router was described in Verilog HDL, and we used the ModelSim tool to simulate the code. We analyzed the average power consumption, area, and frequency results to a Spartan 3AN FPGA kit on Xilinx tool. The power results were obtained with a 201.126MHz frequency in architecture. Table I presents the results obtained for this configuration. The channel width contains bits, for data bits and 2 bits for control. As shown in Figure. 5.2, for the same average latency, using a reconfigurable router with buffer depth 4, different buffer sizes were needed while using the homogeneous router to the four applications mentioned. For this reason, we demonstrate the gains obtained in the synthesis results when one considers the same average latency results for a homogeneous and reconfigurable router (meanwhile, for each benchmark the homogeneous router needs to use different buffer sizes when compared with the reconfigurable router).

In a router, the largest power dissipation comes from the flipflops of the buffers. By using the proposed reconfiguration, and using extra multiplexers, it has been possible to reduce the total number of required flip-flops. One can still obtain power reduction because the multiplexers present less power consumption than flip-flops. A flip-flop dissipates power even when no data changes at its input, since the clock is always switching. Notice that thanks to the high activity present in the routers, clock gating is a less affective technique.

Hence, the larger the buffers, the larger are the power dissipation in the flip-flops. Using the homogeneous router, in order to sustain the same performance that the proposed buffer scheme can achieve, a buffer of fixed size would have to be much larger, with many more flip-flops. Thus, the proposed router reduces power considering the same performance, once it uses much smaller buffers, hence, less flip-flops and less power dissipation. In electronics, a flip-flop or latch is a circuit that has two stable states and can be used to store state information. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems.

Flip-flops and latches are used as data storage elements. Such data storage can be used for storage of state, and such a circuit is described as sequential logic. When used in a finite-state machine, the output and next state depend not only on its current input, but also on its current state (and hence, previous inputs). It can also be

used for counting of pulses, and for synchronizing variably-timed input signals to some reference timing signal. In electronics, a multiplexer (or MUX) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2^n inputs has n select lines, which are used to select which input line to send to the output. Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth. A multiplexer is also called a data selector. They are used in CCTV, and almost every business that has CCTV fitted, will own one of these.

The reason for the power increase being a nonlinear function of the buffer depth is due to the utilization of multiplexers, as present in Figure. 5.1(a). These multiplexers define which flit of the FIFO must be sent to channel output. Hence, the gate increase of the FIFO depth 7 to 9 is lower than from the FIFO depth 9 to 11. With the applications simulated in this work, we confirmed that the original homogeneous NOC presents a large under utilization of the router, since not all of its channels are used. In such cases, the extra buffer words on channels not used in the original router would be unnecessarily consuming power. When the channel width is defined equal to 18 bits, the proposed router does not present penalties in the maximum frequency compared with the homogenous router designed with the same latency.

Besides, for the same performance results, the reconfigurable router presents a great reduction in power dissipation. The larger the link size, the larger the power savings allowed by the reconfigurable router, since in this case the impact of the extra circuits required to allow reconfiguration are amortized.

RECONFIGURABLE ROUTER				
CHANNEL WIDTH	BUFFER DEPTH	POWER CONSUMPTION	AREA	FREQUENCY
32 BITS	4	19MW	21%	201.126MHZ

TABLE : Frequency, power, area Result

Considering the four applications utilized here, the reconfigurable router reduces the power consumption, and to the same performance results, it uses smaller buffer depths. Besides, another advantage in the use of the NoC with the reconfigurable router instead of the homogeneous router is that one can dynamically change the buffer depth to each channel, in accordance with the necessity of the application.

Thus, we can conclude that the obtained results emphasize the fact that the proposed NoC router does not degrade the system performance, and can save power. With the proposed router it is possible to have one single NoC connecting different applications that might change their communicating patterns at run time. In the same way, this architecture allows application updates without compromising the performance of the system. Meanwhile, if a homogeneous router had been used in these situations, design modifications at design time would have had to be made to achieve the optimum case. In such case, one would need to redesign the homogeneous NoC to set buffer sizes and position of the cores in the network. The technique here proposed avoids costly redesigns and new manufacturing.

IV. Conclusion

We verified that to reach the same performance obtained with the reconfigurable router the original architecture needs many more buffers. The new router, while reaching the same performance than the original architecture, obtained a reduction of approximately 25% of power consumption in the worst case, and of 52% for the best case analyzed. Moreover, with the new architecture it is possible to reconfigure the router in accordance with the application, obtaining similar performances even when the application radically changes. When compared with the ViChar architecture, our proposal obtains 78% of power reduction for the same configuration. Moreover, the reconfigurable router obtains the same performance of the homogeneous router with a buffer depth of 64% smaller. Moreover, with the new architecture it is possible to reconfigure the router in accordance with the application, obtaining similar performances even when the application radically changes.

References

- [1] L. Benini and G. De Micheli, "Network on chips: A new SoC paradigm," IEEE Computer, vol. 35, no. 1, pp. 70–78, Jan. 2002.
- [2] M. Azimi, N. Cherukuri, D. N. Jayasimha, A. Kumar, P. Kundu, S. Park, I. Schoinas, and A. S. Vaidya, "Integration challenges and tradeoff for tera-scale architectures," Intel Technol. J., vol. 11, no. 3, Aug. 2007.
- [3] L. Manferdelli, N. K. Govindaraju, and C. Crall, "Challenges and opportunities in many-core computing," Proc. IEEE, vol. 96, no. 5, pp. 808–815, May 2008.
- [4] J. Andrews and N. Baker, "Xbox 360 system architecture," IEEE Micro, vol. 26, no. 2, pp. 25–37, Mar./Apr. 2006.
- [5] M. Vestias and H. Neto, "Area and performance optimization of a generic network-on-chip architecture," in Proc. Symp. Integr. Circuits Syst. Des. (SBCCI), 2006, pp. 68–73.
- [6] C. Xuning and L. Peh, "Leakage power modeling and optimization in interconnection networks," in Proc. Int. Symp. Low Power Electron. Des. (ISLPED), 2003, pp. 90–95.
- [7] T. Benini and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in Proc. 39th Des. Autom. Conf. (DAC), 2002, pp. 524–529