International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 9 2963 – 2968

VERILOG Implementation of Reconfigurable Routers for Low Power

D. Manaswi[1]
Roll NO: 12jm1d5710
M.Tech in VLSI Design
GDMM College of Engineering & Technology
Nandigama
manaswi.dokku@gmail.com

P. Koteswara Rao[2]
Assos Prof
Head of ECE Dept
GDMM College of Engineering & Technology
Nandigama
hodecegdmm@gmail.com

***Abstract*--**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. However, setting all parameters, such as buffer size, at design time can cause either excessive power dissipation (originated by router under utilization), or a higher latency. The situation worsens whenever the application changes its communication pattern, e.g., a portable phone downloads a new service. Large buffer sizes can ensure performance during the execution of different applications, but unfortunately, these same buffers are mainly responsible for the router total power dissipation. Another aspect is that by sizing buffers for the worst case latency incurs extra dissipation for the mean case, which is much more frequent. 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. 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 (NOC), Power Dissipation, Reconfigurable Router.

******

1.INTRODUCTION:

The VLSI design has taken a new trend in the present technology. In 1965 Gordon Moore shown that for many MOS transistors technology, there exists a minimum cost that maximizes the number of components per integrated circuits. He also predicted that as transistor dimensions shrunk from one technology generation to the next, the minimal cost point allows doubling the number of transistors every two to three years. This trend has been sustained and expected to be maintained well into the first 20 years of the century.

It‘s a process that means to create integrated circuits by combining thousands of transistor-based circuits into a single chip. VLSI finds immediate application in DSP, Communications, Microwave and RF, MEMS, Cryptography, Consumer Electronics, Automobiles, Space Applications, Robotics, and Health industry. Nearly all modern chips employ VLSI architectures, or ULSI (ultra large scale integration). The line that demarcates VLSI from ULSI is very thin . When a technology matures, it leads to a paradigm shift in system scope. Shown here is the chip scope in LSI, VLSI, and ULSI, the sequence of technologies leading to the enabling of SoC designs.

2.OVERVIEW OF THE PROJECT

Router is the building block of a Network-on-Chip (NoCs). 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. Considering the NoC components, as crossbars, arbiters, buffers, and links, in the experiments realized by 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. Regarding dynamic power, the buffers‘ consumption is also high, and it increases rapidly as the packet flow throughput increases. 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 neighboring 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 neighboring 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 favorably with other dynamic topologies like virtual channels.

3 RECONFIGURABLE ROUTER

3.1 INTRODUCTION

we will present the proposed reconfigurable router, where the NoC efficiency can be increased as a function of

 ISSN: 2321-8169
2963 – 2968

the possibility to reconfigure the buffer size according to the requirements of each channel of the router at run time, without the need to oversize buffers to guarantee performance.

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 neighboring routers (North, South, West, and East).

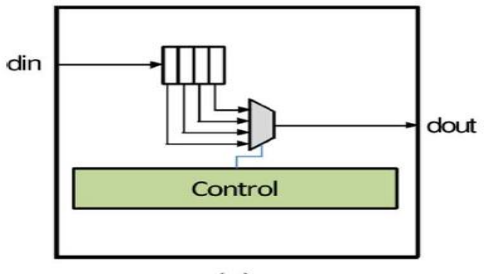


Fig 1 Input FIFO Original

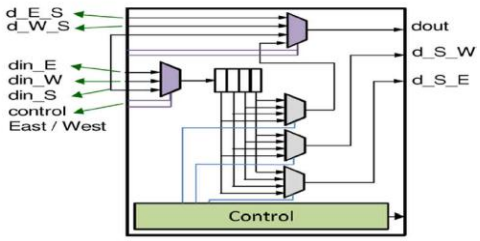


Fig 2 FIFO Proposed router.

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 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 flits is a header with destination address followed by a set of payload flits and a tail flits. To indicate this information (header, payload, and tail flits) two bits of each flits are used.

There is a round-robin arbiter at each output channel. The buffering is present only at the input channel. Each flits is stored in a FIFO buffer unit. The input channel is instantiated to all channels of the NoC, and thus all channels have the same buffer depth defined at design time.

3.2 ALGORITHM

Step1: start the program.
Step2: Read the input data packets from the system and read the channel width.
Step3: Initialize variables i=income flits, j=buffer Flits, k=Right buffer slots, l=left buffer slots.
Step4: Convert the incoming packets into flits.
Step5: Divide the flits according to the channel width.
Step6: If i=j, then no change in buffer size i.e. j=j
Step7: Route the flits.
Step8: Stop the program.
Step9: If i≠j, then checking
If i >j+k, then j=j+k+l and follows step7and
Else j=j+k and follows step7 and step8.
Step10: Stop the program

3.3 RECONFIGURABLE ROUTER ARCHITECTURE

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 neighboring buffers. The input and output of data in a channel is as defined in

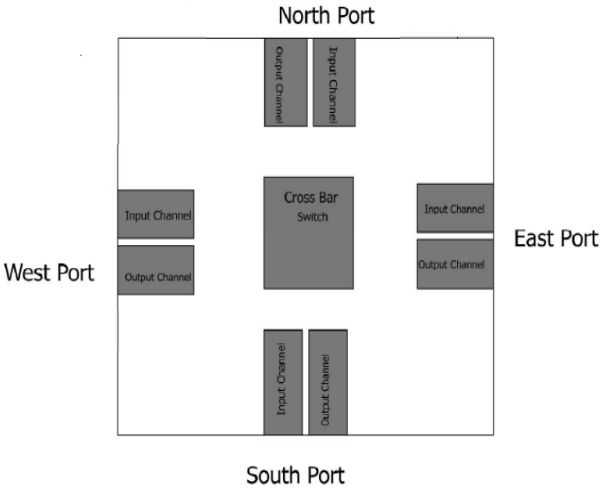


Fig 3: I/P & O/P view of the reconfigurable router

To reduce connection costs, each channel may only use the available buffer slots of its right and left neighbor 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. 1 & 2 shows the original and proposed input FIFO. Comparing the two architectures, the new proposal uses more multiplexers to allow the reconfiguration process. Fig. 3 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 neighbors. First the channel asks for buffer words to the right neighbor, and if it still needs more buffers, it tries to borrow from the left neighbor FIFO. In this manner, some signals of each channel must be sent for the neighboring 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 neighboring channels, and also how much the neighbor 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 slots or in the neighbor 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 neighbor buffer slots, and two more pointers to control the right neighbor 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 neighboring buffers, only the South, North, West, and East Channel of a router can make the use of their adjacent neighbors. As mentioned before, the loan granularity used in this proposal is a buffer slot. The results of the re-configurable 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 what- ever 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. 1 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 neighbor input (din E), and the left neighbor 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 neighbors. As the East Channel occupies two of its four slots, this channel can lend two slots to its neighbor, 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 flits stored in the East Channel, and this flits must be sent to the output, it is passed from the East Channel to the South Channel (d E S), and so the flits 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 out- puts (d_S_E and d_S_W) to send the flits stored in its channel but belonging to neighbor channels.

3.4 FUNCTIONING OF PROPOSED ROUTER:

The choice to resend the flits stored in a neighbor 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 neighbor 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. As each channel know, show many buffers lots it has allocated, when the

pointers present an address belonging to a neighbor buffer slot, the control of the first multiplexer of 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 neighbors are asked about lending/borrowing, no cycle can be made, and hence at the circuit level there is also no possibility of deadlock. Fig.18 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.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.

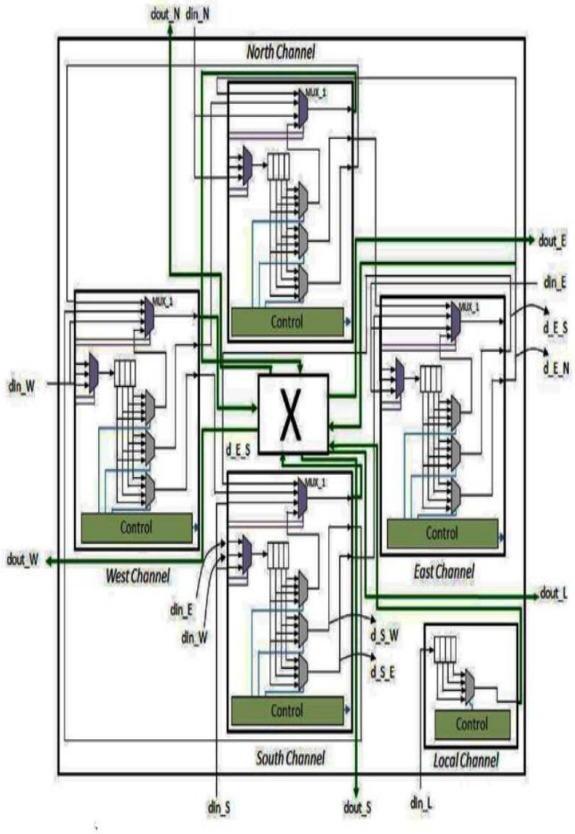


Fig 4 Proposed Router Architecture

The distribution of the buffer words among the neighbor 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).

As shown in Fig.6.Let we assume that the South channel divides part of its buffer with the neighboring channels (West and East channels).

In this case, South channel uses only two buffer slots, five buffer slots are used by West channel, and three buffer slots are used by East channel. The number of buffers of a channel is partitioned according to the need for loans among the channels. In this way, each buffer slot is allocated in a mutually independent way. Pointers to each buffer partition are used in order to control the flit storage process (read and write).

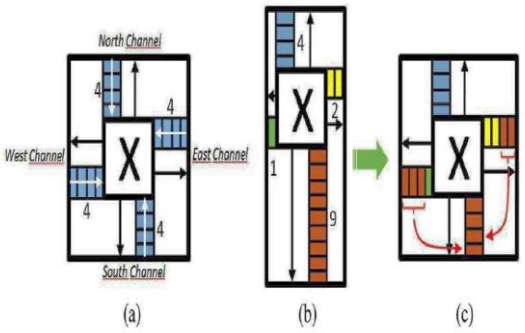


Fig 5 (a) Router designed with FIFO depth 4; (b) One example of need of configuration of the router; (c) Reconfiguration of the buffers to attend the need

Each slice of partitioned buffer in a channel has two pointers, one to control the read and another to control the write in the buffer (for example, addrrd E and addrwr E in Fig. 6). Besides the pointers, there are other control signals that are needed, as the signal that indicates when the partitioned buffers are empty and full. With these signals, each channel allows neighbor channels to allocate buffer slots of this port and to guarantee that the flits are not mixed among the channels.

The information about how many slots of buffers are used for each channel can been used to dynamically adjust their usage, consequently improving the efficiency. With this, one can monitor the NoC traffic flow and analyze how the resources are being used. This information can be used to increase the efficiency of the NoC design

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. 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

ISSN: 2321-8169
2963 – 2968

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. Let us consider an input buffer with ten slots, as shown in Fig.6

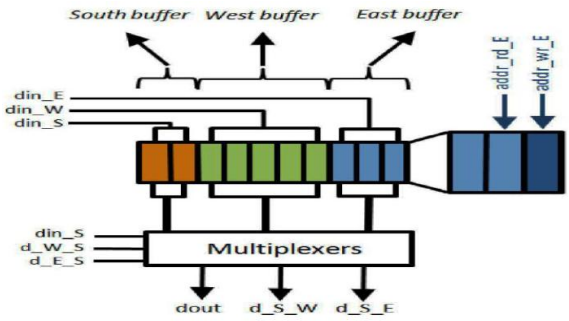


Fig 6: South input channel buffers partitioned for three channels: South, West, East

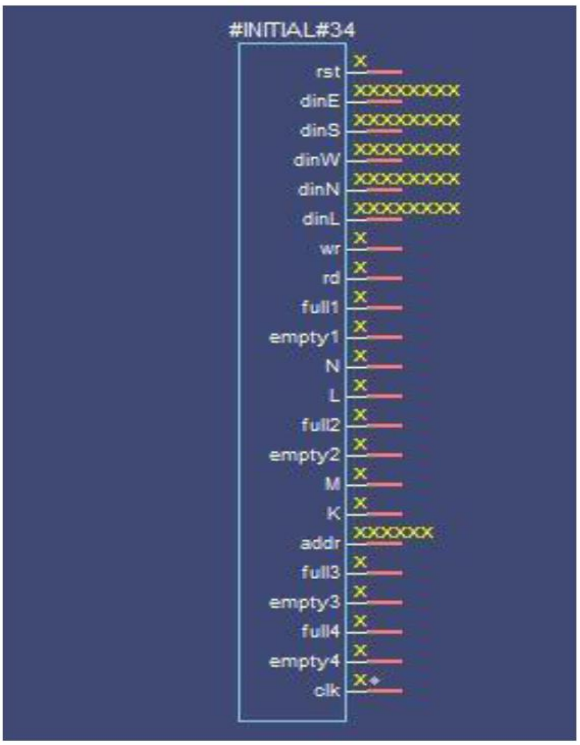
Each slice of partitioned buffer in a channel has two pointers, one to control the read and another to control the write in the buffer. Besides the pointers, there are other control signals that are needed, as the signal that indicates when the partitioned buffers are empty and full.

4 RESULT ANALYSIS

It provides information regarding schematic report and simulation result and how simulation and synthesis are done

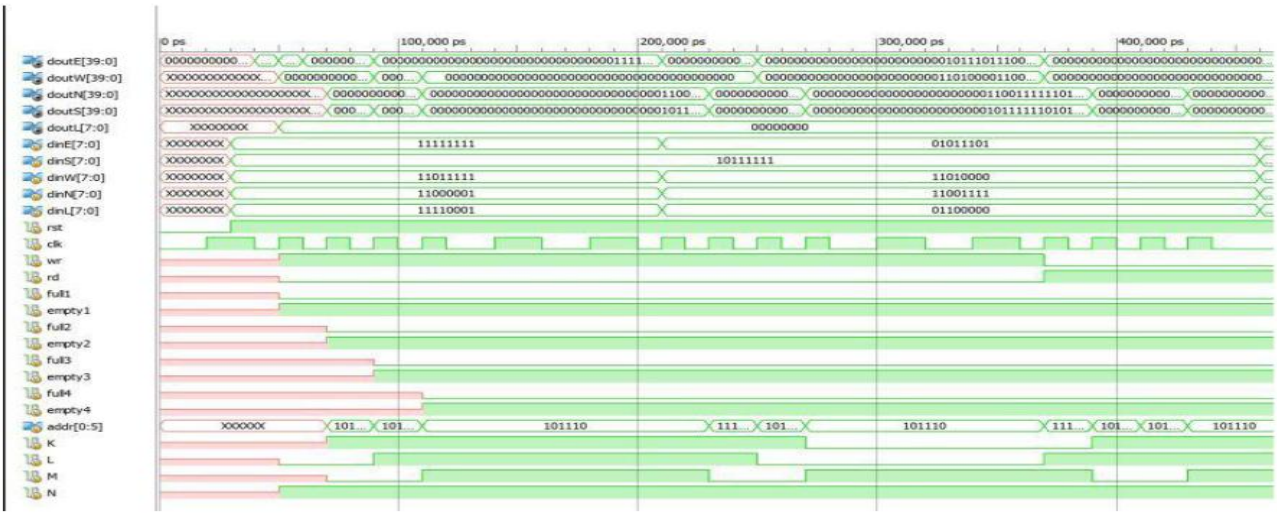
4.1 SCHEMATIC REPORT

The schematic representation of the reconfigurable structure is shown below. It is the representation of the reconfigurable router before giving any inputs



4.2 SIMULATION RESULT

This provides the simulated result for different inputs given to four channels and shows routing is performed



5 CONCLUSION

Here we conclude how the data is routed using our reconfigurable router and this is represented by using simulation waveform.the designed circuit is represented by using schematic report. 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 of the network bandwidth is required among the channels.

6. SCOPE FOR THE FUTURE WORK

As the proposed technique is for the reconfiguration of router in accordance with the application, obtaining similar performances even when the application changes. Now a days the work is more and so the fault tolerance can also be added to this system and also the fault routing avoidance system. One more scope is to implement the low power packet error correction detection along with re-configurability at low power and high performance.

7. REFERENCES:

[1] L. Benini and G. De Micheli, ―Network on chips: A new SoCpara- digm,‖ IEEE Computer, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[2] J. Andrews and N. Baker, ―Xbox 360 system architecture,‖ IEEE Micro, vol. 26, no. 2, pp. 25–37, Mar./Apr. 2006.

[3] 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.

[4] 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.

8. AUTHOR:

I worked four years as a Assistant Professor(ECE) in D.M.S S.V.H Engineering college ,Machilipatnam as a Radified Employee. Currently I am pursuing M.Tech in **GDMM College of Engineering** & **Technology** , Nandigama.