Design and Simulation of Router Using WWF Arbiter and Crossbar

M.Saravana Kumar, K.Rajasekar
Electronics and Communication Engineering
PSG College of Technology, Coimbatore, India

***Abstract* - Packet scheduling algorithms in switches and routers play a critical role in providing the Quality-of-Service (QoS). The combination of fairness, efficiency and low-latency makes Arbiter an attractive scheduling discipline for both best-effort and guaranteed-rate services. The interconnection network also supports random routine that allows the data connection to be routed automatically around the busy ports. In addition, the interconnection network provides broadcast facility which allows one node to send a message to all destination nodes it can used in general network interface applications. In this paper a Wrapped Wave Front Arbiter using Virtual Cut technique and a crossbar are designed. The result of the arbiter and the crossbar architecture is obtained using Xilinx 14.1 ISE design suite and its area, power and delay are calculated. Comparison between the existing routers has less area and delay with 7% of accuracy. Also Layout is generated using cadence SoC Encounter in 180nm technology.**

***Index Terms* - FIFO, Arbiter, Controller, Crossbar**

I. INTRODUCTION

Packet Switching networks have no dedicated connection between two Hosts. Information could take different routes to reach the destination Host. Each packet has a header that contains the source and destination addresses. The routers on a packet switching network regularly exchange information about network topology [2]. QoS is a set of technology for managing network traffic in a cost effective manner. Performance concerns are associated with latency and throughput.

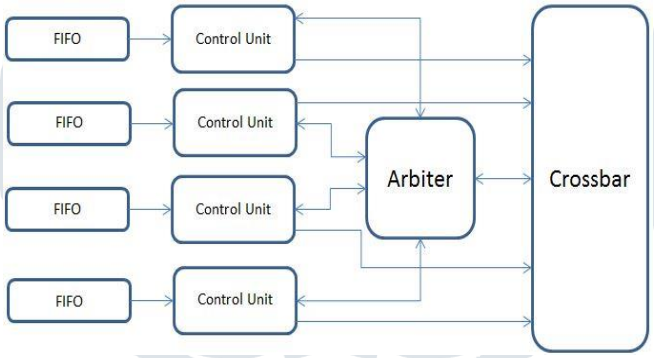

Fig 1 4x4 Crossbar Switch Architecture in NoC

A crossbar switch is an assembly of individual switches between multiple inputs and multiple outputs [2]. A major advantage of crossbar switching is that, as the traffic between any two devices increases, it does not affect traffic between other devices [3]. In addition to offering more flexibility, a cross-bar switch environment offers greater scalability than a bus environment [5].

The 4x4 crossbar switch router consists of FIFO with control unit, arbiter and crossbar is shown in the figure 1. FIFO is commonly used as buffering and its architecture is shown in Fig 2. Each FIFO is controlled by separate clocked read and write signals. Each FIFO is able to store more than two short messages. As soon as the first message is sent off the switch, the second message is ready to make a connection to a output port immediately.

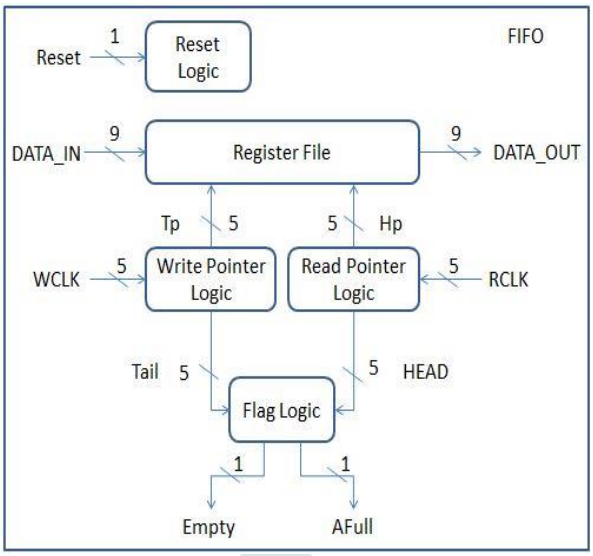

Fig 2 FIFO Logic Block Diagram

The Write operation is controlled by a write clock (WCLK) and a byte of data is written into the FIFO on the rising edge of the WCLK signal. As like the read operation is controlled by read clock (RCLK) and a byte of data is read from the FIFO on the rising edge of the RCLK signal. FIFO provides two status flags AFull and Empty. When the AFull flag is high, it indicates that the FIFO is full else FIFO is empty. Hence no additional data can be written into the FIFO otherwise the FIFO will overflow.

The main function of the control units is 1. To monitor the incoming data, to launch a request of connection if the head of a message is a routing command, to perform random routing if the routing command is random routing type and the selected output port is busy, to schedule data transfer after the request of connection is granted and also to close the connection after the tail of message is detected. The implementation of control unit is shown in the figure 3. On the rising edge of Reset signal, the control unit is reset to Idle State. Then if proceed to Read message State if and only if both of the input signals Empty and broadcast pending stay Low and pulse signal RCLK is generated.

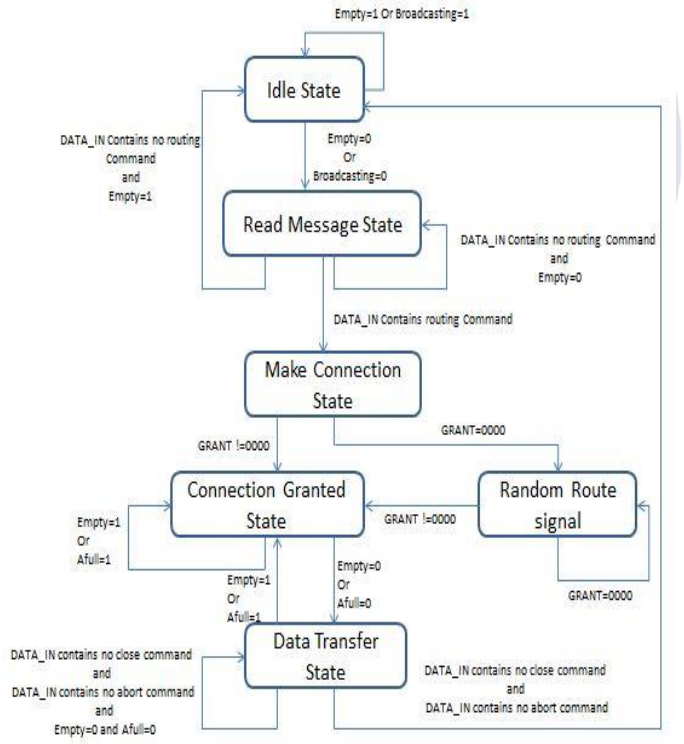

Fig 3 Control Unit

The control unit proceeds to Make Connection State if and only if the contents of Rdata-out is decoded to be a routing command. Depending on the type of routing command, the request may ask for connection to one output port, two output ports or four output ports. The control unit proceeds to Connection Granted State if and only if the request of connection is granted, otherwise the control unit proceeds to Random Route State from Make Connection State. In Random Route State, control unit performs random routing if the head of a message is a random routing type. If the new request of connection is granted, the control unit proceeds to Connection Granted State. In Connection Granted State, two input signal Empty and AFull are examined. If either Empty or AFull is high, the control unit remains in Connection Granted State. If neither Empty nor AFull is

high, the control unit proceeds to Data Transfer State. In Data Transfer State, the RCLK pulse is generated during the first half of clock cycle. The content of the register is represented by the signal DATA_OUT which goes to the crossbar. During the second half of the clock cycle, the WCLK pulse is generated. If the byte of data read from the FIFO is either a close or ah abort command, the control unit returns to Idle state or the Idle condition.

II. ARBITER

Crossbar is capable of supporting four logical connections simultaneously, a problem arises when more than one input port want to connect to a same output port. This conflicting demand for resources introduces delay to the interconnection network and reduces the network performance. Hence the 4x4 crossbar switch must have an arbiter to resolve the conflicts and provide efficient and fair scheduling of these resources.

The Arbiter consists of 16 REQUESTi.j input signals, one per crosspoint. A REQUESTi.j signal is asserted (High) when the use of the particular crosspoint is needed. The arbiter produces 16 GRANTi,j output signals, one per crosspoint. These output signals indicate which crosspoint requests have been granted.

In order for the arbiter to provide fair arbitration services, the Wrapped Wave Front Arbiter (WWFA) technique is adopted in out arbiter design. The WWFA use a wave front to select which of the four arbitration calls have top priority during the arbitration process. The wave front either moves diagonally from top left to the bottom right corner of the arbiter or moves vertically from the top row to the bottom row of the arbiter. The direction of movement depends on what mode the arbiter is operating in. if the arbiter is operation in non-broadcast mode, the wave front moves diagonally. If the arbiter is operating in broadcast mode, the wave front moves vertically. All the wave movements are controlled by a 4-bit circular shift register.

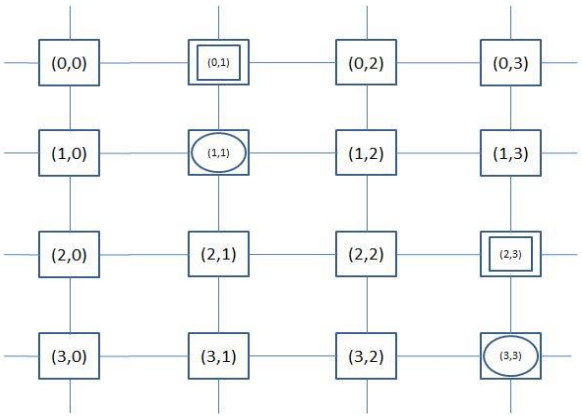

Fig 4 Diagonal Matrix of Wrapped Wave Front Arbiter

When the 4x4 crossbar switch is operating in the non-broadcast mode, the four top priority arbitration cell selected by the diagonal wave front are guaranteed to be located in different rows and columns. This maximizes the probability that multiple arbitration cells will win the arbitration. Diagonal matrix of wrapped wave front arbiter is shown in the figure 4. Arbitration process where the diagonal wave front consists of cells (0, 1), (1, 0), (3, 2), (2, 3).

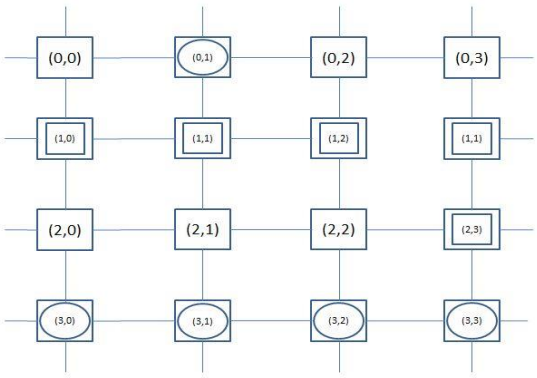

Fig 5 Horizontal Matrix of Wrapped Wave Front Arbiter

When the 4x4 crossbar switch is operating in the broadcast mode, the four top priority arbitration cell selected by the horizontal wave front are guaranteed to be located in the same row. This maximizes the probability that multiple arbitration cells in the same row will win the arbitration. Horizontal matrix of wrapped wave front arbiter is shown in the figure 5. Arbitration process where the horizontal wave front consists of cells (1, 0), (1, 1), (1, 2), and (1, 3).

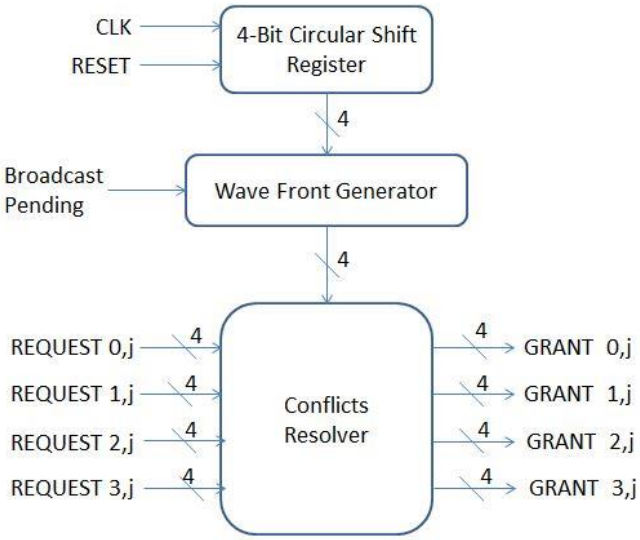

Fig 6 Arbiter Block Diagram

The arbiter consists of a 4-bit circular shift register, a wave front generator (WFG) and a conflict resolver (CR). The general block diagram of arbiter is shown in the figure 6.On the rising edge of the RESET signal, the shift register is initialized to have value 0001 in binary. The 4-bit output of the shift register is used by the WFG to generate the wave front. As mentioned above, two types of wave front can be generated depending on the value of input signal BroadcastPending. If BroadcastPending is high, it indicates that the 4x4 crossbar switch is operating in a broadcast mode. The horizontal wave front is generated. If the BroadcastPending is low, it indicates that the 4x4 crossbar switch is operating in a non-broadcast mode. The diagonal wave front is generated.

CR is used to resolve the conflicts when more than one input port makes the requests for the same output port. The CR performs the arbitration process and those arbitration cells with top priority according to the wave front generated by the WFG will the arbitration. In case there is only one request signal in column, the request is granted immediately regardless of the wave front.

III. CROSSBAR

The crossbar provides a logical connection between an input port and an output port. It consists of four horizontal buses (rows) and four vertical buses (columns). A horizontal bus intersects a vertical bus at a crosspoint. Each of the horizontal bus and vertical bus is 9-bits wide. The four pairs of input data bus and output data bus are interconnected together. Each port has two data flow signals. For an input port, the data flow signals are (WCLK_INi and STOP_OUTi). For an output port, the data flow signals are (WCLK_INj and STOP_OUTj). The values of WCLK_OUTj and STOP_OUTi are determined by the following Boolean equation.

$$WCLK\_OUT_j = (WCLK\_IN0 \times GRANT_{0,j}) + (WCLK\_IN1 \times GRANT_{1,j}) + (WCLK\_IN2 \times GRANT_{2,j}) + (WCLK\_IN4 \times GRANT_{4,j})$$

$$STOP\_OUT_i = ((STOP\_IN0 \times GRANT_{i,0}) + (STOP\_IN1 \times GRANT_{i,1}) + (STOP\_IN2 \times GRANT_{i,2}) + (STOP\_IN3 \times GRANT_{i,3})) \times (GRANT_{i,0} + GRANT_{i,1} + GRANT_{i,2} + GRANT_{i,4})$$

A logical connection is made by a routing command by a control signal GRANTi,j at each crosspoint, where 'i' is the input port number and j is the output port number range between ($0 \leq i \leq 3$, $0 \leq j \leq 3$). Upon receipt of the close or abort command, the logical connection will be closed. When GRANTi,j is high, the connection between input port i and output port j is established. Whatever the value appears at DATA_INi will also appears at DATA_OUTj.

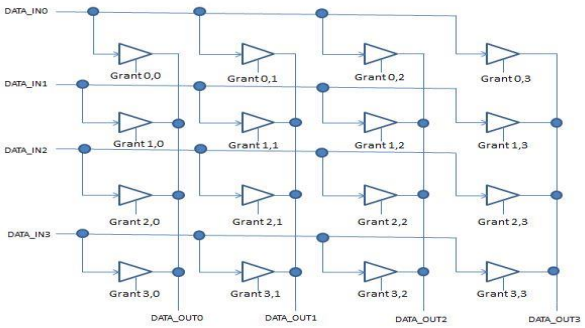

Fig 7 Functionality of Cross Bar

IV. SIMULATION AND RESULT

A. Output Waveform For Router

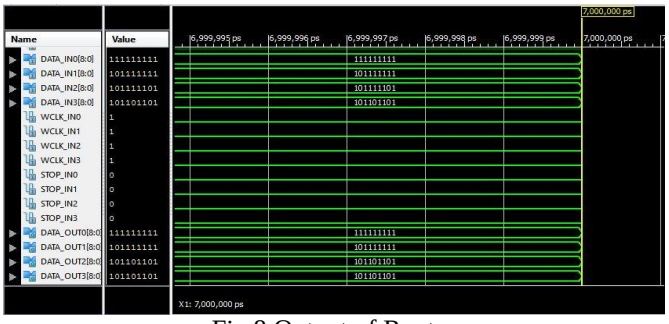

Fig 8 Output of Router

B. Layout Of Router

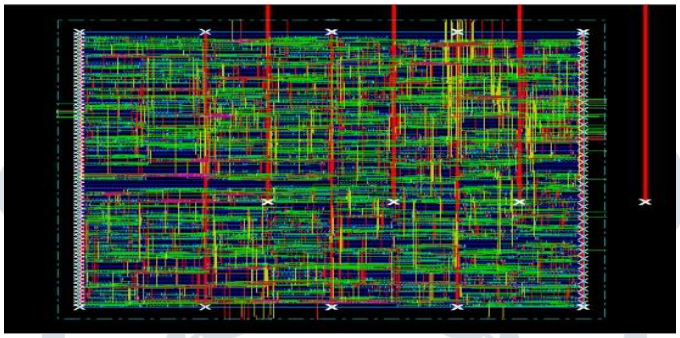

Fig 9 Layout of Router in Cadence SoC Encounter

Table 1 Performance of Various Designs in Router

S.No	Type	Area of Cell (nm)	Delay (ns)
1	Existing Router	2450.17	9.342
2	Router using Wrapped Wave Front Arbiter	2299.45	7.193

Table 2 Performances of Routers in Cadence Soc Encounter

S.NO	ROUTER SWITCH	LEAKAGE (NW)	SWITCHING (NW)	INTERNALS (NW)
1	FIFO BUFFER	2555.04	134058.60	10759.38
2	CONTROL UNIT	319.98	23719.74	11897.35
3	ARBITER	242.56	3712.98	1941.85
4	CROSSBAR	669.64	6768.59	2038.30
5	ROUTER	11962.48	822180.39	487187.13

V. CONCLUSION

The implementation and performance of router were analyzed with individual modules using Xilinx 14.1 ISE design suite and cadence SoC encounter for calculating area and delay. When compared with other existing switch design, our designed crossbar switch provides less area and delay with 7% of accuracy.

REFERENCES

[1] Deng Pan and Yuanyuan Yang, "FIFO-Based Multicast Scheduling Algorithm for Virtual Output Queued Packet Switches", IEEE Transactions on computers, vol. 54, no. 10, October 2005.
[2] Hsin-Chou Chi and Yuval Tamir "Decomposed Arbiters for Large Crossbars with Multi-Queue Input Buffers"
[3] Gene Eu Jan , Shao-Wei Leu, W. R. Shu I Chen "FPGA Implementation of a Multicasting Crossbar Switch", International Conference on Microelectronics 2009
[4] Mr. Suyog K.Dahule,Dr. M.A.Gaikwad ,"Design & Analysis of Matrix Arbiter for NoC Architecture", International Journal of Advanced Research in Computer Science and Electronics Engineering Volume 1, Issue 5, July 2012 .
[5] P. B. Domkondwar, Dr. D.S.Chaudhari, "Implementation of Five Port Router Architecture Using VHDL", International Journal of Advanced Research in Computer Science and Electronics Engineering Volume 1, Issue 3, May 2012.
[6] David Bafumba-Lokilo, Yvon Savaria," Generic Crossbar Network on Chip for FPGA MPSoCs".

[7] Tarun Kumar Gauttam, Rekha Agrawal, Sandhya Sharma, "Arbiter Design Using Verilog for Switching to Communicate in Between Multiple Resources", International Journal of Innovative Technology and Exploring Engineering (IJITEE) Volume-3, Issue-3, August 2013.
[8] Suyog K.Dahule, M.A.Gaikwad "Design & Simulation of Round Robin Arbiter for NoC Architecture", International Journal of Engineering and Advanced Technology (IJEAT) Volume-1, Issue-6, August 2012.