Design and Analysis of On-Chip Router for Network On Chip

Ms. A.S. Kale [#1]
M.Tech IInd yr, *Electronics Department,*
Bapurao Deshmukh college of engineering, Wardha M. S.India
Prof. M.A.Gaikwad [#2]
Professor, Electronics Department,
Bapurao Deshmukh college of engineering, Wardha M. S.India

Abstract- **Continuous scaling of CMOS technology makes it possible to integrate a large number of heterogeneous devices that need to communicate efficiently on a single chip. For this efficient routers are needed to takes place communication between these devices. This paper gives the design of on-chip routers based on optimizing power consumption and chip area. Proposed architecture of on-chip router in this paper give the results in which power consumption is reduced and silicon area is also minimize.**

Keywords- Arbiter, Network on chip (NOC), Router

I. INTRODUCTION

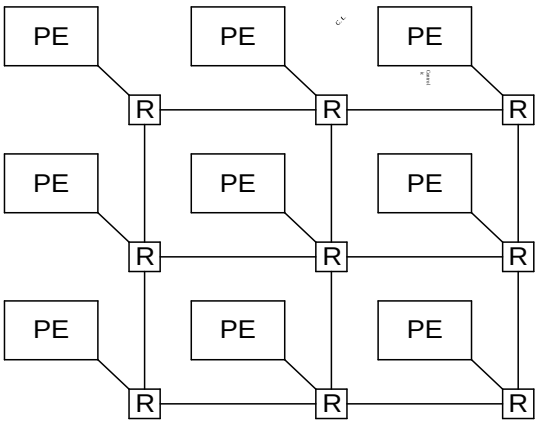
Fig.2 - 5 Input 5 Output port Unidirectional On-Chip Router

1. FIFO

FIFO (First In First Out) is used as input buffer to store the data temporarily. The status of FIFO decides the communication can start or not. If the FIFO is empty the data can be written in it and communication can start. If FIFO is full, data can be forwarded to its destination router. The read and write operation of FIFO is controlled by control logic. If the input is present on input port and ready signal 1.1 Network on chip

A variety of interconnection schemes are currently in use, including crossbar, buses and NOCs. Of these, later two are dominant in research community. However buses suffers from poor scalability because as the number of processing elements increases, performance degrades dramatically. Hence they are not considered where processing elements are more. To overcome this limitation attention has shifted to packet-based on-chip communication networks, known as Network-On-Chip (NOC).

A typical NoC consists of computational processing elements (PEs), network interfaces (NIs), and routers. The latter two comprise the communication architecture. The NI is used to packetize data before using the router backbone to traverse the NoC. Each PE is attached to an NI which connects the PE to a local router. When a packet was sent from a source PE to a destination PE, the packet is forwarded hop by hop on the network via the decision made by each router. For each router, the packet is first received and stored at an input buffer. Then the control logics in the router are responsible to make routing decision and channel arbitration. Finally, the granted packet will traverse through a crossbar to the next router, and the process repeats until the packet arrives at its destination.



PE – Processing Element

R – Router

Fig:1 Typical NOC Architecture

1.2 On-Chip Router

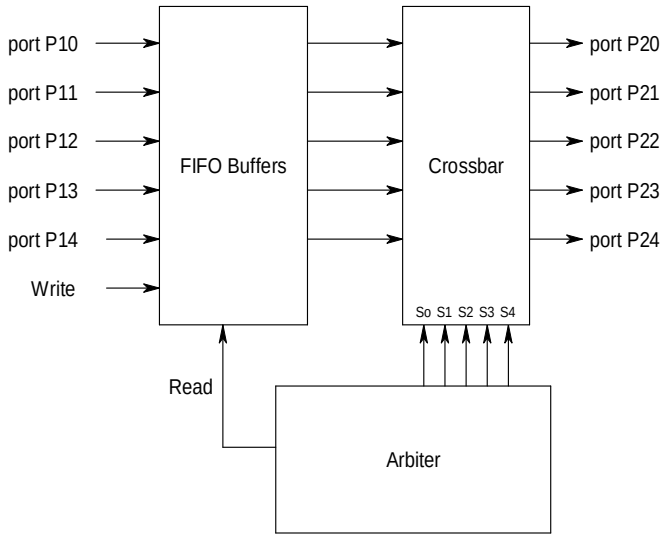
The heart of an on-chip network is the router, which undertakes crucial task of co-ordinating the data flow. The router operation revolves around two fundamental regimes: (a) the datapath and (b) the associated control logic. The datapath consist of number of input and output channels to facilitated packet switching and traversal. Generally 5 input X 5 output router is used. Out of five ports four ports are in cardinal direction (North, South, East, Waste) and one port is attached to it's local processing element

Like in any other network, router is the most important component for the design of communication back-bone of a NoC system. In a packet switched network, the functionality of the router is to forward an incoming packet to the destination resource if it is directly connected to it, or to forward the packet to another router connected to it. It is very important that design of a NoC router should be as simple as possible because implementation cost increases with an increase in the design complexity of a router.

II. PROPOSED ROUTER ARCHITECTURE

The proposed architecture consist of mainly three parts:

1. FIFO
2. Crossbar Switch
3. Arbiter



corresponding to that port is high then read or write operation can be performed.

Read counter (cr) and write counter (cw) are the variables which stores number of read and write operation on the FIFO buffers. These variables are used to know whether the FIFO is empty or full.

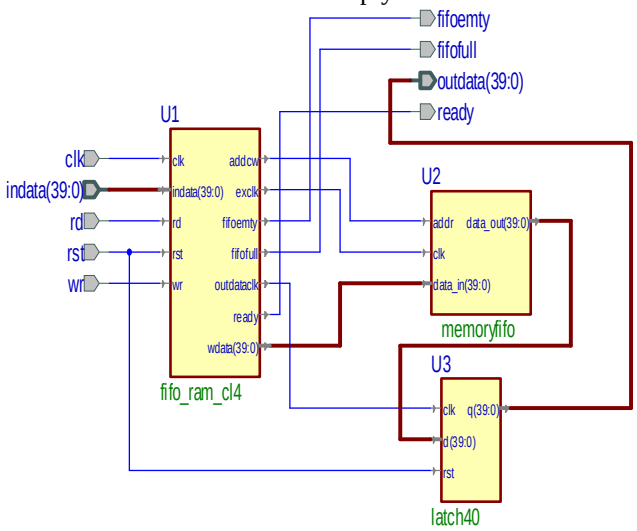


Fig.3 FIFO buffer

When read signal (rd) is high, control logic first check fifoempty signal. If it is high operation is terminated and if it is low packet is read from the memory and cr is incremented by one. When write signal (wr) is high, control logic first check fifofull signal. If fifofull is high, it means memory is full and no more packets are added in it and operation is terminated. But if it is low, packet is write into the memory. The address of memory where the packet is to

store is also generated by control logic and cr is incremented by one.

After all these read counter (cr) and write counter (cw) are compared. If cr is equal to cw that means read operation is equal to write operation therefore fifofull=0 and fifoempty=1. If cw=4 then fifofull=1 and fifoempty=0. If cw>cr and cw<4 then fifofull=0 and fifoempty=0.

Fig.3 shows FIFO buffer.

2. Arbiter

Arbiter controls the arbitration of the ports and resolves contention problem. It keeps the updated status of all the ports and knows which ports are free and which ports are communicating with each other.

Packets with the same priority and destined for the same output port are scheduled with a round-robin arbiter. Supposing in a given period of time, there was many input ports request the same output or resource, the arbiter is in charge of processing the priorities among many different request inputs. The arbiter will release the output port which is connected to the crossbar once the last packet has finished transmission. So that other waiting packets could use the output by the arbitration of arbiter.

In proposed work, we are using ROUND ROBIN ARBITRATION ALGORITHM. A round-robin arbiter operates on the principle that a request which was just served should have the lowest priority on the next round of arbitration.

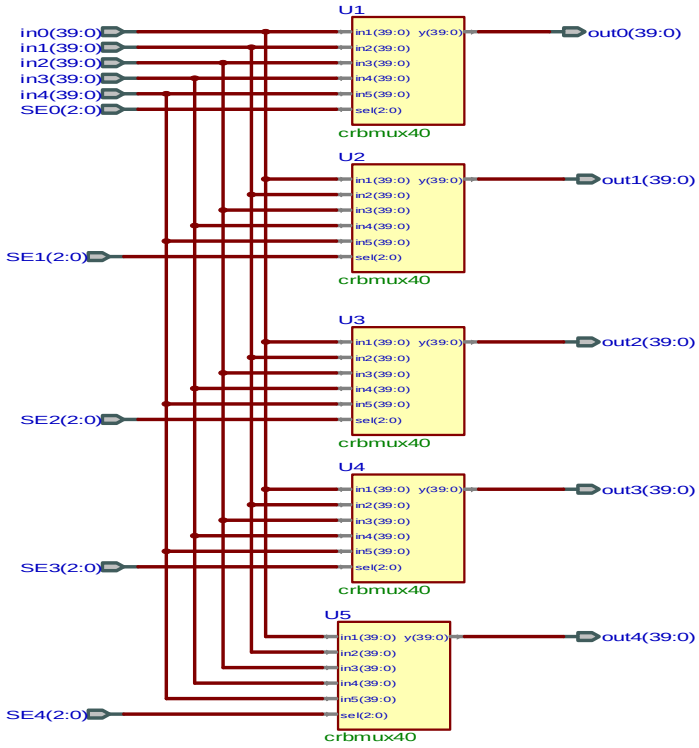


Fig.4 Crossbar

3. Crossbar

A crossbar switch (also known as cross-point switch, crosspoint switch, or matrix switch) is a switch connecting multiple inputs to multiple outputs in a matrix manner. The design of crossbar switch has 5 inputs and 5 outputs. Fig.4 shows Multiplexer based crossbar switch. As we are getting five input packets of 40 bits each from five ports of router, number of 5:1 multiplexers used inside the crossbar are five. All five inputs are given to all the multiplexers. Select line is of three bit. Out of five select lines which one is selected is depend on the logic of arbiter. Outputs of multiplexers are the output ports of the 5X5 router.

III SIMULATION

Simulation refers to the verification of a design, its function and performance. It is process of applying stimuli to a model over time and producing corresponding responses from a model. Fig.6 shows the simulation result of unidirectional router. This simulation is performed on Active-HDL software.

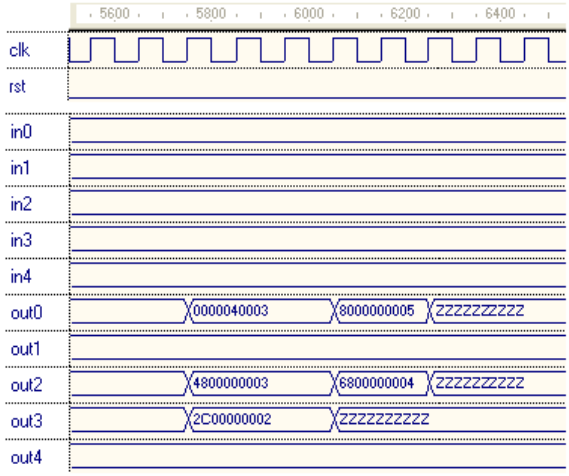


Fig. 6 Simulation result of unidirectional router

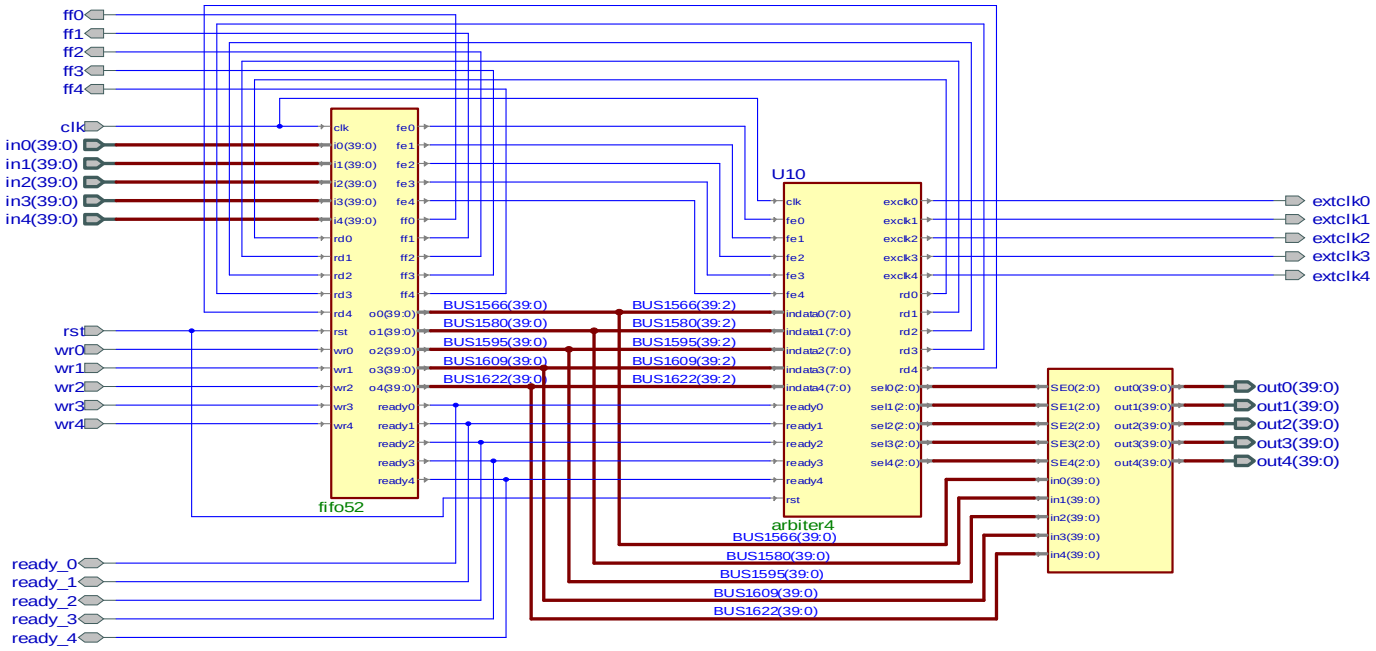


Fig. 5 complete Unidirectional Router

IV SYNTHESIS

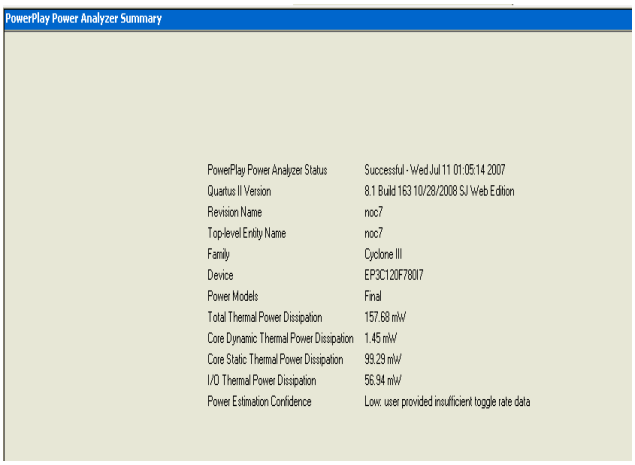


Fig. 7 power report

This is the power analysis report of unidirectional router calculated in Quartus II software

IV CONCLUSION

Finally after simulating above unidirectional router we get a synthesis and simulation repot of five port router with the help of Active-HDL and Quartus II web edition, that help in understanding proper functioning of Five port router for network on chip.

V REFERENCES

[1] W. 1. Dally and B. Towles, Route Packets, Not Wires: On-Chip Interconnection Networks, In Proceedings of
the 38th Design Automation Conference, p.684 (2001)

[2] L. Benini and D. Micheli, Networks on Chips: A New SoC Paradigm, IEEE Computer, 35, p.70 (2002)

[3]Khalid Latif, Tiberiu Seceleanu, Hannu Tenhunen, Power and Area Efficient Design of Network-on-Chip Router Through Utilization of Idle Buffers

[4]Cheng Liu·, Liyi Xiao, Fangfa Fu, Design and Analysis of On-Chip Router

[5]Eung S. Shin, Vincent J. Mooney III and George F. Riley, Round-robin Arbiter Design and Generation