P. Gowri Swetha* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.4, Issue No.5, August – September 2016, 3497 – 3501.

Efficient Router for MPSoC Communication

P. GOWRI SWETHA
Assistant professor, Department of ECE
Indo American Institution Technical Campus
Sankaram, Anakkapalle, Visakhapatnam

Abstract: **Moore's law fashioned a major revolution in semiconductor industry which is the System-on-chip (SoC). Multiple independent circuit implementations are integrated into a single chip in SoC. The applications which require more number of processors leads to the development of Multiprocessor System on Chip (MPSoC). The MPSoC design is having their own share of issues and challenges. The ever increasing complexity in modern MPSoC designs makes challenges to the design engineers. The communication between different IP cores is one of the major problems faced by MPSoC. This works presents an efficient FSM and FIFO base router design for MPSoC communication. The design functional verification and synthesis is done by using Xilinx-ISE. The obtained results matching with expected ones show that this will be an efficient solution for the communication field.**

Keywords: **Multiprocessor System-On-Chip; Network On Chip; Open Core Protocol;**

I. INTRODUCTION

The dynamic changes in the semiconductor processing technologies led to the rapid growth of Very Large Scale Integration (VLSI). As VLSI technology scales down the number of modules on a chip multiplies, hundreds or even thousands of Processing Elements (PEs) can be integrated on the same chip. The integration of all electronic devices in a system into a single integrated circuit can be called as "System-on-Chip". System-on-chip (SoCs) applications are having higher demand because of data-processing capability as well as performance of parallel and multi-threading tasks. To meet the performance goals of some applications single processor may be insufficient and we required multiple processors on a single System-On-Chip (SoC). Multiprocessor Systems-On-Chips (MPSoCs) are custom architecture that balances the constraints of VLSI technology with an application need [14]. MPSoC consists of large number of components as well as the interconnection between all these components. Large number of computational units may include in a single chip. In order to achieve a proper inter core communication as well as to achieve high scalability and performance the traditional communication architectures become insufficient. The ongoing effort from all over the world to develop an efficient system to provide communication in system on chip leads to the emergence of "Network on Chip" (NoC) [2] [8] [13] [15]. Designing a Network on chip which fulfils all the application needs involves complex processes. To achieve this we have to use architecture which places all the resources in an efficient manner so that the entire system will be more optimized.

The router and the routing algorithms implemented using switches can give better performance. Switches are very small to implement as well as it can achieve data flow control. The thought of designers to develop a reconfigurable crossbar switch for NoCs helps to gain a high data throughput and is capable of adapting various topologies. These switches help to achieve better output latency, resource usage, and power consumption than a traditional crossbar switch. In different types of topology the implementation of switch architecture also plays an important role in communication.

This works presence the use of finite state machine and FIFO for switch architecture implementation. In this flow control can be achieved using request and acknowledgement. FIFO buffers are used at the output port in order to reduce the congestion and to improve performance. The design is coded in Verilog language and the simulation and functional confirmation is done using Xilinx ISE simulator.

II. SWITCH ARCHITECTURE DESIGN USING FSM AND FIFO

Since the Multiprocessor system on chip uses Network on chip approach for communication and the switch based network on chip is giving better performance, design of switch architecture is the most important task in the design of communication system for MPSoC. The switch architecture should be in such a way that it can easily perform the routing function as well as components of the architecture should be placed in such a manner to improve the system performance in terms of power, delay and resource utilization. The initial step is the selection of switch based topology.

The Clos network topology, a family of multistage network and can be used to build scalable multiprocessors. A 3 stage clos network (shown in figure 1) can be represented as C(n,m,p) where 'n' represents the number of input in the first stage switches and 'm' is the no of second stage switches and 'p' is the number of first stage switches [1] [3] [9] [12].. Most practical MPSoCs make use of C

(4,4,4) as network topology in order to get a parallelism degree of 16 for the designed network.

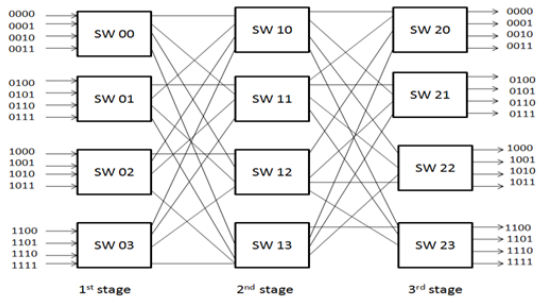

Figure .1: Clos Network Topology

The each stage switches uses a path diversity scheme in order to support circuit switching scheme (shown in figure 2). The network uses a pipelined circuit switching scheme which is having three phases the setup, the transfer and the release [7]. The setup phase is used for the dynamic path setup scheme which can support the run time path arrangement. The bit format used by the handshake signal include 1 bit request (Req) and 2 bit answer(Ans).When the switch request an idle link the corresponding downstream switch in the setup phase will set Req=1.Whenever the switch releases the output link Req=0 will set. The answer having two bit which is having three modes [1][5][10]. The Ans=01(Ack) means that destination ready to receive data from the source. When Ans=01 propagate back to source it indicate that the path is set up and the data transfer can be started immediately. The Ans =11(nAck) is reserved for end to end flow control which will set when the receiving circuit is not ready to receive data due to being busy with other tasks or overflow at the receiving buffer etc..If the link is blocked the Ans=10(back).

When permutation is changed, the run time path arrangement can be done with the help of dynamic path setup scheme .Dynamic probing mechanism is used for the path setup, which starts from an input to find a path leading to its corresponding output. The concept of probing is that a probe is dynamically sent under a routing algorithm in order to establish a path towards the destination. Exhaustible profitable backtracking (EPB) is use to route the probe in the network. Always an available path will exist from an idle input leading to an idle output. With the help of exhaustive property of EPB based approach, the path set up completely searches all the possible paths within the setup of path diversity between an idle input and idle output.

In the C (4, 4, 4) topology the EPB based path set up scheme can always find an always find an available path within the set of 4 possible paths between the input and idle output. The proposed C (4, 4, 4) topology can realize the path arrangement for full as well as partial permutation. In this network each input sends a probe which contains a 4-bit output address to find an available path to the corresponding output. If a free link will be available the probe will move forward else it will move forward. With this non-repetitive movement, the probe finds an available path between the input and its corresponding output. The EPB based path setup scheme is designed with a set of probe routing algorithm. Algorithm is depending upon the probe movement [1] [4] [11].

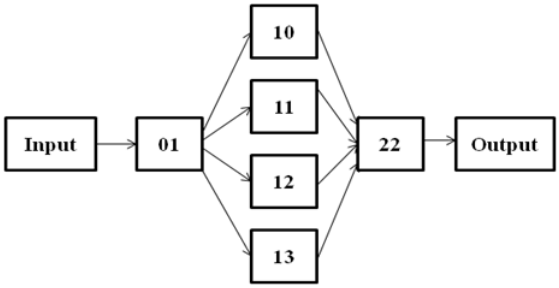

Figure 2: Path Diversity Scheme

1.1. Dynamic path set up scheme

If an input from switch "01" wants to send data to the destination switch or the third stage switch "22". First the probe will check all the four possibility in second stage switches 10, 11, 12 and 13.If the switch "10" is available then the "01" switch places a request to "10" and arrives at the switch "10". Again the switch checks "22" is available or not. If the destination is available the switch reaches the destination and acknowledgement signal propagates to the source. If the destination is not available then the probe will move backwards and answer will be back signal and the link will be disabled by releasing the link keeping request=0.Thus if the free path is available probe will move forward and if a blocked link is there then it will move backward and will try another link. By continuously following this manner probe can dynamically set the path [1] [6].

1.2. Switch Architecture

The on-chip network is having three kinds of switches. All these three are based on a common switch architecture shown in the figure 3. The switch architecture having four basic components Input Control's (ICs), Output Control's (OCs), an arbiter and a crossbar. The wiring cost can be reduced by the passage of incoming probes in the setup phase through the data path. The arbiter is the main functional block in the design and it is having mainly two functions. One is to cross connect Ans_out and input controls. The second function is to act as a referee for all the requests coming from the input controls. At the same time, the corresponding input control will send a request to the arbiter to grant the access. When the arbiter will receive the request it will cross connect the

Ans_out with the input controls [1] [7]. When several input controls will request the same free output then we can make use of arbiter based on predefined priority rule.

For the implementation of ICs we can make use of Finite state machine (FSM). The complete operation and the routing algorithm are completely implemented using the FSM. The OCs working is based upon the arbiter commands [1]. The crossbar is designed using multiplexers. The function of the control part is to perform dynamic path set up where us data path simply provides the circuit switched data. For the synchronous data transfer between the input and output end FIFO wrappers are used.

III. DESIGN OF SWITCH ARCHITECTURE

The FSM (Finite State Machine) is the major block which performs all the operations. The routing algorithm is implemented using this block. The path for the data flow is based on routing algorithm thus based on FSM. Based on the control signal provided by FSM router the input signal will be stored in one of the four FIFO (First In First Out) buffer. FIFO buffer can decouple the adjacent channel thus it can improve the efficiency. The synchronization between these two modules can achieve using FIFO synchronization module.

The FSM router will be the controlling entity of all the routing architecture. When a new packet is sent to the router this block will generate the entire control signal. FSM controls the read and write operations of FIFO. So FSM should make a connection to FIFO for the coding. Thus the combined operation of FSM and FIFO can perform read and write operation or the data transfer between two nodes in a system on chip.

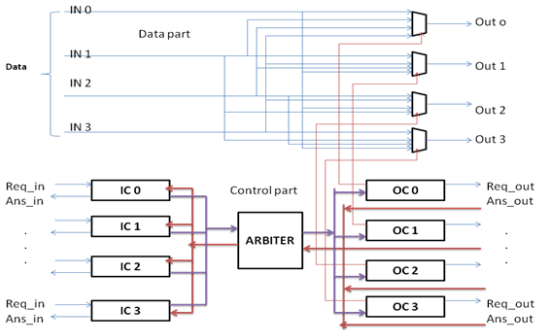

Figure 3: Switch Architecture

The pin diagram for the FSM router is shown in figure 4. The FSM changes its state depending on the conditions fed to its input side. Depending on condition it will be in one of the eight states.

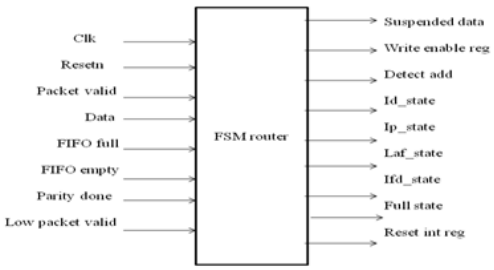

Figure 4: FSM Router

The FIFO (First in First out) buffer can be used to store the data temporarily. The status of the FIFO buffer decides whether the data can transfer or not. The writing as well as reading operation of the buffer is controlled by the FSM. The block diagram for FIFO is shown in the figure 5.

The operation of FIFO is based upon the clock as well as it will be depends on the resetn which is the negation of reset. If this resetn is a low value the output full will be equal to a zero value and the empty will be having a value of one. At the same time output data out will be 0. FIFO full implies that all the location in the FIFO is full with data. FIFO empty implies that all locations in the FIFO will be empty. The condition for the write operations are FIFO should not be full and write enable should be activate. At this stage data at the input stage will be passing at rising edge of the clock. When the read enable is activated and FIFO is not empty data can be read from the data_out.

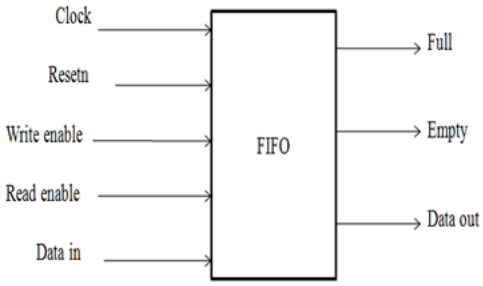

Figure 5: FIFO block

The synchronization between FSM and FIFO modules can be achieved using FIFO synchronization module and is shown in fig.6.. It provides a reliable communication between the input and output module which are the FSM and FIFO. The FSM will give data as well particular address of the channel. The data can be passing to the output depending on the specified channel address. This will latch the packet till the packet_valid is asserted and when it will be active it will pass to the output FIFO.

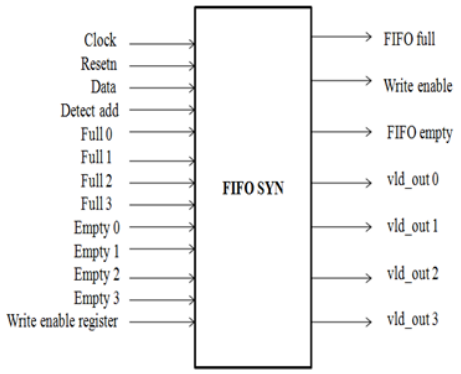

Figure 6: FIFO synchronization module

IV. RESULTS AND DISCUSSIONS

The design is implemented using widely used Hardware Description Language (HDL) Verilog. The simulation and verification is done using Xilinx ISE 8.2i and the simulated waveforms are shown in figure 7,8,9,10, and 11.

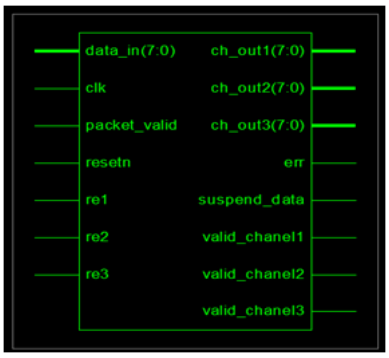

Figure 7: Pin Diagram of Router

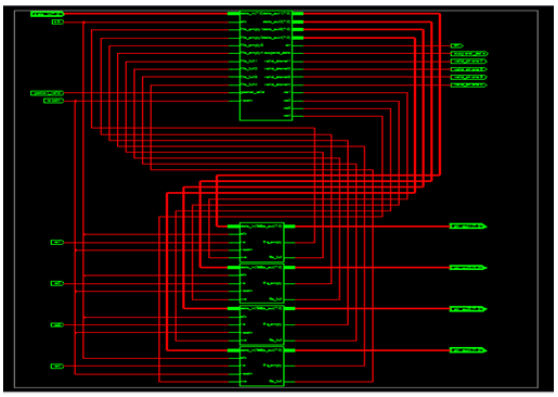

Figure 8: RTL schematic of Router

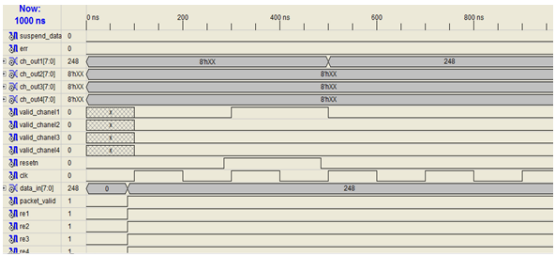

Figure 9: Simulated waveform for Router

Device Utilization Summary (estimated values)		
Logic Utilization	**Used**	**Available**
Number of Slices	194	6144
Number of Slice Flip Flops	148	12288
Number of 4 input LUTs	310	12288
Number of bonded IOBs	53	240
Number of GCLKs	2	32

Figure 10: Device Utilization Summary for Router

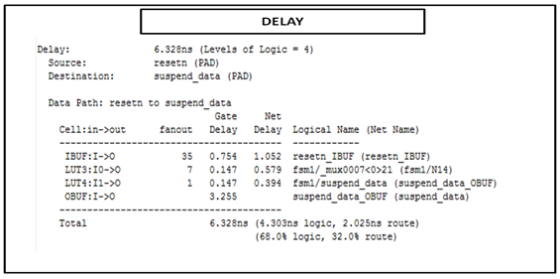

Figure 11: Delay for Router

V. CONCLUSION

An efficient router for MPSoC is developed using finite state machine and FIFO buffer. The specification is developed using verilog code and is simulated using Xilinx ISE. The simulation waveforms indicated that the communications between two IP cores are carried out properly. The results also show that its performance is excellent for FSM and FIFO based design in terms of parameters such as delay and resource utilization.

VI. REFERENCES

[1] Phi-Hung Pham, Junyoung Song, Jongsun Park, Chulwoo Kim, "Design And Implementation Of An On-Chip Permutation Network For Multiprocessor System-On-Chip". IEEE Transaction On Very Large Scale Integration Systems, Vol. 21 No.1, January 2013.

[2] Naveen Choudhary, "Bursty Communication Performance Analysis Of Network-On-Chip With Diverse Traffic Permutations". International Journal Of Soft Computing And Engineering Issn: 2231-2307, Volume-1, Issue-6, January 2012.

[3] B.Jagadeeswara Reddy, L.S.Devaraj "Low Latency On Chip Permutation Network For Multiprocessor System On Chip". International Conference On Electrical, Electronics And Computer Science (Eecs-2014)-28th September 2014.

[4] Zhonghai Lu, "Design And Analysis Of On-Chip CommunicationFor Network-On-Chip Platforms",.Stockholm 2007.

[5] P Sebastin Ashok, Ayogananth, P Rajasekar1 And S Shyamaladevi, "Design Of An On-Chip Permutation Network For

Multiprocessor Soc". Issn 2319-5991 Ijerst Vol. 3, No. 2, May, 2014.

[6] J.Janaki, S.Jyothirmaye, "On-Chip Permutation Network For Multiprocessor System-On-Chip". International Journal Of Research In Electronics And Communication Technology (Ijrect 2014).

[7] B.Sree Lekha, G.Dilli Rani, V Thrimurthulu, "Pipelined Routing Network For Multiprocessor Network On Chip" Proceedings Of Eleventh Irf International Conference, 17th August 2014.

[8] Luca Benini, Giovanni De Micheli, "Networks On Chips: A New Soc Paradigm" 0018-9162/02/$17.00 © 2002 IEEE.

[9] M.V.V.Prasanthi, V.Sreevani, "Multiprocessor System-On- Chip Implementation For Network-On-Chip Communication With 5-Bit Data Line". Internationa Journal Of Vlsi And Embedded Systems-Ijves, Issn: 2249 – 6556.

[10] Manjunath E1, Dhana Selvi D, "Design And Implementation Of An On-Chip Permutation Network For Multiprocessor System-On-Chip", International Journal Of Advanced Research In Electrical, Electronics And Instrumentation Engineering Vol. 3, Issue 6, June 2014.

[11] Ravindra Reddy Vinta, V.T.Venkateswarlu, "Configuration And Programmable Arbiter For Runtime Traffic Permutation Using Multi Stage Switching". International Journal Of Science, Engineering And Technology Research (Ijsetr), Volume 3, Issue 12, December 2014.

[12] Arun Arjunan., Karthika Manilal. "Noise Tolerant And Faster On Chip Communication Using Binoc Model" International Journal Of Modern Engineering Research (Ijmer) Vol. 3, Issue. 5, Sep - Oct. 2013 Pp-3188-3195 Issn: 2249-6645.

[13] Lionel Torres, Pascal Benoit, Gilles Sassatelli, Michel Robert, "An Introduction To Multi-Core System On Chip –Trends And Challenges". University Of Montpellier 2, France

[14] "A Comparison Of Network-On-Chip And Busses" Arteris

[15] Ravali K, Kumar Chowhan, "A New Approach To Design Fault Coverage Circuit With Efficient Hardware Utilization For Testing Applications". International Journal Of Advanced Research In Computer Science And Software Engineering, Volume 2, Issue 6, May 2012.

