

Implementation of Scheduling Tasks between different processing elements in a Network-On-chip

J.Senthil Kumar^{#1}, P.K.SaranyaDevi^{#2},

^{#1} Assistant Professor (Sr.), Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

^{#2} PGStudent, Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India

Abstract— Scheduling of tasks between various processors in a Network-on-Chip is a challenging job. Interconnection of various processing elements with guaranteed traffic permutation and optimality of Scheduling over the Network-On-Chip has been reported in this paper. It supports to reduce the complexity of on chip network implementation. The proposed network topology contains a dynamic probing mechanism to interconnect different processing elements to make a path setup dynamically by using the probe routing algorithm, multistage network topology and communication locality on Network-on-Chip were implemented with reduced area constraints on chip. The communication locality on chip, using an optimized routing algorithm with 8 processing elements communicate over the Network-On-Chip, it provides guaranteed traffic throughput and perfect scheduling. The implementation of scheduling and the proposed network-on-chip also results in reduced number of switches, reduced Cost, improves bandwidth usage, and improves average path length usage with better passable Permutation. It also provides better strategy for scheduling of tasks between various processing elements in the on chip network.

Keywords— Network-On-Chip, Traffic Permutation, multistage network Topology, Scheduling, processing elements.

I.INTRODUCTION

Nowadays there are many different ways of scheduling the work in a Network-on-Chip (NoC). An MPSoC (Multiprocessor System-On-Chip) application uses NoC to make the interconnection between multiple processing elements [1]-[2]. To connect the multiple processing elements by using the network which consists of no of switches to direct the traffic. According to the network topology the connection is directed to switch-by-switch. Finally it reaches its destination processing element. The multistage network topology is used to interconnect a multiple processors and its corresponding memory

Modules in MPSoC.

In network each input sends traffic to only one output and also each output receives traffic from only one input is called as traffic pattern. This traffic pattern is produced over the network. Application aware On-chip routing algorithm is an important feature of MPSoC applications [3]. Permutation traffic is predominantly, occurred in general-purpose MPSoC applications. To reduce the permutation traffic in network is needed [9]. The topology brings out a reduced communication cost. A routing algorithm is used to arrange the processors and memory modules by using round robin technique; it will optimize the routing and produce a minimized total computational time & communication cost [3] [8].

In networks there are three switching systems are available for guaranteed permutation traffic. 1. *Packet Switching Scheme*. 2. *Message Switching Scheme*. 3. *Circuit Switching Scheme*.

Circuit switching is a scheme that directly connects the sender and the receiver in an unbroken path and supports dynamic path-setup scheme under a multistage network topology. The dynamic path-setup will make a path dynamically for conflict-free permuted data. It doesn't use any complex routing algorithm and Queuing buffers, so, it results low cost, low area and less power consumption [3]. Schedule the switches in NoC to control the traffic over the IPC in network.

Multistage network topologies can be classified into many types based on the Network-On-Chip (NoC) applications [4]-[7], Butterfly Network, Flattened Butterfly Network, Flattened Baseline Network, Benes Network, Omega network and Clos Network. Excluding from these multistage networks the Clos network was first conceived, it is a multistage circuit switching consists of cross point switches. [10] – [12]. The direct or regular topologies are

tend to throughput degradation due to the individual source to destination pair stresses [7]. So, indirect multistage topologies are useful for Network-on-Chip applications [13].

II. NETWORK-ON-CHIP DESIGN

A. Multi-stage network topology

The Multi-stage network topology is used to make a connection between various processing elements and memory modules. It also supports various Quality of Service (QoS) [4].

1) Clos network: Clos Network is states as (n, m, p), where, In ingress stage, n is a no of inputs in single switch, m is a no of outputs in single switch and also the no of switches in a middle stage, p is a no of switches in a ingress and egress stages and also the no of inputs and outputs of a single switch in middle stage.

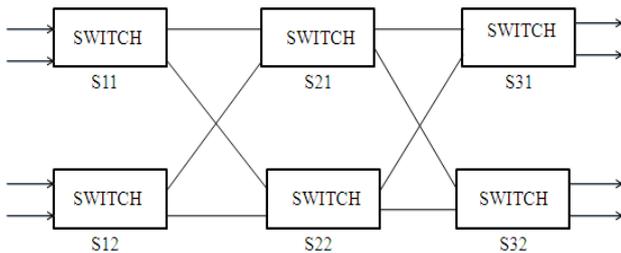


Fig. 1: Basic Clos Network (2, 2, 2)

The above Fig1 shows the Clos network consists of 3-stage network: the ingress stage, middle stage, and the egress stage. Each stage is made up of a number of switches it is just called as crossbar. C (2, 2, 2) has rearrangeable property [12]. Each call entering an ingress crossbar switch can be routed through any of the available middle stage crossbar switches, & it direct to the relevant egress crossbar switch.

2) Rearrangeably nonblocking Clos networks (m ≥ n): The Clos network is rearrangeably nonblocking when it shows m ≥ n, means the network can be nonblocking like a crossbar switch. Where m-no of outputs from the switches in ingress stage & no of inputs in the switches in egress stage. n-no of inputs from the switches in ingress stage & no of outputs in the switches in egress stage.

3. Cost: Cost of the switch is directly propotional to no-of-gates involved, which is roughly propotional to the cross points within a switch.

For ex: The cost of 2X2 switch is 4.

C α N x N where, C-Cost of the switch N-cross points in switch.

B. Switch Architecture

Switch has four components: Control Input (CIs), Control Output (COs), Arbiter, and a Crossbar unit [3]. It can be divided into two functions: 1. Direct –forwarding or Wave pipeline. 2. Control part. Control part contains CIs,

COs and arbiter as shown in Fig2.

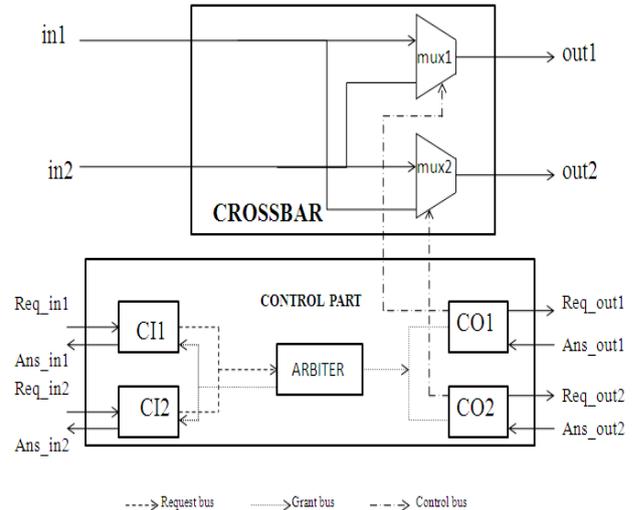


Fig. 2: Switch Architecture

The incoming probe reaches the Control Inputs (CIs) side in the architecture of (2, 2, 2) crossbar switch; it perceives the corresponding output level in Control Outputs (COs) using status bus. ICs sends a request to arbiter through request bus to cross connect the ICs to corresponding OCs. Arbiter is a control part to control the entire communications and it observes the corresponding connection and sends a Grant bus to cross connect the CI to CO. The output of the CO act as selection line of multiplexer that is present in crossbar Switch. Then the data is transferred from one switch to another switch using multistage network. Different probe routing algorithms are applied to the common switch architecture and get the better results [3] [12].

C. Circuit switching scheme

Circuit switching scheme in a network operates almost the same way as the telephone system works. It supports dynamic path-setup scheme under a multistage network topology. This scheme has three phases: setup, transfer and release [3].

D. Dynamic Probing Method

Network-On-Chip is designed using switch to make a path dynamically for guaranteed traffic permutation. The circuit-switching system is based on a backtracking dynamic probing approach for path setup, operates with a source-synchronous wave-pipeline approach [3]. The switch can supports a dynamic probing approach and results bandwidth, area, power and energy in efficient manner.

E. Scheduling Scheme in Network-On-Chip (NoC)

There are many different ways of scheduling tasks used in a NOC. There can be many reasons why an organization has a specific schedule, By scheduling the tasks between multiple processing elements will provides the guaranteed network topology. Types of scheduling-Fixed Priority Scheduling, EDF Scheduling, Round Robin algorithm, Shortest-Execution-Time-First (SETF) & Longest -Time - Execution –First (LETF).

1) *Fixed Priority Scheduling Algorithm:* In Fixed Priority Scheduling Algorithm each process is assigned a priority and the runnable process with the highest priority is allowed to run.

To prevent high-priority processes from running indefinitely; the scheduler may decrease the priority of currently running process at each clock tick. I/O bound processes are given highest priority.

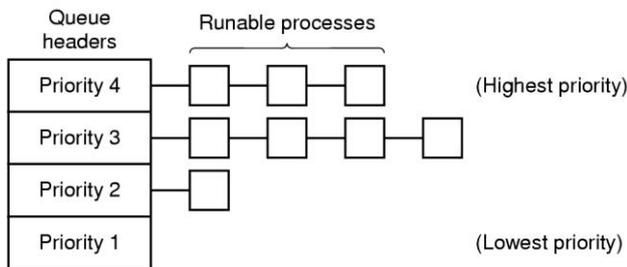


Fig3: Fixed Priority Scheduling

Fig3. Shows the fixed priority scheduling algorithm used in clos network.

2) *EDF Scheduling Algorithm:* EDF Scheduling is a dynamic scheduling. In EDF Scheduling priority is inversely proportional to the deadline value. If the deadline value for a task is high, it gets higher priority.

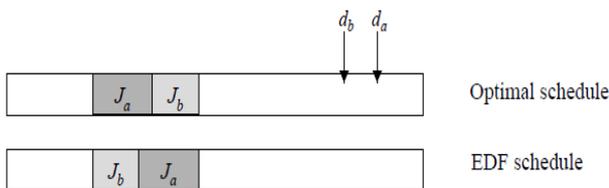


Fig4: EDF Scheduling

Fig. 4 Shows the example for EDF scheduling algorithm. It contains two jobs J_a & J_b , and its deadline value is d_a & d_b by considering the deadline value the priority level should be assigned inversely.

$$D \propto 1/P$$

Where, D-Deadline value
P-Priority level

F. Master-Slave Configuration

It is used for a load sharing purposes, in the Inter-Processor Communication (IPC). The process has unidirectional control over one (or) more other devices. In some systems a master is elected from a group of eligible devices, with the other devices acting in the role of slaves.

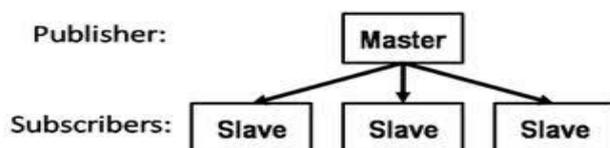


Fig. 5 Master-Slave Configuration

IV IMPLEMENTATION RESULTS & DISCUSSION

The clos network is designed with 8switches. A single switch is designed based on the dynamic probing mechanism and apply the scheduling algorithm to interconnect various processing elements efficiently. Synthesis and implementation is done using Modelsim 6.4, XSE Design Site 14.3 and Xilinx Plan Ahead 14.3. Get the results for Clos Network with 8 switches and Simple Processing Elements. The network is designed for clos network (2, 2, 2) stage, scheduling the task between various processing elements using scheduling algorithms, by giving the clock to every stage it produces the results. By crossing the clos network the input data is transferred from sender to receiver.

The below Fig6. Shows the FPGA editor layout for Clos network (2, 2, 2) interconnects two processing elements using shared memory concept.

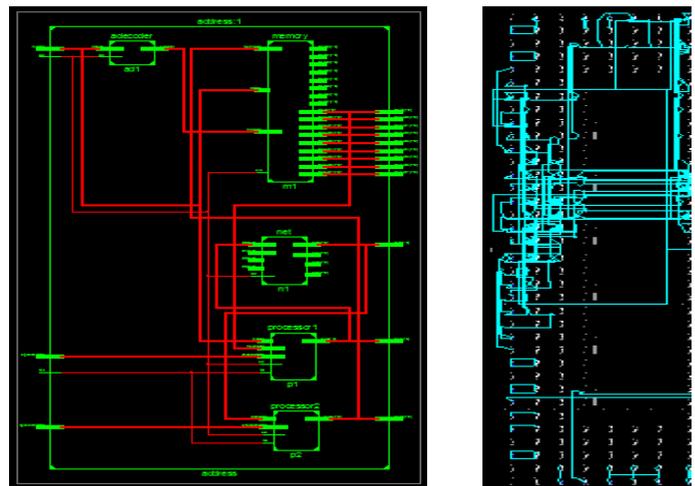


Fig.6: RTL & FPGA Editor View of Clos Network (2, 2, 2) between two PEs using Shared memory

The resource usage of LUTs and Slices are gradually decreased in clos network (2, 2, 2) shared memory concept. So, by considering the resource utilization the shared memory concept is efficient in Network-On-Chip (NoC). The FPGA editor layout shows the resource usage implemented by VIRTEX 5 kit. Scheduling the tasks over the network-on-chip is shown in Fig6. It will explain the resource utilization of the processes.

In Fig7. The RTL Schematic diagram shows the modules used in EDF Scheduling to schedule the tasks in various processing elements by using the EDF scheduling algorithm with Master-Slave Configuration. It contains one Master processing element and 7 simple processing elements, this master processing element gets deadline value for all other processing elements and by considering the deadline values the master processing element gives the priority level to other processing elements then the corresponding processing element produced the result. The results based on the resource utilization. No -of Flip-Flops and LUTs used in EDF scheduling with master-slave configuration is less when compare to Fixed Scheduling algorithm.

Implementation of Scheduling Tasks between multiple Processing Elements....

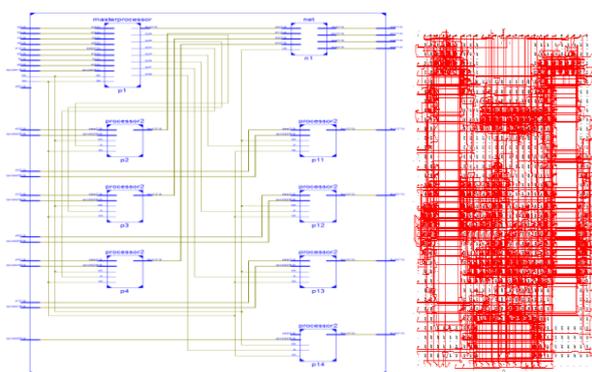


Fig. 7: RTL & FPGA Editor view of Clos Network (2, 2, 2) using EDF Scheduling algorithm with Master-Slave configuration.

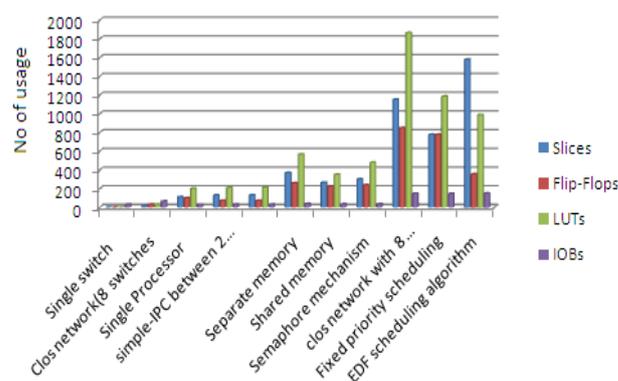


Fig. 8: Graphical representation for Clos network (2, 2, 2) using shared memory concept

TABLE I

COMPARISON BASED ON RESOURCE UTILIZATION

Hardware Module	Slices	Flip-Flops	LUTs	IOBs	Power supply(W)	No of clock cycles
Clos network with Separate memory	369	288	567	38	0.452	7
Clos network with Shared memory	264	224	360	34	0.450	4
Fixed Priority scheduling	776	776	1,184	146	0.463	5
EDF scheduling algorithm	638	606	1,306	192	0.461	7
EDF scheduling using Master-Slave configuration	420	602	1,142	189	0.460	3

The graphical representation of area occupied by the resources in the Clos Network, Processing Elements is shown in Fig8. The graph shows the resources used for single switch, a Network, simple Inter-process communication established between two Processing Elements and the overall design. The graph shows that the resource used in Clos network (2, 2, 2) with shared memory concept is less when compared to Clos network (2, 2, 2) with separate memory concept.

Table I. shows the comparison based on Resource Utilization in clos network (2, 2, 2) using separate memory, shared memory, Fixed priority scheduling algorithm, EDF scheduling algorithm & Master-Slave configuration in EDF scheduling. Compare to separate memory concept Clos network (2, 2, 2) using shared memory concept will use less resource and less power. To schedule the tasks between various processing elements is done by EDF scheduling algorithm, EDF scheduling with Master-Slave configuration & fixed priority Scheduling algorithm.

Comparison based on the resource utilization, No-of clock cycles & power supply is shown in Table I. It displays that the EDF scheduling with Master-Slave configuration used less resource and less power supply,

when compared to other 2 scheduling algorithms. It also displays the no-of-clock cycles used to get the final output from the processing elements. From that EDF scheduling with Master-Slave configuration takes only 3cycles to give the result.

V. CONCLUSION

As inferred in Table I, it indicates the usage of hardware modules, no-of clock cycles used to get output, usage of power supply is designed using HDL code in modelsim 6.4a and implemented using XSE Design suite14.3. The HDL was developed for a single switch, clos network c (2, 2, 2) with six switches, two simple processors, an IPC between 2 processors. When compared to other modules, the processor occupies more on chip area. Usage of shared memory drastically reduces the number of slices, flip-flop, LUTs, IOB's in the design of Network-on-Chip. Apply the Scheduling algorithm to schedule the tasks between various processing elements. It is proposed to implement more number of PEs in the Network-on-Chip, with various other topologies, integrating the IPC mechanisms between the PEs in the network.

REFERENCES

- [1] Phi-Hung Pham, Junyoung Song, Jongsun Park, and Chulwoo Kim, "Design and Implementation of an On-Chip Permutation Network for Multiprocessor System-On-Chip," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol.21, no.1, January 2013.
- [2] Ville Rantala, Teijo Lehtonen, and Juha Plosila, "Network-On-Chip Routing Algorithm," *University of Turku*, August 2006.
- [3] Phi-Hung Pham, Jongsun Park, Phuong Mau, and Chulwoo Kim, "Design and Implementation of Backtracking Wave-Pipeline Switch to Support Guaranteed Throughput in Network-on-Chip" *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol.20, no.2, February 2012.
- [4] Victor Lopez Debuen, "Multistage interconnection network in Multiprocessor system. A simulation study," vol.11, no.3, 1987.
- [5] N. Michael, M. Nikolov, A. Tang, G. E. Suh, and C. Batten, "Analysis of application-aware on-chip routing under traffic uncertainty," in *Proc. IEEE/ACM Int. Symp. Netw. Chip (NoCS)*, 2011, pp. 9–16.
- [6] Carlo Condo, Maurizio Martina, and Guido Masera, "Turbo NOC: a framework for the design of network-on-chip-based turbo decoder architectures," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol.57, no.10, pp. 1766–1769, October 2010.
- [7] Mahsa Moazez, Farshad Safaei, and Majid Rezazadeh, "Design and Implementation of Multistage Interconnection Networks for SOC Networks," *International Journal of Computer Science, Engineering and Information Technology (IJCEIT)*, vol.2, no.5, October 2012.
- [8] S. Subha, "An Optimized Algorithm for Network on Chip," *International Journal of Computer Applications (0975 – 8887)* Vol.3, no.12, July 2010.
- [9] H. Moussa, A. Baghdadi, and M. Jezequel, "Binary de Bruijn on-chip network for a flexible multiprocessor LDPC decoder," in *Proc. ACM/ IEEE Design Autom. Conf. (DAC)*, 2008, pp. 429–434.
- [10] H. Moussa, O. Muller, A. Baghdadi, and M. Jezequel, "Butterfly and Benes-based on-chip communication networks for multiprocessor turbo decoding," in *Proc. Design, Autom. Test in Euro. (DATE)*, 2007, pp. 654–659.
- [11] Nikolay Kavaldjiev, Gerard J. M. Smit, Pascal T. Wolkotte, and Pierre G. Jansen, "Routing of guaranteed throughput traffic in a network-on-chip," *University of Twente*, 2005, Netherlands.
- [12] Y. Yang and J. Wang, "A fault-tolerant rearrangeable permutation network," *IEEE Trans. Comput.*, vol. 53, no. 4, pp. 414–426, Apr. 2004.
- [13] S. R. Vangal, J. Howard, G. Ruhl, S. Dige, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, "An 80-tile sub-100-w TeraFLOPS processor in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no.1, pp. 29–41, Jan. 2008.
- [14] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*: San Francisco, CA: Morgan Kaufmann, 2004.
- [15] Jingcao Hu and Radu Marculescu, "Energy-Aware Communication and Task Scheduling for Network-on-Chip Architectures under Real-Time Constraints", *Research supported by NSF CCR and DARPA/Marco Gigas-cal e Research Center (GSRC)*, 2004.