

**Improved Data Efficiency of Programmable Arbiter Based On-Chip Permutation
Network for MPSOC**P.LAKSHMI KANTH¹, L.S.DEVARAJ²¹Pg Scholar, Vlsi System Design , Intellectual Institute Of Technology, Ap, India.²Assistant Professor, Intellectual Institute Of Technology, Ap, India.

Abstract--This paper presents the design of a novel OCP network to support guaranteed traffic permutation in multiprocessor SOC applications. The proposed network employs a pipelined circuit-switching approach combined with a dynamic path-setup scheme under a multistage network topology. The dynamic path-setup scheme enables runtime path arrangement for arbitrary traffic permutations. The existing system having only fixed priority logic scheme for dynamic path set up. In this paper we proposed a new PAB based priority logic to rectify the drawbacks in previous arbiter system in proposed OCP network. The PAB contains F-Priority, RR-Priority, D-Priority logics. This circuit-switching approach offers a guarantee of permuted data and its compact overhead enables the benefit of stacking multiple net-works. The proposed on chip network improves the efficiency. Finally implemented the design using Xilinx ISE 12.1 software on FPGA Spartan 3E family kit, NEXYS 2 board and showed the synthesis result and power result. The Proposed system OCP network with PAB improves the power, delay and improves the data efficiency.

Keywords: SOC, PAB, OCP, FPGA, Circuit Switching, Dynamic Path Setup, F-Priority, RR-Priority, D-Priority.

I. INTRODUCTION

For applications of parallel processing, scientific computing, and so on, In a present trend of multiprocessor system on chip (MPSoC) design are interconnected with on-chip networks is currently emerged[1-6]. Permutation traffic, a traffic pattern in which each input sends traffic to exactly one output and each output receives traffic from exactly one input, is one of the important traffic classes exhibited from on-chip multi-processing applications[7-8]. Standard permutations of traffic occur in general-purpose MPSoCs, for example, polynomial, sorting, and fast Fourier transform (FFT) computations cause shuffled permutation, whereas matrix transposes or corner-turn operations exhibit trans- pose permutation[6]. Recently, application specific MPSoCs targeting flexible Turbo/LDPC decoding had developed, and they exhibit arbitrary and concurrent traffic permutations due to multi-mode and multi-standard feature[3-5]. In addition to that many of the MPSoC applications (e.g., Turbo/LDPC decoding [3-5]) compute in real-time, therefore, guaranteeing throughput is critical for such permutation traffics. Most on-chip networks in practice are general-purpose and use routing algorithms such as minimal adaptive routing and dimension-ordered routing. To support permutation traffic patterns, on-chip permutation networks using application-aware routings are needed to achieve better performance compared to the general-purpose networks[8]. These application-aware routings are configured before running the applications and can be implemented as source routing or distributed routing. But such application-aware routings cannot handle the dynamic changes of a permutation pattern efficiently, which is described in many of the application phases[8].

The difficulty hold in the design effort to compute the routing to the permutation changes in runtime, as well as to guarantee [9] the permuted traffics for efficient support. This will become a great challenge when these permutation networks needed to implemented under very limited on-chip power and area relatedly reduce level. Reviewing on-chip permutation networks (supporting either full or partial permutation) with regard to their implementation shows that most the networks employ a packet-switching mechanism to deal with the conflict of permuted data [3-6]. In this paper we present a new hardware architecture figure 1 which is based on to improve the permutation traffic efficiency. In arbiter system we use the programmable arbiter priority (PAB) logics to produce the data under priority if at each switch no.of inputs came at a time. Actually in existing system only use the fixed priority arbiter for priority based data transferring. In this paper the new proposed arbiter is with programmable logics. It provides three priority logics according to requirement priority that three priority logics are F-priority (Fixed Priority), Round Robin priority (RR-priority), Dynamic Priority(D-priority). In D-priority is mixing operation of both F-priority and RR-priority to speed up the parallel operation.

So the below sections illustrates the operation of proposed architectures and switching activities well in manner. Section II describes the top of architecture and interconnections for circuit switching activity .Section III describes about the dynamic path set up scheme for programmable data transfer according to switching operations. Here itself describes about the PAB (Programmable arbiter) as well. Section IV illustrates the Implementation and results .Section V illustrates the conclusion of project paper.

II. PROPOSED ON CHIP NETWORK

As per explanation motivated in section I is the key idea to design the proposed on chip network based on a pipelined circuit switching approach with dynamic path-setup scheme supporting runtime path arrangement. In this section II it discuss the about path set up scheme and network topology well to understand for design OCP(on chip permutation) network. And later design of switching nodes presented well to understand.

A. On-Chip Network Topology

A family of multistage networks is Clos network, it is applied to build scalable commercial multiprocessors with number (thousands) of nodes in macro systems [7-11]. A three-stage typical Clos network is defined a $C(m,n,p)$ where the m number of inputs represents in each of first-stage switches and n is the number of second-stage switches. In order to support a parallelism degree of 16 inputs as in most practical MPSoCs [3-5], for the designed network $C(4,4,4)$ we proposed to use as a topology (see Fig. 1). This network has a property of rearrangeable [11] that all possible permutations can realize between its input and outputs. The choice of the three-stage Close network with a modest number of middle-stage switches is to minimize implementation cost, whereas it still enables a property rearrangeable for the network.

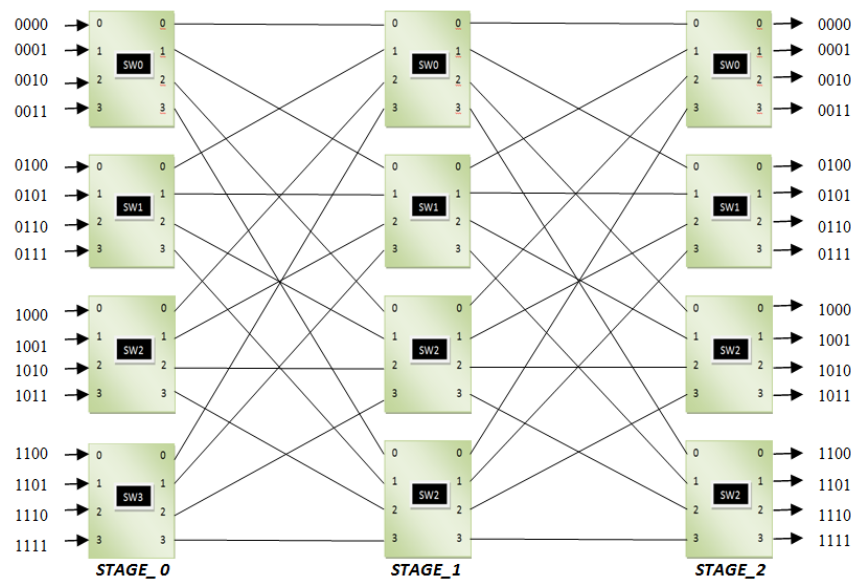


Fig1. Proposed OCP network of Corcuit switvhing mechanism Architecture.

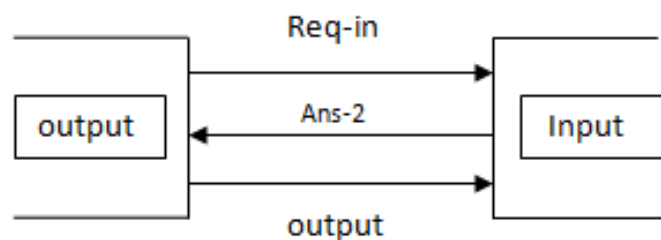


Fig2. The Mechanism Interface Path Diagram Of Port to Port

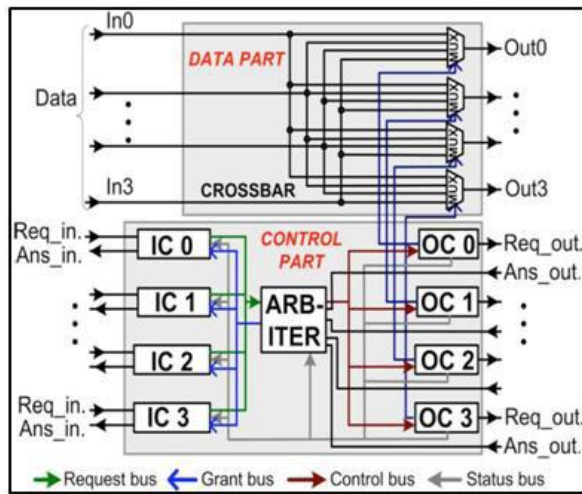


Fig3. Common switch architecture.

Req	1: Setup	0: Idle
1.	00: Idle	10: Back
2.	01: Ack	11: Nack

Table1. Switch Activity

Pipelined circuit-switching scheme is designed and introduced for use with the proposed OCP network. This scheme has three phases: the setup, transfer, and release [2] and [9]. A dynamic path-setup scheme which supporting the runtime path arrangement occurs in the phase setup. To support this circuit-switching scheme, a switch-by-switch interconnection and with its handshake signals are proposed, as shown in of the handshake includes a 1-bit Request (Req) and a 2-bit Answer (Ans). Req=1 is used when a switch requests an idle link leading to the corresponding downstream switch in the phase setup. Req=1 is also kept data transfer along with during the set up path. Req=0 denotes that the switch releases the link which have to be occupied. This code is also used in for both the setup and the release phases. Ans=01(ack) means that the destination is ready to receive data from the source. When Ans=01 the propagates back to the source, it denotes that the path is set up, then a data transfer can started immediately. An Ans=11 is reserved for end-to-end flow control when the receiving circuit is not at ready to receive the data due to it is being busy with other tasks, or overflow at the receiving buffer, etc. An Ans=10 (Back) means that the link is blocked. This Back code is used for a back pressure flow control of the dynamic path-setup scheme, which is discussed in the following subsection.

B. Dynamic Path Setup Scheme to Support Path Arrangement

A dynamic path set up scheme is the important point for the proposed design which to support for runtime path arrangement when the permutation is changed .In this, each and every path setup , starts from an input and find a path for leading to its corresponding output, with support based on dynamic probing mechanism. The probing concept is introduced in works in [2], [9], which a probe (or setup flit) is dynamically sent under a routing algorithm in order to establish a path towards the destination. A question is that can the proposed EPB-based path setups used with the Clos realize all possible full permutations between its A question is that can the proposed EPB-based path setups used with the Clos network $C(m,n,p)$ realize all possible full permutations between its inputs and outputs? As proofed in works [10] and [11], the three-stage Clos network $C(m,n,p)$ is rearrangeable if m greater than or equal to n . In the proposed network of $C(m,n,p)$ $m=4, n=4, p=4$, so it is rearrangeable. There always exists an available path from an idle input leading to an idle output. By the Exhaustive Property of EPB as proofed in work [11], the EPB-based path setup completely searches all the possible paths within the set of path diversity between an idle input and idle output. Directly applying the Exhaustive Property of the search into rearrangeable $C(4,4,4)$.

C. Switch Nodes Topology

Three kinds of switches are designed for the proposed on-chip net- work. These switches are all based on a common switch architecture shown in Fig. 3, with the only difference being in the probe routing algorithms. This common architecture

has basic components: INPUT CONTROLS (ICs), OUTPUT CONTROLS (OCs), an ARBITER, and .The ARBITER has two functions: first, cross-connecting the Ans_Outs and the ICs through the Grant bus, and second, as a referee for the requests from the ICs. When an incoming probe arrives at an input, the corresponding IC observes the output status through the Status bus, and requests the ARBITER to grant it access to the corresponding OC through the Request bus. When accepting this request, the ARBITER cross-connects the corresponding Ans_Out with the IC through the Grant bus with its first function. With the second function, the ARBITER, based on a pre-defined priority rule, resolves contention when several ICs request the same free output. After this resolution, only one IC is accepted, whereas the rest are answered as facing a blocked link (i.e., similar to receiving an Ans = Back).The IC is implemented with finite-state machine (FSM). The probe routing algorithm and the operation of the switches are controlled according to this FSM implementation in the ICs [9].

The three routing algorithms for the switches in the first, the second, and the third stages are detailed in Fig. 4. In the first stage, the switch tries the free outputs in a non-repetitive manner (e.g., outputs 0 $\square$ 1 $\square$ 2 $\square$ 3). This implementation avoids repetitively searching the same path that may result in a live-lock. The second- and third-stage switches rely on the two most significant bits D3, D2 and the two least significant bits D1, D0 of the destination address, respectively, to route the probe. As can be seen from Fig. 4, depending on the availability of the desired output or the feedback (i.e., the signal Ans) from the downstream switch, the IC in a given switch will change its FSM state and reply to the upstream switches accordingly. The OCs work as re-timing stages for the commands from ARBITER placed on the Control bus and control the CROSSBAR. The CROSSBAR is a 4X4 full-connecting matrix designed with output multiplexers.

The ICs and the ARBITER are clocked ones with the rising and the falling edges of the clock, respectively. By this implementation, probing is processed dynamically by the switch in basis one clock cycle. This meets the target of designing the circuit-switched switches for to support EPB-based path setup in C (4,4,4) network. To validate if the designed network works as desired, a testbench is applied for to test the capability of realizing full permutation with six-teen path setups. To avoid a path setup interfering with others during the search and incurring a rearrangement of existing paths, a delay is set between the path setups launched one-by-one in a sequence in the test bench. This is to ensure that the previous path setup is completed before a new one is launched.

D. Arbiter Priorities

In this proposed network at arbiter section we proposed the three priority logics Fixed priority (F-priority), RR-priority (Round Robin Priority) and D-priority (Dynamic Priority) logics which are programmable according to the priority requirement. when —00 the arbiter acts like F-priority and when —01 it acts like RR-Priority and when —11 it acts like D-priority. we know when compared to fixed priority Round Robin is efficient one. So in this paper I proposed the programmable arbiter with three priorities act. I generated output for the RR-priority in this paper.

III. IMPLEMENTATION, SIMULATION AND SYNTHESIS RESULT

This proposed architecture as shown in fig 1 is designed using verilog language and it is simulated in Modelsim software for simulation result. The below which have showed fig 4, fig 5, fig 6 are the results of simulation. In paper we given —Req enable pins at four addresses according to the above fig 2 switch activity with dynamic path set scheme the switches arranged the path to transfer the data from source to destination. In fig 4 shows the Req pins enable at four addresses and data given at that point and fig 5 result shows the switch activity enable pins like Ans_in, Ans_out and fig 6 shows the destination address result at stage 2 from source.

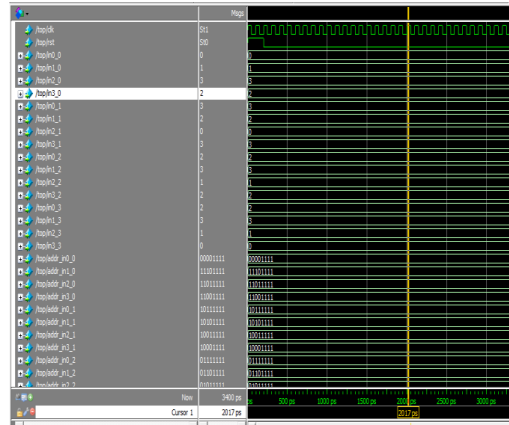


Fig4. Simulation result part1 for Proposed OCP network.

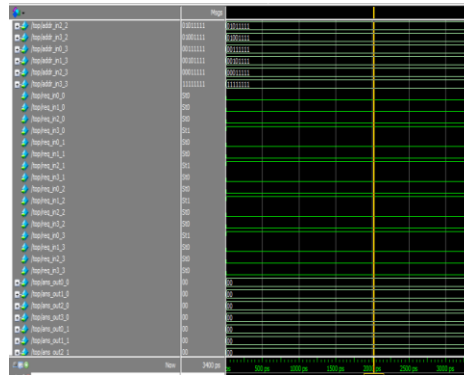


Fig5. Simulation Result Part2 for Proposed OCP network.

The fig7 shows the NEXYS 2 FPGA kit of SPARTAN 3E family .On this kit implemented the above OCP architecture. The output LED,s which are blinked is the output pins of stage 2 and the switches which are enabled acts like Req's ,when the second switch from left is enabled ,ie; the Req pin at port of SW0 is enbled and data is transferred from source to destination depends the switching data path set up mechanism,here the output is came stage 2 switch 0 at that point data is —10101. And by Table II gives the synthesis report of architecture according to FPGA.In this FPGA we use device X3cs1200E and package is FG320 and Device speed is -5.



Fig7. FPGA Implemented result of proposed OCP Network.

Name	Used	Available	Utilization
No.of Slices	1326	8672	15%
No.of 4 input LUTs	2483	17344	14%
No.of IOBs	146	250	58%
Total delay	12.624ns		

Table II. Synthesis Report.

IV. CONCLUSION

This paper has presented an OCP network design supporting traffic permutations in MPSoC applications. By using a circuit-switching approach combined with dynamic path-setup scheme under a Close network topology, the proposed design offers arbitrary traffic permutation in runtime with compact implementation overhead. Design is implemented using Xilinx ISE 12.1 on FPGA Board of Spartan 3E family , NEXYS 2 kit and obtained the synthesis result regarding delay and do power analysis regarding power by that proved that efficiency is improved when compared to existing systems.

V. REFERENCES

- [1] C. Neeb, M. J. Thul, and N. Wehn, —Network-on-chip-centric approach to inter leaving in high throughput channel decoders, in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2005, pp. 1766–1769.
- [2] 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.
- [3] 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.
- [4] S. Borkar, —Thousand core chips—A technology perspective, in Proc. ACM/IEEE Design Autom. Conf. (DAC), 2007, pp. 746–749.
- [5] P.-H. Pham, P. Mau, and C. Kim, —A 64-PE folded-torus intra-chip communication fabric for guaranteed throughput in network-on-chip based applications, in Proc. IEEE Custom Integr. Circuits Conf. (CICC), 2009, pp. 645–648.
- [6] decoding, in Proc. Design, Autom. Test in Euro. (DATE), 2007, pp. 654–659.
- [7] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, 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.
- [8] D. Ludovici, F. Gilabert, S. Medardoni, C. Gomez, M. E. Gomez, P. Lopez, G. N. Gaydadjiev, and D. Bertozzi, —Assessing fat-tree topologies for regular network-on-chip design under nanoscale technology constraints, in Proc. Design, Autom. Test Euro. Conf. Exhib. (DATE), 2009, pp. 562–565.
- [9] Y. Yang and J. Wang, —A fault-tolerant rearrangeable permutation network, IEEE Trans. Comput., vol. 53, no. 4, pp. 414–426, Apr. 2004.
- [10] Y. Yang and J. Wang, —A fault-tolerant rearrangeable permutation network, IEEE Trans. Comput., vol. 53, no. 4, pp. 414–426, Apr. 2004.
- [11] P. T. Gaughan and S. Yalamanchili, —A family of fault-tolerant routing protocols for direct multiprocessor networks, IEEE Trans. Parallel Distrib. Syst., vol. 6, no. 5, pp. 482–497, May 1995.

ACKNOWLEDGMENT

FIRST AUTHOR: P.LAKSHMI KANTH received the Bachelors degree in Electronics and Communication Engineering in the year 2013 and pursuing Masters degree in VLSI System design from Intellectual Institute Of Technology. Area of interests includes Communication and VLSI Design.

SECOND AUTHOR: L.S.DEVARAJ received the Bachelors degree in Electronics and Communication Engineering and Masters degree in VLSI. Currently working as Head of the Department for Electronics and Communication Engineering, Intellectual institute of technology, Ananthapuramu .Having five years of teaching experience. Area of interest includes VLSI design, vhdl and verilog .