International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 9, September 2018)

An On-Chip Traffic Permutation Network for Multiprocessor System-On-Chip

Sheetal waghmare
Dept of Electronics and Communication Engineering
VTU Regional office, Gulbarga
sheetal.waghmare108@gmail.com

Prof. Seema Deshmukh
Dept of Electronics and Communication Engineering
VTU Regional office, Gulbarga
seema_s_d@yahoo.co.in

***Abstract*— The Networks on Chip (NoC), due to their flexibility, scalability and high bandwidth features they are considered as on-chip communication fabrics for future multiprocessor system-on-chips(MPSOCs).In this paper the design of a network on chip to support a guaranteed throughput is explained. The close nework topology is used with the three stages of switches and each stage having three switches of 4 ports of inputs and outputs. This network to support guaranteed traffic permutation in multiprocessor system-on-chip 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 circuit-switching approach offers a guarantee of permuted data and its compact overhead enables the benefit of stacking multiple networks. Network on chip or network on a chip (NoC or NOC) is a communication subsystem on an integrated circuit, typically between IP cores in a system on a chip (SoC). NoCs are an attempt to scale down the concepts of large scale networks, and apply them to the embedded system on chip domain. Traditional busses are not suitable to the system on chip domain. The fundamental unit of building a Network on Chip is the router, it directs the packets according to a routing algorithm to the desired host. In this project, switch is designed using VERILOG language and simulated with the help of Integrated software environment (ISE12.2) , and then (3x3) clos topology network is designed.**

***Index Terms*—Guaranteed throughput, multistage interconnection network, network-on-chip, permutation network, pipelined circuit-switching,traffic permutation.**

I. INTRODUCTION

Now a days there is a trend that on-chip networks and multi processor system on chip are being interconnected for currently emerging applications such as parallel processing, scientific computing and many more. [1]– [6]. One of the traffic patterns in which each input sends traffic to only one input and each output receives traffic from only one input is considered as the Permutation traffic is one of the important traffic classes exhibited from on-chip multiprocessing applications [7], [8]. In general-purpose MPSoCs, Standard permutations of traffic occur for example, *shuffled permutation is caused by* polynomial, sorting,and fast Fourier transform (FFT) computations, whereas *transpose permutation is caused by* matrix transposes or corner-turn operations [6]. Recently, flexible Turbo/LDPC decoding have been developed which is application specific MPSoC, because of multi mode and multi standard feature they exhibit arbitrary and concurrent traffic permutations [3]–[5]. Thus , many of the MPSoC applications like Turbo/LDPC decoding [3]–[5] compute in real-time, and giving guaranteed throughput i.e., data lossless, predictable latency, guaranteed bandwidth, and in-order delivery is critical for such permutation traffics.

The on-chip networks mainly are general purpose and dimension-ordered routing and minimal adaptive routing are the routing algorithms mainly used. To achieve better performance compared to the general-purpose networks, on-chip permutation networks using application-aware routings are needed and it supports the permutation traffic patterns [8]. These application-aware routings can be implemented as source routing or distributed routing and are configured before running the applications. These application-aware routings cannot efficiently handle the dynamic changes of a permutation pattern, which is exhibited in many of the application phases [8]. There is a difficulty in the design effort to compute the routing to support the permutation changes in *runtime*, as well as to *guarantee* the permutated traffics.This becomes a great challenge when these permutation networks need to be implemented under very limited on-chip power and area overhead.

The on-chip permutation networks supports either full or partial permutation. while considering their implementation many networks use packet-switching mechanism to deal with the conflict of permuted data [3]–[6]. These can be implemented using first-input first-output (FIFO) queues for the conflicting data [3], [5],[6], or they may use time-slot allocation within he system along with the costly routing stages [5], or a complex routing which employs the deflection technique that avoids buffering of the conflicting data [4]. The on-chip implementation has different effects for the different choices of network design factors, i.e., topology, switching technique and the routing algorithm.

The *topology* are regular direct topologies, like mesh and torus [2], [3], [6], are intuitively feasible for physical layout in a 2-Dchip.where as the indirect topologies like Benes or Butterfly [4], [5] requires the high wiring irregularity and the large router radix and they pose a challenge for physical

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 9, September 2018)

implementation [10].Where as intensive load on individual source-destination pairs for an arbitrary permutation pattern stresses the regular topologies and it leads to throughput degradation [7]. Thus the indirect topologies with multistage neworks are preferred for on-chip traffic-permutation demanding applications [4],[5].

Considering the *switching technique*, in packet switching for the queuing buffers (FIFOs) there is requirement of an excessive amount of on-chip power and area [3]–[6].

Considering the *routing algorithm*,when we compare the the deflection routing [4] and the minimal routing,the deflection routing is not energy-efficient due to the extra hops needed for deflected data transfer [2], [3], [5].

II. PROPOSED ON-CHIP NETWORK DESIGN

As mentioned in Section I, the proposed on-chip network design is based on a dynamic path-setup scheme which provides run time path arrangement with a pipelined circuit-switching approach. Now lets have a look at , the network topology and then the dynamic path-setup scheme and switching nodes designing.

A. 3 Stage CLOS Network Topology

We are applying Clos network topology which is a family of multistage networks, to build scalable multiprocessors in which the macrosystems have thousands of nodes [7]. A Clos network with three-stage is defined as C(n,m,p). Here p is the number of first-stage switches and n is the number of inputs in each of first stage switches and m is the number of second-stage switches.

As in most practical MPSoCs [3]–[5], we are using C(4,4,4) as a topology to support a parallelism degree of 16 for the designed network (see Fig. 1). This network can realize all possible permutations between its input and outputs as it has a rearrangeable propert and to minimize implementation cost, we have employed it with a modest number of middle-stage switches .

we are using the pipelined circuit-switching scheme having three phases: the *setup*, the *transfer*, and the *release* In the setup phase a dynamic path-setup scheme which supports the runtime path arrangement occurs.A switch-by-switch interconnection with its handshake signals is proposed, as shown inFig. 2 to support this circuit-switching scheme.The handshake signals have the following bit format, a 1-bit *Request (Req)* and a 2-bit *Answer (Ans)*. Whenever switch requests an idle link leading to the corresponding downstream switch in the setup phase, Req=1 is used.

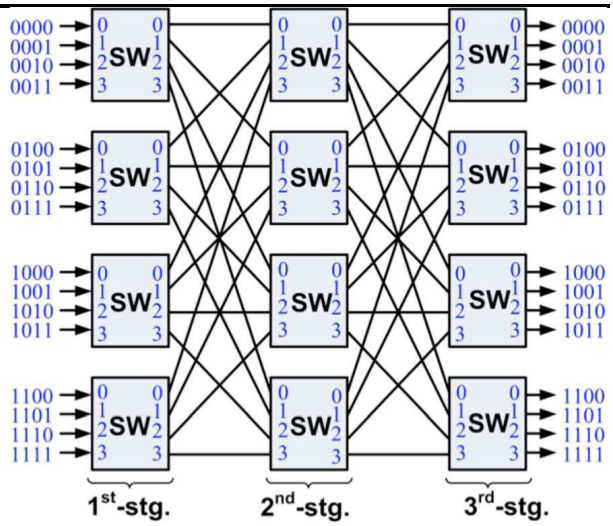

Fig. 1. Proposed on-chip network topology with port addressing scheme.

In this paper there is a novel silicon-proven design for an on-chip permutation network,which under arbitrary permutation also supports the guaranteed throughput of permutated traffics.in this paper we are employing multistage network topology with circuit-switching mechanism and a dynamic path-setup scheme. The dynamic path setup helps to find the runtime path arrangement to have a conflict-free permuted data. As there are pre-configured datapaths thus they enable a throughput guarantee. Thus stacking multiple networks to support concurrent permutations in runtime is feasible and by removing the excessive overhead of queuing buffers, a compact implementation is achieved.

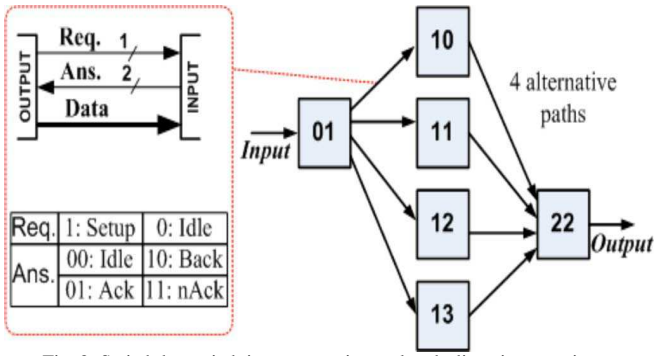

Fig. 2. Switch-by-switch interconnection and path-diversity capacity.

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 9, September 2018)

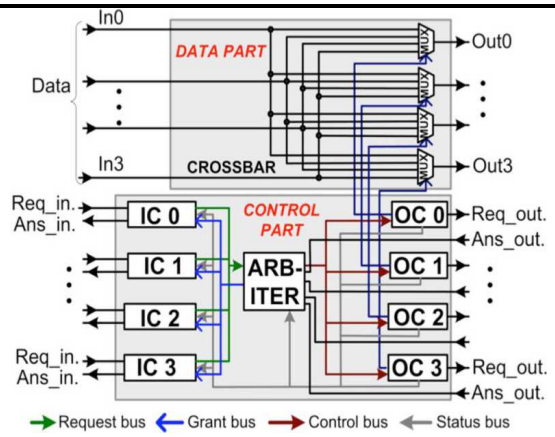

Fig. 3. Common switch architecture.

This paper explains the proposed on-chip network design with its dynamic path-setup scheme to support runtime path arrangement in the section II and finally conclusion.

During data transfer along the set up path then Req=1 is used whereas when the switch releases the occupied link Req =0 is used. In both the setup and the release phase this code is used. When the destination is ready to receive data from the source Ans=01(Ans) is used. When the path is set up the Ans = 01 propagates back to the source enabling the data transfer immediately.While considering the end to end data flow an Ans = 11(nAck) indicates that receiving circuit is not ready to receive data due to overflow at the receiving buffer, or it is busy with other tasks, etc

An Ans=10 (*Back*) is used for a backpressure flow control of the dynamic path-setup scheme means that the link is blocked.

B. Run Time Path arrangement with Dynamic Path Setup scheme

When the permutation is changed a dynamic path-setup scheme is used in the proposed design to support a runtime path arrangement. A *dynamic probing* mechanism is used to enable the path setup, where the path is chosen from the requested input to the requested output.A probe (or setup flit) is dynamically sent under a routing algorithm to establish a path towards the destination this concept of probing is introduced in works [2], [9], in which. Exhausted profitable backtracking (EPB) [12] is proposed to use to route the probe in the network work.

In the proposed design for the path-setup scheme, it is obvious that the path arrangement for full (as well as partial) permutation can always be realized using the EPB-based algorithm. As designed in this network,based on theEPB based path set up scheme each input sends a probe having a 4-bit output address to find an available path leading to the requested output. During the search,if there is a free link the probe moves forwards and when it faces a blocked link it moves backwards. Because of the non-repetitive movement, the probe finds an available path between the input and its corresponding idle output. The EBP-based path-setup scheme has a set of probe routing algorithms as described in Fig. 4. The following example explains how the path setup works to find an available path by using the set of path diversity shown in Fig. 2.

It is assumed that a probe from a source (e.g., an input of switch 01) is trying to set up a path to a target destination (e.g., an available output of switch 22). First, the probe will non-repetitively try

paths through the second-stage switches in the order of 10=>11=>12=>13. Assuming that the link 01-10 is available, the probe first tries this link (Req =1) and then arrives at switch 10.

• If link 10-22 is available, the probe arrives at switch 22 and meets the target output. An Ans=Ack then propagates back to the input to trigger the transfer phase.

• If link 10-22 is blocked, the probe will move back to switch 01(Ans=Back) and link 01-10 is released (Req=0). From switch 01, the probe can then try the rest of idle links leading to the second-stage switches in the same manner. By means of moving back when facing blocked links and trying others, the probe can dynamically set up the path in runtime in a conflict-avoidance manner.

C. Switching Node Designs

Three kinds of switches are designed for the proposed on-chip network.These switches have common switch architecture shown in Fig. 3, with different probe routing algorithms for each stage. This common architecture has basic components: INPUT CONTROLs (ICs), OUTPUT CONTROLs (OCs), an ARBITER, and a CROSSBAR. To save on wiring costs incoming probes in the setup phase can be transported through the data paths.

The ARBITER has two functions: firstly,it cross-connects the Ans_Outs and the ICs through the *Grant bus*, and second,it acts as a referee for the requests from the ICs. As the incoming probe appears 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.The second function of the ARBITER is that based on a pre-defined priority rule, when several ICs request the same free output it resolves contention. 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 ARBITER has two functions: firstly,it cross-connects the Ans_Outs and the ICs through the *Grant bus*, and second,it acts as a referee for the requests from the ICs. As the incoming probe appears 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. The second function of the ARBITER is

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 5, Issue 9, September 2018)

that based on a pre-defined priority rule, when several ICs request the same free output it resolves contention. 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 Input Control has the finite-state machine (FSM) and this FSM controls the probe routing algorithm and the operation of the switches. Fig 4 explains the probe routing algorithms and their corresponding handshake signals. For supporting the probing path setup, ICs are implemented with different probe routing algorithms depending on its switch stage. The 4-bit address of the destination, i.e D3,D2,D1,D0 will be there in the probe(see Fig. 1 for the addressing scheme). In fig 4 the three routing algorithms for the switches in the first, the second, and the third stages are being explained. In the first stage, the switch tries the free outputs in a non-repetitive manner (e.g., outputs 0). 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 signification 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 4_4 full-connecting matrix designed with output multiplexers. The ICs and the ARBITER are clocked with the rising and the falling edges of the clock, respectively. By this implementation, probing is dynamically processed by the switch in one clock cycle basis. As denoted in Fig. 3, the control part of switches performs the dynamic EPB-based path setup, whereas the data part simply provides configured paths for guaranteed circuit-switched data. This meets the target of designing the circuit-switched switches to support EPB-based path setup in C(4,4,4) network. From fig 2 we can calculate that based on the path-diversity graph , the worst-case path setup needs 14 steps (hops) of moving its probe back and forth to search for a path. Each step of moving the probe needs two cycles.

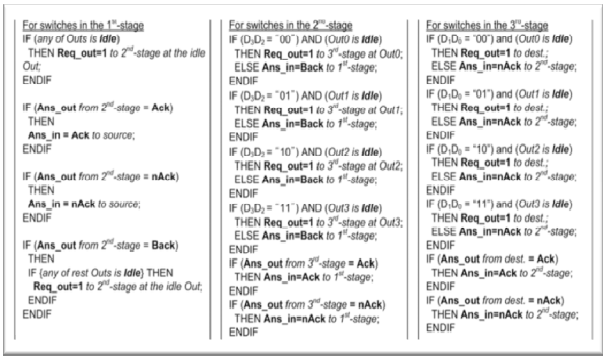

Fig 4 Probe routing algorithms designed to route probe in each stage switches

CONCLUSION

In This paper the presented an on-chip network design supports the traffic permutations in MPSoC applications. It employs the Clos network topology with circuit-switching approach combined with dynamic path-setup scheme.The proposed design supports runtime arbitrary traffic permutation and suggests the use as of the design as an on-chip infrastructure-IP supporting traffic permutation in future MPSoC researches.

REFERENCES

[1]. S. Borkar, "Thousand core chips—A technology perspective," in *Proc.ACM/IEEE Design Autom. Conf. (DAC)*, 2007, pp. 746–749.
[2]. P.-H. Pham, P. Mau, and C. Kim, "A 64-PE folded-torus intra-chip communication fabric for guaranteed throughput in network-on-chipbased applications," in *Proc. IEEE Custom Integr. Circuits Conf.(CICC)*, 2009, pp. 645–648.
[3]. C. Neeb, M. J. Thul, and N.Wehn, "Network-on-chip-centric approach to interleaving in high throughput channel decoders," in *Proc. IEEE Int.Symp. Circuits Syst. (ISCAS)*, 2005, pp. 1766–1769.
[4]. 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.
[5]. H. Moussa, O. Muller, A. Baghdadi, and M. Jezequel, "Butterfly and Benes-based on-chip communication networks for multiprocessorturbo decoding," in *Proc. Design, Autom. Test in Euro. (DATE)*, 2007,pp. 654–65
[6]. 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.
[7]. W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks:*. San Francisco, CA: Morgan Kaufmann, 2004.
[8]. N. Michael, M. Nikolov, A. Tang, G. E. Suh, and C. Batten, "Analysisof application-aware on-chip routing under traffic uncertainty," in *Proc.IEEE/ACM Int. Symp. Netw. Chip (NoCS)*, 2011, pp. 9–16.