# Design and Implementation of an On-chip Multistage Network Topology for System On-Chip

Chatrinaik.R and Sowmya sunkara
MTech Student ,4th Semester, Electronics.
Asst.Professor Dept of ECE ,BMSCE.

***Abstract*— This paper presents the silicon-proven design of a novel on-chip 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. It is a challenging task in a network- on-chip to design an on-chip switch/router to dynamically support (hard) guaranteed throughput under very tight on-chip constraints of power, timing, area, and time-to-market, and coded using Xilinx 12.2.**

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

## I. Introduction

A trend of multiprocessor system-on-chip (MPSoC) design being interconnected with on-chip networks is currently emerging for applications of parallel processing, scientific computing, and so on 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 multiprocessing applications. In addition, many of the MPSoC applications compute in real-time and guaranteeing throughput is critical for such permutation traffics.[1]

The on-chip permutation networks with regard to their implementation shows that most the networks employ a packet-switching mechanism to deal with the conflict of permuted data[2]. Their implementations either use first-input first-output (FIFO) queues for the conflicting data or time-slot allocation in the overall system with the cost of more routing stages or a complex routing with a deflection technique that avoids buffering of the conflicting data .The choices of network design factors, i.e., topology and the routing algorithm, have different impacts on the on-chip implementation[3].

The on-chip permutation network to support guaranteed throughput of permutated traffics under arbitrary permutation. Conventional packet-switching approaches, on-chip network employs a circuit-switching mechanism with a dynamic path-setup scheme under a multistage network topology. The dynamic path setup tackles the challenge of runtime path arrangement for conflict-free permuted data. The pre-configured data paths enable a throughput guarantee. By removing the excessive overhead of queuing buffers, a compact implementation is achieved and stacking multiple networks to support concurrent permutations in runtime is feasible.

The key idea of proposed on -chip network design is based on a pipelined circuit-switch approach with a dynamic path- setup scheme supporting runtime path arrangement. Before mentioning the dynamic path setup scheme ,the network topology is first discussed .Then the design of switching nodes are presented.

### A. *On-Chip Network Topology.*

The Clos network a family of multistage networks, is applied to build scalable commercial multiprocessors with thousands of nodes in macro systems. A typical three-stage Clos network is defined as C (n,m,p) where n represents the number of inputs in each of m first-stage switches and p is the number of second-stage switches. In order to support a parallelism degree of 16 as in most practical MPSoCs .We proposed to use C(4,4,4,) as a topology for the designed network (see fig .1) . The choice of three stage network Clos network with modest number of middle–stage switches is to minimize implementation cost, whereas it still enables a rearrange able property network.

A pipelined circuit-switching scheme is designed for use with the proposed network. This scheme has three phases: the setup, the transfer, and the release A dynamic path-setup



ISSN: 2278 – 1323 

scheme supporting the runtime path arrangement occurs in the setup phase. Restriction of the routing function for deadlock-free data transfer in the virtual circuits with a priority approach may lead to throughput degradation in packet-switched NoCs . Moreover, the implementation of queuing buffers in the packet-switched routers dramatically increases the cost in terms of area required and power consumption.

The circuit-switching approach is favored to provide hard guaranteed throughput due to its attractive QoS property, once a circuit is set up .After this setup, end-to-end data can be pipelined in order at the full rate of the dedicated links with low delay, no data jitter, and in a lossless manner (i.e.,without data dropping) due to there being no collisions among the data streams. Importantly without queuing buffer and complex routing/arbitrating implementation, the circuit switched router results in a low-cost (i.e., area, power) design suitable for the limited on-chip budget. However, a path-setup scheme used in circuit-switched NoCs needs critical considerations for proper functioning (i.e., dead and live-lock free) and low-latency setup with minimization of introduced hardware overhead. Moreover, the dynamic and distributed feature of the path-setup in the circuit- switched NoC is also mandatory to ensure system flexibility and scalability in dynamic management (allocation) of the guaranteed communication circuits. This work advocates the guaranteed throughput implementation with the circuit- switching approach due to compact implementation of routers suitable for on-chip environment and an intrinsic hard QoS property after a circuit has been setup. A novel, practical pipeline circuit-switched switch design is proposed, termed backtracking wave-pipeline switch (or BW switch), to support on-chip hard guaranteed throughput applications.

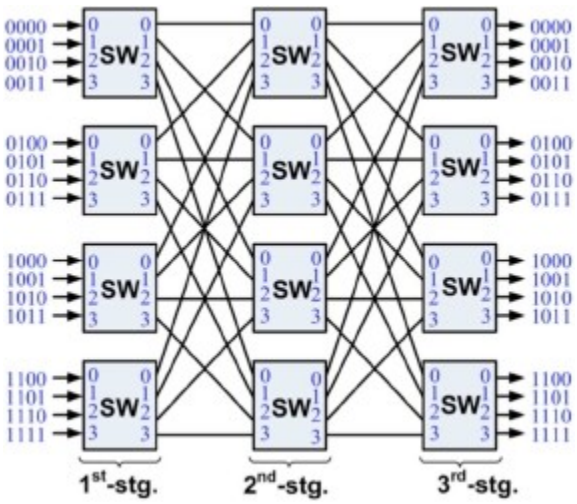



Fig. 1. Proposed on-chip network topology with port addressing scheme.

### *B. Dynamic Path Setup to Support Path Arrangement.*

A dynamic path-setup scheme is the key point of the proposed design to support a runtime path arrangement when the permutation is changed. Each path setup, which starts from an input to find a path leading to its corresponding output, is based on a dynamic probing mechanism. The concept of probing is introduced in works [2] in which a probe (or setup flit) is dynamically sent under a routing al-gorithm in order to establish a path towards the destination. Exhausted profitable backtracking (EPB) [5] is proposed to use to route the probe in the network work. A path arrangement with full permutation consists of sixteen path setups, whereas a path arrangement with partial permutation may consist of a subset of sixteen path setups.

## II. MOTIVATION AND CONTRIBUTION

The relevant design issues of a pipeline circuit- switched switch are considered from both the network-level and the Implementation viewpoints. These lead to the key contributions of this paper. First, as introduced in Section I, a path-setup scheme is critical for circuit-switched NoC. The path-setup scheme (or path configuration) in circuit-switched NoC can be classified as static (at design time or at system boot up time) or dynamic (at runtime) approaches . The static path-setup approach lacks flexibility and scalability, while the dynamic path- setup scheme is flexible and more favored in some circuit-switched NoCs, . The mesh-based SoC BUS presented can dynamically set up a path in distribution, but aces high path-setup latencies, because the occupied channels block the setup of a new path. The work in proposes to use a supplemental packet-switched best-effort(BE) network for the delivery of the path configuration from a central control node (CCN). However, the use of a CCN limits system scalability, and the use of a BE network may result in quite high path-setup latencies, even in the order of , as stated in . At a neutral point, a hybrid packet-/circuit-switching architecture, requires an additional packet-switched network for circuit-setup and back pressure signaling, thus increasing overall system cost and complexity . Moreover, if congestion of the setup headers occurs, the preferred circuit-switching mode reluctantly changes to packet-switching mode . This degrades the QoS property of the circuit-switched data flows. However, no design efforts for dead-/live-lock free delivery of the setup headers are clearly mentioned in previous works . In a pipelined circuit-switching scheme, the data do not immediately follow the header into the network. Hence, it is first observed that it is beneficial if the setup header can flexibly backtrack (under a backtracking routing algorithm) to search for available alternative paths rather than wait (or queue) until the blocking channels become available. Such a backtracking-based path-setup scheme can be implemented in distribution to easily scale the system without the need of additional control network. Second, the implementation of NoCs without global synchronization schemes, e.g.,mesochronous or asynchronous, where inter-clock domain data transfers are needed, becomes common in practice. In such schemes, source-synchronous1 wave-pipeline data


ISSN: 2278 – 1323 

transfers with transceiver designs are efficiently used to combat multiple-clock-cycle multi-Gb/s transmissions in global (inter-router) links. An early study presented in a smaller area and being more energy-efficient than the pipeline approaches using latches or flip-flops. In the design of a pipeline circuit-switched switch (or router), a separate implementation between the data path and the control part is feasible, since, after the path is set up, data can be directly pipelined from source to destination in a control-free manner.
From this point, it can be observed that the design of a pipeline circuit-switched switch, in which its intra-switch data path allows direct-forwarding (i.e., wave-pipelining) of the source-synchronous data from the links, can result in the required latency/throughput property of an on-chip end-to-end path close to that of a dedicated interconnection. The above observations are the motivation for using both backtracking and wave-pipeline techniques for the proposed BW circuit-switched switch, to support on-chip hard guaranteed throughput. The main contributions of this paper are given here.

TABLE I
FORMAT OF SWITCH-BY-SWITCH HANDSHAKE

| | |
|---|---|
| **Request Signal: Req** (1 bit) | |
| 0: Idle (Release) | 1: Circuit Request |
| **Answer signal: Ans** (2 bits) | |
| 00: Idle | 10: Network Blocked |
| 01: Circuit Acknowledge | 11: Busy Destination |

## III. BACKTRACKING WAVE-PIPELINE SWITCH ARCHITECTURE

Here, we propose the backtracking wave-pipeline switch architecture for use under a torus topology. The torus topology is chosen, as the folded torus, a laid-out version of the torus, can fit the tile-based NoC implementation in a conventional 2-D chip, and its good path diversity can be naturally suited to a path-setup scheme with backtracking. The operation of end-to-end communication with the proposed probing path-setup scheme is explained from a network-transaction view point before detailing the proposed BW switch architecture.

### *A. End-to-end flow-control operation with backtracking probing path-setup scheme.*

A typical end-to-end communication, as used in the pipelined circuit-switching approach, is described to provide a clear description of the backtracking probing path-setup scheme used with the BW switch. Communication includes three basic phases: path-setup (or probing), transmission, and release. In the path-setup phase, a probe header containing destination address is sent from the source towards the destination to setup a communication circuit. When congestion occurs, the probe header needs to backtrack and search for alternative links, instead of waiting for a busy link to become idle. That is, the path is set up under a backtracking probing path-setup scheme. When the probe header reaches its destination, an ACK signal is returned to the source. Then, in the transmission phase, the source starts to transmit source-synchronous data through the set up path to the destination. Fig. 2 illustrates the inter-switch and switch-wrapper interconnections with this handshake. Each switch has five bidirectional ports: four ports are connected to corresponding neighboring switches, and the remaining port is connected to the on-chip IP through a wrapper. According to this handshake scheme, one bit is used for the Request (Req) signal to denote the on- probing state (circuit request) and the circuit idling state. Two bits are used for the Answer (Ans) signal. An Ans status of "01" denotes that the receiver is ready to accept data from the sender, whereas a status of "10" denotes that the intended path is blocked in the network, forcing the probe header to backtrack to discover possible alternative paths.

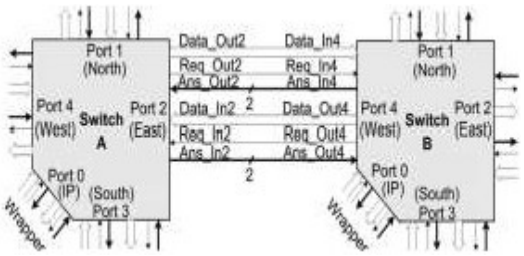



Fig. 2. Switch-by-switch interconnection scheme

An Ans after the last data status of "11" denotes that the receiver is not ready to receive data (e.g., due to being busy, or having an overflow at the receiving buffer). During the setup and the transmission phases, Req is set to "1." When Ans is "00," the probe header continuously advances until it reaches the destination. Then, the destination returns "01" to the source wrapper. When the source wrapper receives "01", it immediately starts to transmit the pipelined data.

### B. *Proposed switch architecture and design.*

As motivated in Section II, the key targets of a proposed BW switch architecture is to support the backtracking probing path setup scheme, and to allow direct-forwarding of source-synchronous data transmissions. As for the backtracking feature of the path-setup scheme, some design issues are first considered. The EPB-based probing path-setup performs a straightforward depth-first search of the network using only profitable links Overall Architecture and Operation**:** The proposed switch architecture has the following main


ISSN: 2278 – 1323 

components that can be divided into two function groups.

i. The data path includes CROSBAR with internal transceivers to support a direct-forwarding (wave-pipelining) of the source-synchronous data.

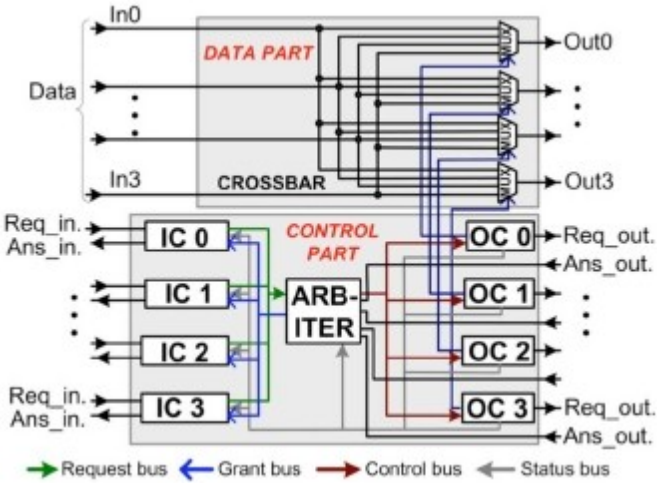



Fig .3 Proposed BW Switch Architecture

ii. The control part includes Ctrl Ins, Ctrl Outs, and ARBITER. Each pair of Ctrl In and Ctrl Out perform handshaking activity at certain bidirectional at port ,namely North ,South, West, and IP.

***A .Function block:*** The Ctrl Ins are in charge of processing the incoming probe headers from upstream switches or from the wrapper (IP). When an incoming probe header arrives at an input (with Req_in "1"), the corresponding Ctrl In monitors the output status through Monitor bus and requests ARBITER to grant it access to the desired Ctrl Out through the internal Request bus. Based on output status or the feedback from ARBITER placed in Grant & Answer bus, the Ctrl In operates appropriately and replies to the upstream switch through its Ans In.

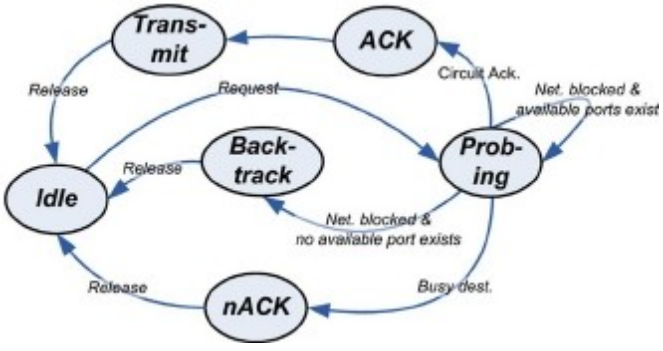



Fig. 4. Simplified state diagram for FSM implementation of Ctrl In to support backtracking probing operation.

The simplified FSM diagram of the operation of Ctrl In with its states to support backtracking probing operation. The Ctrl In is triggered by the rising edge of the probing clock when processing an arriving probe header. Meanwhile, the ARBITER is triggered by the falling edge of the probing clock. The Ctrl Out is simply implemented as a retiming stage for the control signals from the ARBITER and for handshake signals (i.e., Req and Ans) with the downstream switch. This implementation basis in the BW switch ensures that an arriving probe header is always processed in every probe cycle.

**B** ***.Design of Ctrl In:*** The Ctrl In is the key component to perform the backtracking probing task. This includes functions, such as processing history information of backtracking and dynamically constructing a table of possible output ports for probing (i.e., route-probing table). As shown in Fig. 4, when a request with the incoming probe header arrives, Ctrl In goes into the Probing state, compares the current switch address and the destination address (i.e., performs address decoding) to find possible outputs for probing. Based on history information of backtracking, current availability of the output ports, and/or the feedback from the downstream switches, Ctrl In may change into the ACK, nACK or Backtrack states, correspondingly. These operations of Ctrl Ins consistently constitute the back-tracking probing path-setup scheme supported by BW switches throughout each guaranteed throughput lane of the NoC.

## IV. Design of CROSSBAR with Internal Transceivers

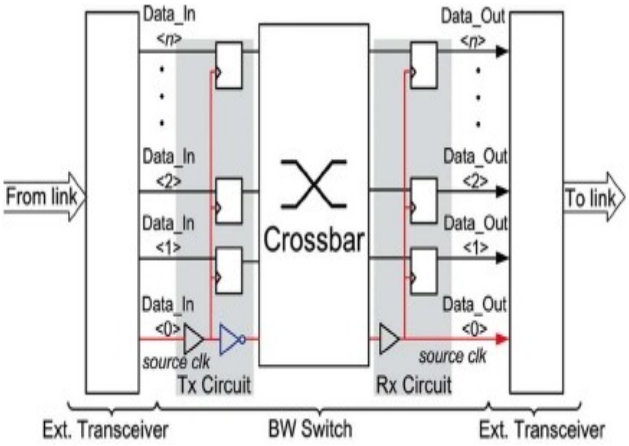



Fig. 5 Internal transceiver (gray boxes) in use with CROSSBAR

The CROSSBAR with internal transceivers is the key component to perform the wave-pipelining of source-synchronous data. Regarding the layered design concept in the NoC paradigm, the router/switch and the transceiver (with inter-router link) can be designed independently. They


ISSN: 2278 – 1323 

can cooperate in NoCs, provided the interface between them is defined. As introduced in Section II, the design of source-synchronous transceivers (with wave-pipelined links) becomes common practice. It is studied to improve energy efficiency and data rate and to combat PVT variations, random mismatch, and crosstalk. In these transceivers, the received data an be realigned with the received (source) clock (as in circuit-switched NoC), or with a local (router) clock (often with synchronizing first-in first-out (FIFO), as in packet-switched NoC.

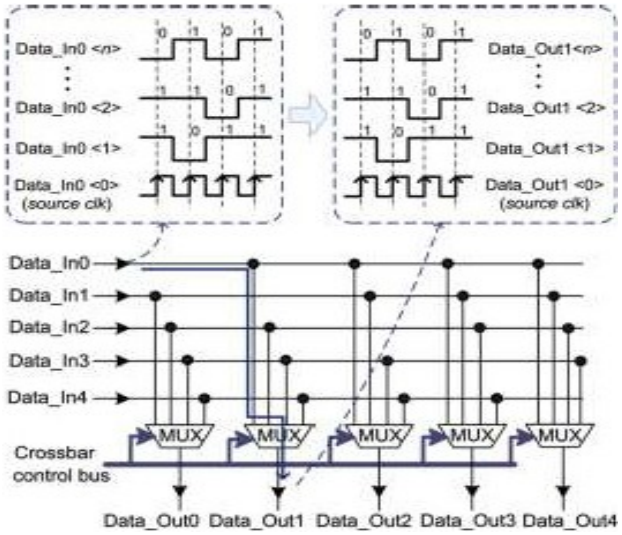
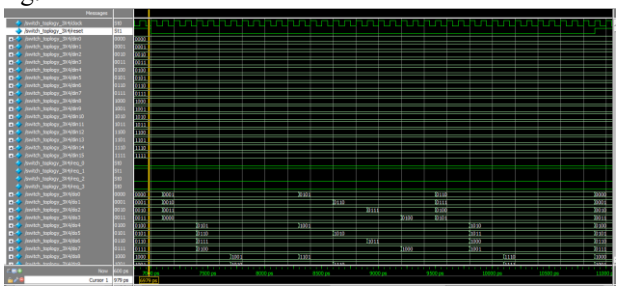



Fig 6. Mux based crossbar structure supporting data wave pipelining

The CROSSBAR is designed as a cross-connection matrix implemented with output multiplexers (Fig. 6), in which each multiplexer is controlled by its corresponding Ctrl Out. In combination with the internal transceivers, this cross-connection structure is capable of direct-forwarding a source-synchronous transmission of a RZ clock along with data pipelined through the switch, as illustrated in Fig. 6 In cooperating with a circuit-switched switch, like the BW switch, the realignment of data to the source clock can be applied.. In the BW switch, the concept of wave-pipeline is illustrated in the sense that it allows direct forwarding source synchronous data (i.e, data along with source clock) form its inputs corresponding outputs. In other words, the crossbar of BW switch can be considered to be set of intra – switch links that requires internal transceivers to works with a source-synchronous direct-forwarding scheme.

## V .RESULTS

After on chip network design configured with a 16-bit data width is synthesized and implemented. Due to the pre-configured circuit-switched data paths, applying a source-synchronous data transfer scheme is feasible. The 32 W*16 bit FIFO is used to log the test data transmitted from the source to the destination, and to support the source-synchronous transfer scheme. Fig. 6.details the source-synchronous transfer scheme, in which one wire of the data path is dedicated for source clock (strobe) transmission and also different type's simulation results of switch 3*4 network topology is as shown fig.



## VI. CONCLUSION

The proposed of an on-chip network design supporting traffic permutations in MPSoC applications. By using a circuit switching approach combined with dynamic path-setup scheme under a Clos network topology. The backtracking feature provides a dynamic and dead- and live-lock free path-setup scheme in a distributed fashion. The wave-pipelining feature practically provides low fall-through latency and high multi-Gbps bandwidth, and suggests suitability for an end-to-end source-synchronous data transmission.

## REFERENCES

[1].C. Neeb, M. J. Thul, and N.Wehn, "Network-on-chip-centric approachto interleaving in high throughput channel decoders," in *Proc. IEEE Int.Symp. Circuits Syst. (ISCAS)*, 2005, pp. 1766–1769.

[2].H. Moussa, O. Muller, A. Baghdadi, and M. Jezequel, "Butterfly andBenes-based on-chip communication networks for multiprocessorturbo decoding," in *Proc. Design, Autom. Test in Euro. (DATE)*, 2007,pp. 654–659.

[3].H. Moussa, A. Baghdadi, and M. Jezequel, "Binary de Bruijn on-chipnetwork for a flexible multiprocessor LDPC decoder," in *Proc. ACM/IEEE Design Autom. Conf. (DAC)*, 2008, pp. 429–434.

[4].P.-H. Pham, J. Park, P. Mau, and C. Kim, "Design and implementation ofbacktrackingwave-pipeline switch to support guaranteed throughputing network-on-chip," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*,10.1109/TVLSI.2010.2096520.

[5].D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta,"Low-power, high-speed transceivers for network-on-chip communication,"*IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 1, pp. 12–21, Jan. 2009.



ISSN: 2278 – 1323 

**Author profile**

**Chatrinaik.R** received the B.E degree Electronics and communication engineering from the SJM institute of technology, Chitradurga fort field, VTU University, Karnataka, India, in 2009. Currently doing **M.Tech.** in Electronics and communication engineering (Electronics) from BMS college of engineering in VTU , Belgaum, India. He research interest includes Low power VLSI, Communication networks.

**Sowmya sunkara** is Assistant Professor department Electronics and communication engineering. Currently Working in BMSCE, Bangalore, from 2005 till date. Her research interest includes ASIC design, National papers in Conferences: 01 and also Projects at Masters level Guiding: 04



ISSN: 2278 – 1323 