Indian J.Sci.Res. 17(2): 167-169, 2018 ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

AGUARANTEED THROUGHPUT IN NETWORK-ON-CHIP PERMUTATION

[1]T. Suganya, [2]Vani Cherukupally, [3]R. Jothi Kumar
[1] Electronics and Communication Engineering MIET Engineering College, Tiruchirappalli.
[2] Electronics and Communication Engineering, Nalla Malla Reddy Engineering College, Hyderabad.
[3] Department of Computer Science and Engineering, Kumaran Institute of Technology, Minjur, Chennai

Abstract - This project presents 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 circuit-switching approach offers a guarantee of permuted data and its compact overhead enables the benefit of stacking multiple networks. In our system both fault mitigation and data priority can be achieved.This paper also proposes a fault-tolerant solution for a bufferless network-on-chip, including anfault-diagnosis mechanism to detect both transient and permanent faults and priority approach for data transmission.which empowers the router to choose the most suitable packet forwarding path, based on the priority of the router and the current energy status of the forwarding router A xilinx 9.1 validates the feasibility and efficiency of the proposed design.

Keywords –Fault diagnosis, Single-chip systems, Fault-tolerance.

I. Introduction

The design of a chip is based on four distinct aspects: computation, memory, communication and I/O. The increase of the processing power and the emergence of data intensive applications has attracted major attention on the challenge of the communication aspect in single-chip systems (SoC).

The possibilities of providing throughput guarantees in a network-on-chip by appropriate traffic routing. A source routing function is used to find routes with specified throughput for the data streams in a streaming multiprocessor system-on-chip. The influence of the routing algorithm, network topology and communication locality on the routing performance are studied.The results show that our method for providing throughput guarantees to streaming traffic is feasible. The communication locality has the strongest influence on the routing performance while the routing algorithm has weakest influence. Therefore, the mapping algorithm is of greater importance for the system performance than the routing algorithm and it is profitable to use a more complex mapping algorithm that preserves the communication locality together with a simple routing algorithm

This paper presents a novel design of an on-chip permutationnetwork to support guaranteed throughput of permutated traffics under arbitrary permutation. Unlike conventional packet-switchingapproaches, our on-chip network employs a circuit-switching mechanismwith a dynamic path-setup scheme under a multistage networktopology. The dynamic path setup tackles the challenge of runtime patharrangement for conflict-free permuted data. The pre-configured datapaths enable a throughput guarantee. By removing the excessive overheadof queuing buffers, a compact implementation is achieved andstacking multiple networks to support concurrent permutations in runtimeis feasible

II. Existing On-Chip Network Design

On-chip network design is based on a pipelined circuit-switching approach with a dynamicpath-setup scheme supporting runtime path arrangement

A dynamic path-setup scheme is the key point of the proposed designto support a runtime path arrangement when the permutation ischanged. Each path setup, which starts from an input to find a pathleading to its corresponding output, is based on a dynamic probing mechanism. The concept of probing is introduced in works in which a probe (or setup flit) is dynamically sent under a routing algorithmin order to establish a path towards the destination. Exhaustedprofitable backtracking (EPB) is proposed to use to route the probe in the network work. A path arrangement with *full permutation* consistsof sixteen path setups, whereas a path arrangement with *partial permutation*may consist of a subset of sixteen path setups.As designed in this network, each input sends a probe containinga 4-bit output address to find an available path leading to the targetoutput. During the search, the probe moves forwards when it finds afree link and moves backwards when it faces a blocked link. By meansof non-repetitive movement, the probe finds an available path betweenthe input and its corresponding idle output. The EBP-based path-setupscheme is designed with a set of probe routing algorithms The following example describes how the path setupworks to find an available path by using the set of path diversity. It is assumed that a probe from a source (e.g., an input of switch 01) is trying to set up a

[1]Corresponding Author

path to a target destination (e.g., anavailable output of switch 22. First, the probe will non-repetitively trypaths 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

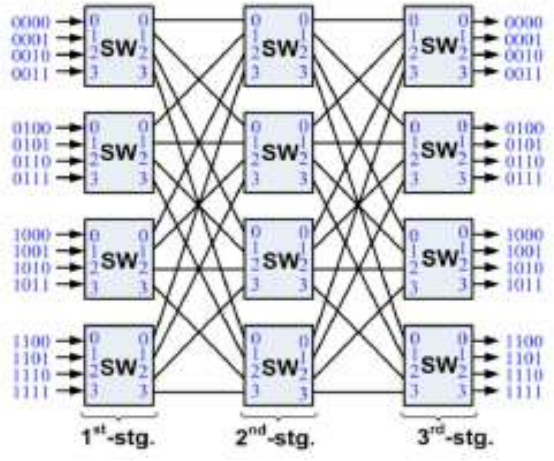

Fig 2 : Existing Soc Proposed On-Chip Network Design

In our system both fault mitigation and data priority can be achieved to develop fault tolerant on chip network

Priority network which empowers the router to choose the most suitable packet forwarding path, based on the priority of the router and the current energy status of the forwarding router.

Fault detecting Our mitigation technique identify the faults in a router while detect any fault in the path the alternate router will be selected for data transmission to ensure the guaranteed data Fault-tolerance or graceful degradation is the property that enables a system (often computer-based) to continue operating properly in the event of the failure of (or one or more faults within) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naïvely-designed system in which even a small failure can cause total breakdown. Fault-tolerance is particularly sought-after in high-availability or life-critical systems.

Recovery from errors in fault-tolerant systems can be characterised as either roll-forward or roll-back. When the system detects that it has made an error, roll-forward recovery takes the system state at that time and corrects it, to be able to move forward.

Within the scope of an *individual* system, fault-tolerance can be achieved by anticipating exceptional conditions and building the system to cope with them, and, in general, aiming for self-stabilization so that the system converges towards an error-free state. However, if the consequences of a system failure are catastrophic, or the cost of making it sufficiently reliable is very high, a better solution may be to use some form of duplication. In any case, if the consequence of a system failure is so catastrophic, the system must be able to use reversion to fall back to a safe mode. This is similar to roll-back recovery but can be a human action if humans are present in the loop.In addition, fault tolerant systems are characterized in terms of both planned service outages and unplanned service outages. These are usually measured at the application level and not just at a hardware level.

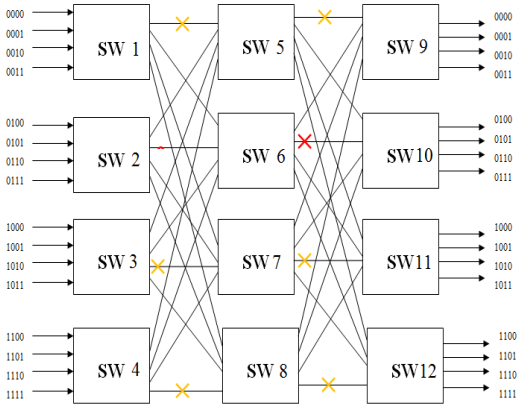

Fig 3 Proposed Soc

In above NOC ,switch 6 are affected by fault. It change the routing path automatically to switch 7 when try to access switch 7 by fault detecting mechanism.

III. Experimental Results

The proposed SOC system, with fault detecting mechanism are simulated by using Xilinx ISE 12.1i and implemented in sparten FPGA processor.

IV. Performance Analysis

The performance of the system analyzed in terms number slices and LUTs obtained by performing synthesis process in Xilinx tool.

s.no	parameter	used
1	Number of Slices	44
2	Number of 4 input LUTs	77

V. RTL Schematic

After performing the synthesize process, the RTL schematic has been created automatically based on the functionality. The routing between the different cells can be viewed clearly by this schematic

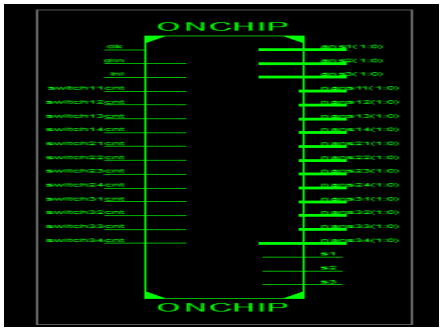

Fig 4 Rtl

VI. Conclusion

This paper has presented an on-chip network design supportingtraffic permutations in MPSoC applications. By using acircuit-switching approach combined with fault detecting scheme under a Clos network topology, the proposed design offersarbitrary traffic permutation in runtime with compact implementationoverhead.

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-chipcommunication 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 approachto 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-chipnetwork 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 andBenes-based on-chip communication networks for multiprocessorturbo decoding," in Proc. Design, Autom. Test in Euro. (DATE), 2007, pp. 654–659.

[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 InterconnectionNetworks:. 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.