ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Area Efficient system on chip processor for routing operation

Vankudoth Prabhakar
MTech Student
Department of ECE
AnuBose Institute Of Technology(ABIT)
Paloncha, Khammam, India

T.Venu Gopal
Professor
Department of ECE
AnuBose Institute Of Technology(ABIT)
Paloncha, Khammam, India

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. A 0.13-μ m CMOS test-chip validates the feasibility and efficiency of the proposed design.***

Index Terms—Neighbor position verification, mobile ad hoc networks, vehicular networks

INTRODUCTION:

A trend of multiprocessor system-on-chip (MPSoC) designbeing 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. Many of the MPSoC applications (e.g., Turbo/LDPC decoding) compute in realtime, therefore, guaranteeing throughput (i.e., data lossless, predictable latency,guaranteed bandwidth, and in-order delivery) is critical for such permutation traffics1-7. 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. Such application-aware routings cannot efficiently handle the dynamic changes of a permutation pattern, which isexhibited in many of the application phases. The difficulty lies in the design effort to compute the routing to support the permutation changes in runtime, as well as to guarantee the permutated traffics6-11. Most on-chip networks employ a packet-switching mechanism to deal with the conflict of permuted data. Their implementations use first-input first-output (FIFO) queues for the conflicting data.

Existing System:

Regarding the switching technique, packet switching requires an excessive amount of onchippower and area for the queuing buffers (FIFOs) with pre-computed queuing depth at the switching nodes and/or network interfaces. Regarding the routing algorithm, the deflection routing is not energy-efficient due to the extra hops needed for deflected data transfer, compared to a minimal routing. Moreover, the deflection makes packet latency less predictable; hence, it is hard to guarantee the latency and the in-order delivery of data. This paper presents a novel silicon-proven design of an on-chip permutation network to support guaranteed throughput of permutated traffics under arbitrary permutation. Unlike conventional packet-switching approaches, our on-chip network employs a circuitswitchingmechanism 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

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

networks to support concurrent permutations in runtime is feasible.

Proposed System:

As motivated in Section I, the key idea of proposed on-chipnetwork design is based on a pipelined circuit-switching approach with a dynamic path-setup scheme supporting runtime path arrangement. The network topology is first discussed. Then the designs of switching nodes are presented. On-Chip Network TopologyClose network, a family of multistage networks, is applied to build scalable commercial multiprocessors with thousands of nodes in macrosystems. A typical three-stage Clos network is defined as C (n, m , p), where n represents the number of inputs in each of p first-stage switches and m is the number of second-stage switches. In order to support a parallelism degree of 16 as in most practical MPSoCs ,here proposed to use C (4,4, 4) as a topology for the designed network(See Fig. 1). This network has a rearrangeable property that can realize all possible permutations 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 rearrangeable property for the network24-28. 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.

PROPOSED ON-CHIP NETWORK DESIGN

To meet the growing computation-intensive applications and the needs of low-power, high performance systems, the number of computing resources in single-chip has enormously increased, because current VLSI technology can support such an extensive integration of transistors. By adding many computing resources such as CPU, DSP, specific IPs, etc to build a system in System-on-Chip, its interconnection between each other becomes another challenging issue. In most System-on-Chip applications, a shared bus interconnection which needs arbitration logic to serialize several bus access requests, is adopted to communicate with each integrated processing unit because of its low-cost and simple control characteristics. However, such shared bus interconnection has some limitation in its scalability because only one master at a time can utilize the bus which means all the bus accesses should be serialized by the arbitrator. Therefore, in such an environment where the number of bus requesters is large and their required bandwidth for interconnection is more than the current bus, some other interconnection methods should be considered. This network has a rearrange able property that can realize all possible permutations between its input and outputs.

Figure 1: Proposed on-chip Network Topology with Port Addressing Scheme

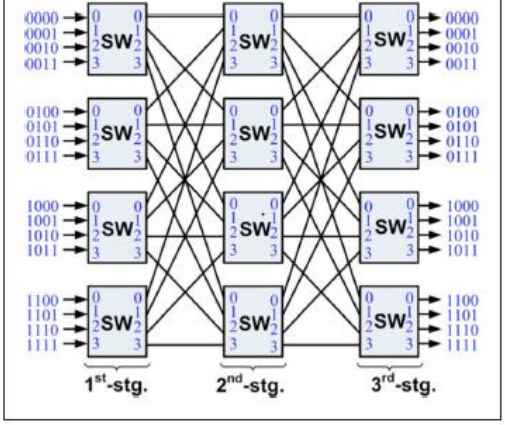
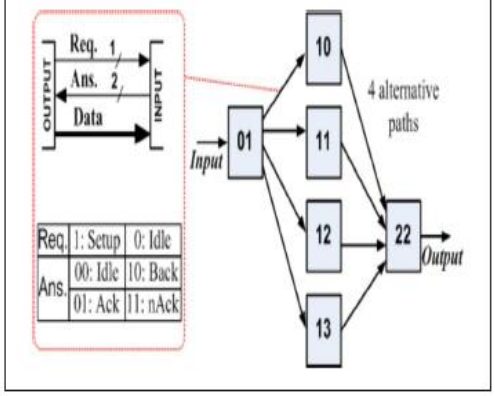


Figure 2: Switch-by-Switch Interconnection and Path-Diversity Capacity



ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Synchronization

Synchronization refers to one of two distinct but related concepts: synchronization of processes, and synchronization of data. Process synchronization refers to the idea that multiple processes are to join up or handshake at a certain point, in order to reach an agreement or commit to a certain sequence of action. Data synchronization refers to the idea of keeping multiple copies of a dataset in coherence with one another, or to maintain data integrity. Process synchronization primitives are commonly used to implement data synchronization.

The latency of synchronizer chains has a detrimental impacton the performance of latency-sensitive applications and so alternative methods to reduce or eliminate this latency have been proposed. These methods can be broadly divided into three distinct categories as follows.

TECHNOLOGIES

Currently there are four technologies in use. They are: static RAM cells, anti-fuse, EPROM transistors, and EEPROM transistors. Depending upon the applications, one FPGA technology may have features desirable for that application.

1. Static RAM Technology

In the Static RAM EPGA programmable connections are made using pass-transmission, transmission gates, or multiplexers that are controlled by SRAM cells. This technology allows fast in-circuit reconfiguration. The major disadvantage is the size of the chip required by the RAM technology and that the chip configuration needs to be loaded to the chip from some external source (usually external non-volatile memory chip). The FPGA can either actively read its configuration data out of external serial or byte-parallel PROM (master mode), or the configuration data can be written into the FPGA (slave and peripheral mode). The FPGA can be programmed an unlimited number of times.

2. Anti-Fuse Technology

An anti-fuse resides in a high-impedance state; and can be programmed in to low impedance or "fused" state. This technology can be used to make program once devices that are less expensive than the RAM technology.

3. EPROM Technology

This method is the same as used in the EPROM memories. The programming is stored without external storage of configuration. EPROM based programmable chip cannot be re-programmed in circuit and need to be cleared with UV erasing.

4. EEPROM Technology

This method is the same as used in the EEPROM memories. The programming is stored without external storage of configuration. EEPROM based programmable chips can be electrically erased but generally cannot be re-programmed in-circuit FUSE-One-time programmable.

Conclusion:

This paper proposes an on-chip multistage interconnection network with the least possible number of hardware, the minimum amount of wiring between stages and the minimum wire lengths. It can be used for high-performance interprocessor communication in real-time applications. Although logN -MINs have been already researched and used in parallel super computers, they can be adapted also for networkon- chips as well. High bandwidth and low latency are combined with a deterministic behavior of interprocessor communication in the proposed NoC. The objective is to use MPSoCs in highperformanceembedded systems with hard realtimeconstraints that can be found in electronic control units for cars or for production machinery.

References:

[1] J. Savir and S. Patil, "Scan-based transition test," *IEEE Trans. Comput.- Aided Design*, vol. 12, no. 8, pp. 1232–1241, Aug. 1993.

[2] J. Savir and S. Patil, "Broad-side delay test," *IEEE Trans. Comput.-Aided Design*, vol. 13, no. 8, pp. 1057–1064, Aug. 1994.

[3] S. Wang, X. Liu, and S. T. Chakradhar, "Hybrid delay scan: A low hardware overhead scan based delay test technique for high fault coverage and compact test sets," in *Proc. Design Autom. Test Eur. Conf.*, 2004, pp. 1296–1301.

[4] I. Pomeranz and S. M. Reddy, "Effectiveness of scan-based delay fault tests in diagnosis of transition faults," *IET Comput. Digital Tech.*, vol. 1, no. 5, pp. 537–545, Sep. 2007
.
[5] I. Park and E. J. McCluskey, "Launch-on-shift-capture transition tests," in *Proc. IEEE Int. Test Conf.*, Santa Clara, CA, Oct. 2008, pp. 1–9.

[6] I. Pomeranz and S. M. Reddy, "On-chip generation of the second primary input vectors of broadside tests," in *Proc. Int. Symp. Defect Fault Tolerance VLSI Syst.*, 2009, pp. 38–46.

[7] I. Pomeranz, "Multi-pattern scan-based test sets with small numbers of primary input sequences," *IEEE Trans. Comput.-Aided Design*, vol. 31, no. 2, pp. 322–326, Feb. 2012.

[8] Q. Xu and N. Nicolici, "DFT infrastructure for broadside two-pattern test of core-based SOCs," *IEEE Trans. Comput.*, vol. 55, no. 4, pp. 470–485, Apr. 2006
.
[9] L. N. Reddy, I. Pomeranz, and S. M. Reddy, "COMPACTEST-II: A method to generate compact two-pattern test sets for combinational logic circuits," in *Proc. IEEE Int. Conf. Comput.-Aided Design*, Santa Clara, CA, Nov. 1992, pp. 568–574.

[10] I. Hamzaoglu and J. H. Patel, "Compact two-pattern test set generation for combinational and full scan circuits," in *Proc. Int. Test Conf.*, Washington DC, Oct. 1998, pp. 944–953.

[11] N. Tendolkar, R. Raina, R. Woltenberg, X. Lin, B. Swanson, and G. Aldrich, "Novel techniques for achieving high at-speed transition fault test coverage for Motorola's microprocessors based on owerPC(TM) instruction set architecture," in *Proc. VLSI Test Symp.*, 2002, pp. 3–8.

[12] Y. Shao, I. Pomeranz, and S. M. Reddy, "On generating high quality tests for transition faults," in *Proc. Asian Test Symp.*, Nov. 2002, pp. 1–8.

[13] W. Qiu, J. Wang, D. M. H. Walker, D. Reddy, X. Lu, Z. Li, W. Shi, and H. Balachandran, "K longest paths per gate (KLPG) test generation for scan-based sequential circuits," in *Proc. Int. Test Conf.*, 2004, pp. 223–231
.
[14] Z. Chen, D. Xiang, and B. Yin, "The ATPG conflict-driven scheme for high transition fault coverage and low test cost," in *Proc. VLSI Test Symp.*, Santa Cruz, CA, 2009, pp. 146–151.

[15] I. Pomeranz and S. M. Reddy, "On test generation with test vector improvement," *IEEE Trans. Comput.-Aided Design*, vol. 29, no. 3, pp. 502–506, Mar. 2010.

[16] I. Pomeranz, "Static test compaction for delay fault test sets consisting of broadside and skewed-load tests," in *Proc. VLSI Test Symp.*, 2011,