

Crossbar Switch For Network Processor application

Prashant wanjari^{#1} and Vijendra Meshram^{#2}

^{#1}Dept. of Electronics Eng., Nagpur university, India, prashantwanjari411@gmail.com

^{#2}Dept. of Electronics Eng., Nagpur university, India,, vijendrameshram@gmail.com

Abstract— A Network Processor is an Integrated Circuit, with a feature set designed to tackle the needs of the networking application domain. Typically NPs are software programmable and have generic functions, which are similar to general purpose Central Processing Units, commonly used in many different products, Due to the fact that in modem telecommunication systems information is transferred in a packet form instead of the analog signals used in older systems, a need has arisen to develop ICs optimized to handle such packet forms of data. These ICs are called NPs and they make use of specific features or architectures to enhance and optimize packet processing in computer networks [2]. NPs play a major role in network applications since they are tasked with handling most of the routing, forwarding and security related functions required by network traffic in modem computer networks of all sizes. This paper is to present the Reconfigurable crossbar switch, a reconfigurable crossbar switch architecture used to connect different inputs and outputs in interconnection and communication networks.

Keywords-Network processor,FPGA,VHDL

I. INTRODUCTION

Network processors appeared in the late 1990s, and flourished as major processor and network vendors led the NPs market (companies like Intel, IBM, and Motorola). Since the concept was new, it created a lot of enthusiasm, and caused a wave of established companies as well as many newcomers to invest and innovate. Due to the fact that in modem telecommunication systems information is transferred in a packet form instead of the analog signals used in older systems, a need has arisen to develop ICs optimized to handle such packet forms of data. These ICs are called NPs and they make use of specific features or architectures to enhance and optimize packet processing in computer networks. By evolving over time, NPs have grown to become more flexible but at the same time more complex ICs [1]. In newer iterations, NPs are programmable, thus providing the advantage of handling many different functions using the same hardware, by only installing the appropriate software. NPs play a major role in network applications since they are tasked with handling most of the routing, forwarding and security related functions required by network traffic in modem computer networks of all sizes. As of the beginning of 2008, major network vendors announced new generations of NPs that are the only way for them to compete and introduce network devices that can sustain network demands. Total investment in NPs development has reached approximately 1 billion U.S. dollars, as of 2008. Companies that introduced various levels of NPs include: there are many terms in use in relation to packet and traffic flows and they are sometimes contradictory. There are also multiple terms for the same thing, or the same term is used to define different things, both specific and broad. In the following, in the interest of clarity, I adopt terminology that I feel is appropriate, although some might argue for the use of alternative terminology.

Corresponding Author: Prashant wanjari,
prashantwanjari411@gmail.com

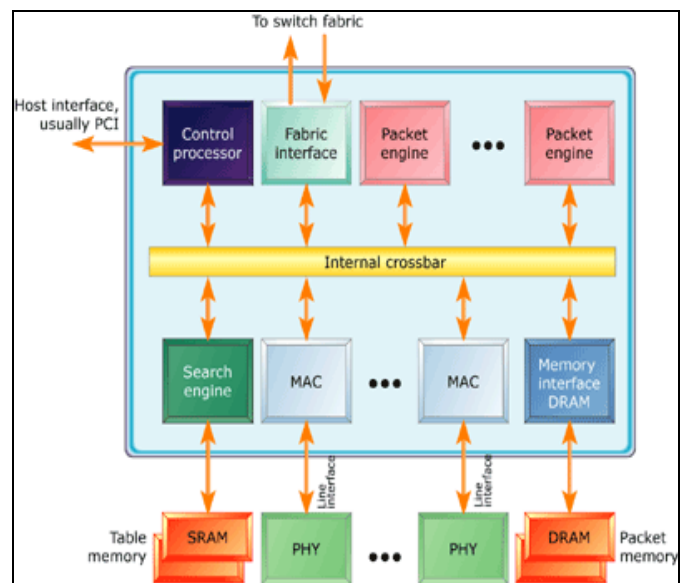


Figure1. Network processor

A typical Network Processor has four basic external interfaces, as shown in figure 1 the first is the line interface, which often connects to external MAC or framer chips. Some NPUs include on-chip MACs or framers (or both), in which case the line interface may connect directly to external PHY (physical-layer) devices. Network processors took the place of some GPPs (General-Purpose Processors) and ASICs (Application Specific Integrated Circuits) in network equipments, targeting two important issues: flexibility and performance. As these features are essential to process the packets, a network processor is the best choice to get them. In a network, a cross-bar switch is a device that is capable of channeling data between any two devices that are attached to it up to its maximum number of ports. A major advantage of

cross-bar switching is that, as the traffic between any two devices increases, it does not affect traffic between other devices [4]. In addition to offering more flexibility, a cross-bar switch environment offers greater scalability than a bus environment.

Modern routers use programmable packet processors on each port to implement packet forwarding and other advanced protocol functionality. This programmability in the data path is an important aspect of router designs in the current Internet in contrast to the traditional approach where custom application-specific integrated circuits with fixed functionality are used. The ability to change a router's operation by simply changing the software processed on router ports makes it possible to introduce new functions (e.g., monitoring, accounting, anomaly detection, blocking, etc.) Without changing router hardware. An essential requirement for these systems is the availability of a high-performance packet processor that can deliver packet processing at data rates of multiple Gigabits per second.

II. GENERAL ARCHITECTURE OF RECONFIGURABLE NETWORK PROCESSOR

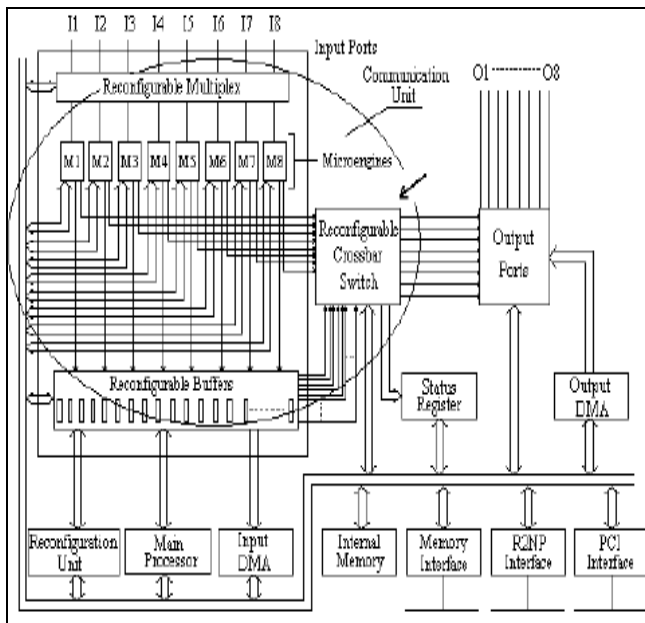


Figure 2 Reconfigurable RISC Network Processor

The figure 2 presents the R2NP (Reconfigurable RISC Network Processor) architecture. The R2NP has been used as a base for the design of the reconfigurable crossbar switch architecture. Thus, the design of RCS (Reconfigurable Crossbar Switch) was based on the use of it in a network processor.

Reconfigurable crossbar switch presented in figure 3, has three main blocks: (1) connection matrix, where the topologies are implemented; (2) decoder, that converts the reconfigurable bits for a matrix bits set and (3) pre-header analyzer (PHA). NPs can add a pre-header in the packet with the output destination. RCS architecture is based on two reconfiguration levels. Using these two levels it is possible to reconfigure and to readapt the crossbar switch to many network topologies and different workload situations.

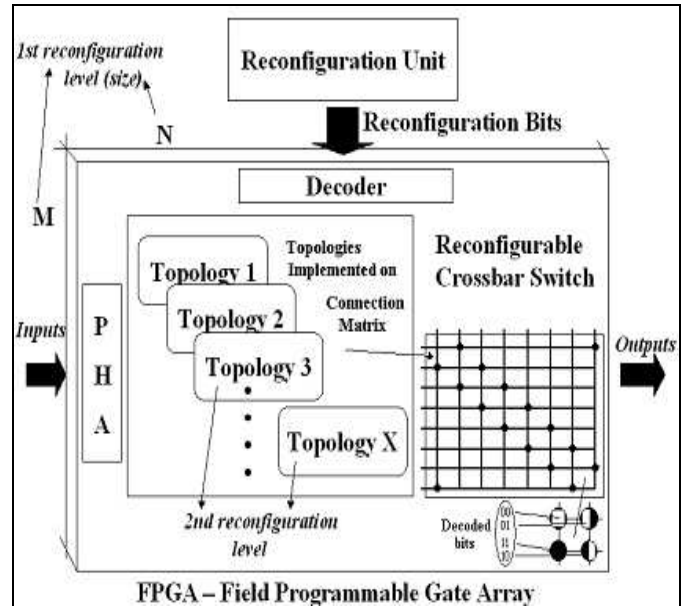


Figure 3 Reconfigurable crossbar switch Architecture

Packet processors are RISC-based processors, with the advantage of being small, fast, inexpensive, easy to integrate with other hardware, and easy to program. PPs perform data-plane tasks and provide fast-path data processing at wire speed. Most packets are processed by PPs. PPs use an instruction set that is optimized for packet processing. Memory I/O latencies affect performance a great deal. To hide memory latencies most PPs employ multi-threading technology on hardware to process multiple packets on a single PP concurrently. It minimizes the overhead of context switching, thus significantly increasing the overall throughput. Data-plane tasks include packet classification, forwarding, filtering, header manipulating, protocol conversion and policing. Most processing in network applications occurs in data planes.

Control processor:

The control processor is a general-purpose processor that runs an embedded operating system. The control processor provides overall control, performs configuration management, and processes exception packets. Exception packets could be control-plane-related, or data-plane-related that may require extra processing such as IP packets with options.

Coprocessors:

The coprocessors are special-purpose hardware, providing specific functions for carrying out common network tasks, including pattern matching, table lookup, buffer management, queue management, hashing, checksum computation, and encryption/decryption. Since these functions are commonly used in packet processing regardless which protocols are used, implementing them via hardware speeds up execution. Coprocessors can be used to simplify software creation, for they provide a single-instruction access to complex operations. Network and fabric interfaces The fabric interfaces handle interaction between processors and fabric switch, and network interfaces handle interaction between processors and the physical layer of the external network. Most network processors also include data transfer units that are responsible for moving packets between NIAC devices and memory directly.

Memory:

High speed memory is expensive. Regular computer systems often use different types of memories in a hierarchical manner to balance between cost and speed. For example, an on-chip level 1 cache has the fastest speed, but with the smallest capacity (i.e., the number of bytes it can store). Level 2 and level 3 caches each provide lower speed with larger capacity than the previous level. The main memory has the largest capacity but with the lowest speed. To achieve good performance, data that are more frequently accessed are stored in faster memories. NPs adopt a similar memory hierarchy. Since NPs are used to process a large volume of network packet data that demonstrates almost no locality, most NPs do not provide cache to packet processors. Some NPs provide on-chip memory for fast accessing. All NPs provide high-speed memory interface for various levels of external memory, where the Static RAM (SRAM) provides faster speed and the Dynamic RAM (DRAM) provides large storage with lower accessing speed. Unlike conventional computer systems, NP programmers need to explicitly choose memory to store data. SRAM is used to store configuration and status information, or packet headers in some cases, which needs to be accessed frequently. DRAM provides large space to buffer the payload data that are less frequently accessed. A number of chip makers manufacture various types of Network Processor. The most popular models include AMCC nPcore family & Intel family [3].

III. NETWORK PROCESSOR ARCHITECTURE

Network processors play a crucial role in packet processing such as packet forwarding in the network equipment. In the network processors of the first generation, general purpose processors were used. Network applications were mostly software based and new features could be easily added. However, scalability was severely limited and some of these processors even failed to meet the speed requirements. As a result, ASIC-based network processors were introduced as second generation processors. ASIC-based network processors are typically used to forward traffic at very high rates. Their disadvantages include high development costs, long time to market, and little flexibility.

As internet traffic continues increasing rapidly and the protocols are becoming more dynamic and sophisticated, network processors are required to be very flexible as well as fast. Thus, network processors that have an instruction set specialized for network applications, flexibility, and support speed of line rates have emerged as the third generation network processors. The origins of the third generation network processors are the Intel's IXP, the IBM's PowerNP, and the Motorola's C-Port. These network processors are actually architecturally quite similar. They are composed of one or more processing elements and of a couple of co-processors for common network applications. The IXP2800 delivers processing capability at OC-192/10 Gbps line rates. As in the IXP2800 example, most programmable NPs on the market today target low performance (from 100 Mb/s to 10 Gb/s), low cost edge routers, leaving the task of routing in the backbone to ASICs [6].

IV. RELATED WORK

There are lots of commercial network processors of different companies. Some companies and respective network processors are: IBM (NP4GS3), Motorola/CPort (C-5 Family),

Lucent/Agere (FPP/RSP/ASI), and Sitera/Vitesse (IQ2000), Chameleon (CS2000), EZChip (NP-1), Intel (IXP1200) and others. None of them presents reconfigurability, except the CS2000 of Chameleon. However, it does not have reconfigurable crossbar switch. NP architectures have dedicated blocks to execute specific functions as an embedded ASIC (NPSoC – Network Processor System-on-Chip). Some blocks are: PCI units, memory units, packet classifiers, policy engines, metering engines, and packet transform engines, pattern processing engine, queue engine, QoS engines and other blocks. There are some documents and papers about crossbar switch, but nothing using reconfigurable crossbar in a network processor.

Programmability in the data path of routers has been introduced as software extensions to workstation-based routers (e.g., Click modular router, dynamically extensible router) as well as multi-core embedded network processors (e.g., Intel IXP platform, Cisco Quantum Flow processor, EZchip NP-3, and AMCC nP series). Programmability in the data path can be used to implement additional packet processing functions beyond simple IPv4-forwarding or in network data path service for next-generation networks[5].

V. SIMPLE 4X4 CROSSBAR SWITCH ARCHITECTURE

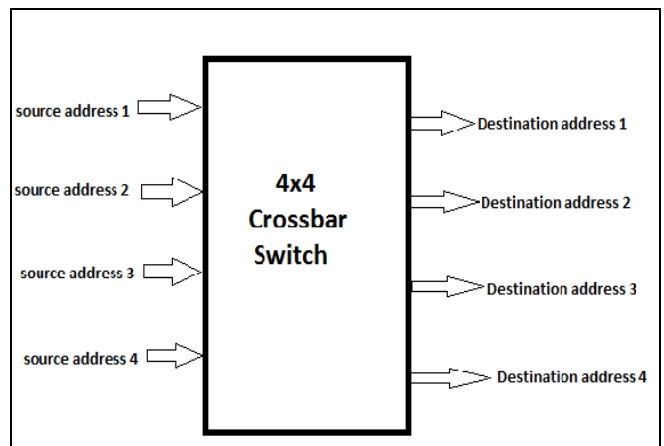


Figure: 4 Simple crossbar Switch

Reconfigurable crossbar switch (RCS-2) uses reconfiguration bits to implement the topology in the space. That topology actually maintains the created connections as a circuit. The reconfiguration bits set are capable of reconfiguring or implement a new topology in RCS-2 whenever necessary. RCS-2 architecture is based on two reconfiguration levels. Using these two levels it is possible to reconfigure and to readapt the crossbar switch to many network topologies and different workload situations. The first level is based on static reconfiguration using a reconfigurable device, like FPGA. Programming this device, it is possible to implement a RCS-2 with number of in and out ports (and consequently rows and columns – circuit and logic gates) limited by the device capacity. The second level of reconfiguration makes possible the implementation of different network topologies. It could be done by dynamically reconfiguration of the connection matrix nodes. These nodes determine which connections will be closed and consequently which paths exist through the

crossbar switch. RCS-2 has two bits of reconfiguration to each node, which define the current topology. Only the Reconfiguration Unit and the instruction set of the network processor are able to change those bits in order to implement new topologies. Although one instruction can modify a reconfigurable bit, it only modifies the 01 and 10 formats the 00 and 11 formats are restricted to Reconfiguration Unit [7]. The reconfigurable crossbar switch has some connection nodes, which, if closed, compose a circuit. This circuit represents a topology in space. Differently from a traditional crossbar switch (TCS), where it is possible to close only one node per line or column, regards the implemented topology, the RCS-2 permits that more than one node can be closed per line or column at the same time. In a TCS, the topologies cannot be implemented in the space, only in the time.

VI. ADVANTAGES & DIS-ADVANTAGES

The main competitors to NPUs are general-purpose microprocessors and custom ASICs. In previous networking systems, microprocessors were used to perform routing functions in low-end devices because of their low cost, general availability, and ease of programming. Microprocessors don't have enough performance for high-bandwidth devices, so these boxes used custom ASICs. ASICs provide ultimate control over the design. Using ASICs, a networking designer can create highly differentiated products. On the other hand, ASICs have long design cycles(9 -18 months), long debug cycles, and high development costs (millions of dollars). As a result, ASIC development is the riskiest portion of system development. Like standard microprocessors, network processors are programmable and available off the shelf, yet they can match the performance of ASICs in demanding networking applications. NPUs replace fixed-function ASICs with a programmable design, providing additional advantages. A programmable device shortens the design cycle and is more easily modified to support new or evolving standards. Programmability not only accelerates time to market, it can even enable an NPU-based router to be field-upgraded with a new protocol something that can't be done with a hardwired solution [8].

VII. COMMON CHARACTERISTICS

A single network data stream contains a large number of individual packets, each of which can be processed fairly independently. In fact, Internet protocol (IP) allows individual packets within a single data stream to be processed in any order again Because of this independence, packet processing is an ideal application for an array of processors. By dividing up the task, one chip can deliver high performance using several processing units of modest speed. These units needn't squeeze out the last bit of performance, using techniques such as superscalar issue or instruction reordering, which require a great many transistors and a corresponding increase in power consumption. Packet processors can thus be small and efficient.

VIII. RESULT

A) Object window for packet data 1

Name	Value	Kind	Mode
packet1	0000000111111111	Signal	In
packet2	UUUUUUUUUUUUUUUU	Signal	In
packet3	UUUUUUUUUUUUUUUU	Signal	In
packet4	UUUUUUUUUUUUUUUU	Signal	In
clk	1	Signal	In
output1	ZZZZZZZZ	Signal	Out
output2	11111111	Signal	Out
output3	ZZZZZZZZ	Signal	Out
output4	ZZZZZZZZ	Signal	Out
input1	11111111	Signal	Internal
input2	UUUUUUUU	Signal	Internal
input3	UUUUUUUU	Signal	Internal
input4	UUUUUUUU	Signal	Internal
source_address1	00	Signal	Internal
dest_address1	01	Signal	Internal
source_address2	UU	Signal	Internal
dest_address2	UU	Signal	Internal
source_address3	UU	Signal	Internal
dest_address3	UU	Signal	Internal
source_address4	UU	Signal	Internal
dest_address4	UU	Signal	Internal

Fig 5.1 object window for packet data 1

B) Wave simulation window for packet data 1

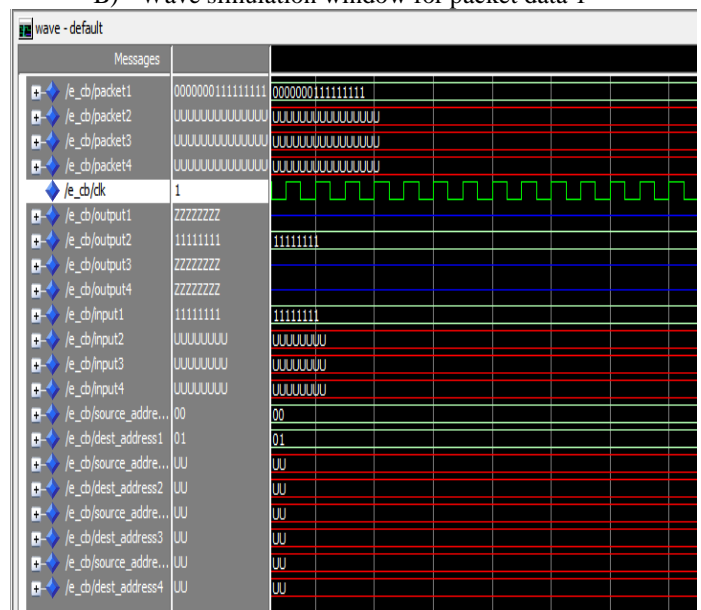


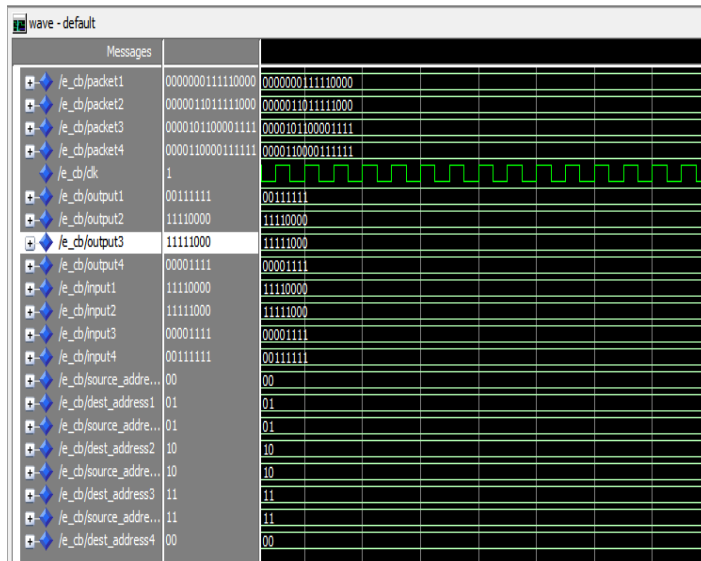
Fig 5.2 wave window for packet data 1

C) Object Window for all Packet data

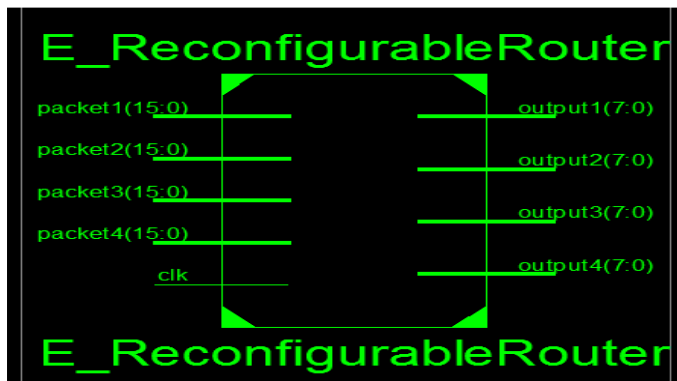
Name	Value	Kind	Mode
packet1	0000000111110000	Signal	In
packet2	0000011011111000	Signal	In
packet3	0000101100001111	Signal	In
packet4	0000110000111111	Signal	In
clk	1	Signal	In
output1	00111111	Signal	Out
output2	11110000	Signal	Out
output3	11111000	Signal	Out
output4	00001111	Signal	Out
input1	11110000	Signal	Internal
input2	11111000	Signal	Internal
input3	00001111	Signal	Internal
input4	00111111	Signal	Internal
source_address1	00	Signal	Internal
dest_address1	01	Signal	Internal
source_address2	01	Signal	Internal
dest_address2	10	Signal	Internal
source_address3	10	Signal	Internal
dest_address3	11	Signal	Internal
source_address4	11	Signal	Internal
dest_address4	00	Signal	Internal

Fig 5.3 object window for all packet data

D) Wave Simulation Window For all packet data



E) RTL View



IX. CONCLUSION

The developed crossbar switch architecture presented advantages due to its flexibility and high performance. This fact justifies its employment in a network processor. The capability of adapting the topology implemented on crossbar switch to the environment changes generates high performance for data processing in several situations as multiprocessors and computer clusters could be reached with modifications in the contribution of this paper is the proposed RCS-2 architecture. The first level of reconfiguration of the RCS-2 could be reached through the codification of the architecture using a hardware description language, allowing it to be implemented in several devices with dimensions determined by device capacity. The second level of reconfiguration the matrix of connections. These modifications generate an overhead. However, through the experiments, it was evidenced that the overhead time is less than the speedup obtained through the topologies implementation in RCS-2. Therefore, the RCS-2 has a better performance when compared to a TCS.

REFERENCES

- [1]. D. E. Comer, "Network Systems Design using Network Processors", Prentice Hall, 2003
- [2]. D. Kim, K. Lee, S. Lee and H. Yoo, "A Reconfigurable Crossbar Switch with Adaptive Bandwidth Control for Networks-on-Chip", IEEE International Symposium on Circuits and Systems, 2005
- [3]. G. Lawton, "Will Network Processor Units Live up to their Promise?" IEEE Computer, Volume 37, Number 4, April, 2004,
- [4]. H. Eggers, P. Lysaght, H. Dick, and G. McGregor, "Fast Reconfigurable Crossbar Switching in FPGAs", Proceedings of 6th International Workshop on Field Programmable Logic and Applications, Springer LNCS 1142, 1996,
- [5]. I. A. Troxel, A. D. George and S. Oral, "Design and Analysis of a Dynamically Reconfigurable Network Processor", IEEE Conference on Local Computer Networks, November 6-8, 2002
- [6]. J. Chang, S. Ravi, and A. Raghunathan, "FLEXBAR: A crossbar switching fabric with improved performance and utilization", IEEE Custom Integrated Circuits Conference, May 2002,
- [7]. L.E.S. Ramos and C.A.P.S. Martins, "A Proposal of Reconfigurable MPI Collective Communication Functions". Third International Symposium on Parallel and Distributed Processing and Applications, LNCS 3758, Nanjing, China, November 2-5, 2005
- [8]. S. Young, et al., "A High I/O Reconfigurable Crossbar Switch", 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Napa, California, April 09-13, 2003