Abstract—The implementation of read-time communication over existing Ethernet architecture is a subject of great interest. Various methods have been proposed to achieve this goal. Thus, the concepts of a scalable architecture that interconnects multiple instances of the Network Code Processor (NCP), which is an Application-Specific Instruction Set Processor coordinates access to the shared medium by executing Time Division Multiple Access (TDMA) based schedules on each node (ASIP) that running concurrently on a single FPGA fabric emerges. By using programmable TDMA schedules to coordinate passing data between the core instances, the system provides a dedicated hard real-time switching service on top of Ethernet. In this paper we have discussed the possible limitations of this architecture and how to improve this work for better performance.

Index Terms — Real Time Distributing System, Multi-Core switch, Amdahl’s law.

I. INTRODUCTION

To achieve the two utmost goals of distributed hard real-time systems - Predictability and Validity, researchers have worked both on software level and hardware level. At the software level, the advent of Network Code Processor (NCP) helped researchers to overcome the limitations of expensive computational resources and to reduce high jitter in the communication delays, making it suitable for soft real-time systems [1]. The next step was to implement it in the existing Ethernet infrastructure (hardware level), which, in turn, was not suitable for real time communication as it needed guaranteed bounded communication delay. The implemented system using Field Programmable Gate hardware (FPGA) provides a throughput close to the line rate of a 100 Mbps Ethernet network with high predictability of communication delay [2]. FPGA allows cycle-accurate simulation and offers similar delays on each board instance while software-based real-time communication frameworks produce timing variance for each code instruction and action differs among workstations, because of differences among interrupt controllers, motherboard, and processors. Thus the Paper "A TDMA Ethernet Switch for Dynamic Real-Time Communication: Limitations & Extensions" is an inevitable outcome of “Hardware Acceleration for Conditional State-Based Communication Scheduling on Real-Time Ethernet [2]” where the shift from a Single-Core Network Code Processor (SC-NCP) to a Multi-Core Network Code Processor (MC-NCP) seems logical. The Multi-core NCP delivers a switching throughput that exceeds 200 Mbps and other facilities.

II. OVERVIEW OF NETWORK CODE SWITCH

Network code switch contains multiple instances of the Multi-Core Network Code Processor (MC-NCP) core, the main functional block that encapsulates all the components showed in Figure 1 for the single-core processor, with the exception of the Msg-data memory block. Then, Single-Core Network Code Processor (SC-NCP), the hardware ASIP that executes the programmed schedules. The system has three main components: the memory space, the Network Code Core (NCC), and the Ethernet core. Lastly, the Network Code framework enables implementing real time communication systems for distributed applications. Its main components include: (1) a domain-specific language to represent state-based TDMA schedules, (2) a compiler with a verification engine that translates the programs into checked executable schedules, and (3) the interpreter entity that executes the schedule.

Fig. 1. Block diagram of and Network Codes Switch
III. LIMITATIONS AND EXTENSIONS

3.1. Dependency of Multi-core concurrency with code efficiency

The Network Code Switch (NCS) model expands the superscalar architecture of the Network Code Processor to a multi-core device that connects multiple instances of the SC-NCP running concurrently in a single chip, offering a Network Code-based real-time packet switching service to external Ethernet nodes. This is the concept of multiprocessing that is the execution of concurrent processes by running them on separate processors which all access a shared memory. Upon introducing this concept, we need to take into account that the speedup of a program using multiple processors in parallel computing is limited by the sequential fraction of the program [4]. For example, if 95% of the program can be parallelized, the theoretical maximum speed up using parallel computing would be 20× as shown in the diagram, no matter how many processors are used as Figure 1 suggests. On the other hand, the maximum speed up in an improved sequential program, where some part was sped up P times is limited by inequality

$$\text{Maximum Speedup} \leq \frac{P}{1 + f(P - 1)}$$

Where $f(\log f(1))$ is the fraction of time (before the improvement) spent in the part that was not improved.

3.2. OPTIMUM NUMBER OF CORES

These lead us to choose the optimum number of cores to utilize the amount of used resources with desired speed. Consequently, [3] shows that it is complicated to extract a direct relation between the number of cores and the clock speed. For a 1Gbps Ethernet connection, the PHY logic runs at 125 MHz. Because Fig. 2. Speedup of the code with respect to number of cores this speed is faster than the requirements for the NCC, each instance will work on two different clock domains: one driving the super-scalar NCC block that executes the schedule, and another one driving the Ethernet MAC logic that interfaces the PHY. The NetFPGA board provides the 125 MHz clock for the PHY, and we use a shift register to generate a 41 MHz common for all the instances. Thus the correlation between efficiency of the NCL, Number of core and the clock speed brings a new dimension in the design procedure.

3.3. PORT NUMBER

In Reference [4] has examined a system designed for the parallel execution of the Fast Fourier Transformation (FFT) computations in a multi-core system on FPGA. It is well-known, that the common shared memory is a “bottleneck” in shared memory systems. For that reason, they have examined systems with different number of ports (one, two, and four). Some of the Important results of the investigations are shown on Fig.2. The results of Figure 2 suggest that the growth of memory port number causes speedup augmentation in the multi-core system with shared memory.

3.4. MOVING ON TO MORE COMPLEX MC-NCP ARCHITECTURE

Even though the purpose of the paper is to interconnect multiple instances of the NCP running concurrently on a single FPGA fabric, the next step should be utilizing the current complex multi-core architecture to take the full advantage of hardware parallelism.

IV. CONCLUSION

We presented our work on the drawbacks of moving from a single-core to a multi-core design in the specific case of a bespoke processor for real-time communication. The architecture relies on the previously reported Network Code Processor which led us to design the architecture with the concepts of code-reuse and modularity in mind. The various drawbacks and their solution should be explored as further work.

REFERENCES


AUTHOR BIOGRAPHY

Asif Khan was born on 1987 in Dhaka, Bangladesh. He has completed his B.Sc. from North South University in ETE in 2011. Currently he is working in Maker communications, as an Engineer, which is a group of companies. His research interests include Wireless sensor network , parallel and distributed computing, wireless communication etc. Email: alaxwest@yahoo.com

Shohel Ahemd was born on 1987 in Bikrompur, Bangladesh. He has completed his B.Sc. from AUST in EEE in 2009. Currently he is working in Esquire Knit composite Ltd. as an Engineer, which is a group of companies. His research interests include power electronics, renewable and automation and instrumentation, VLSI, microelectronics, control system, wireless communication etc. He is a member of IEB. Email: mredulee@yahoo.com

A.K.M. Niaz Morshed was born on 1987 in Sirajgonj, Bangladesh. He has completed B.Sc. in Electrical and Electronic Engineering from the Ahsanullah University of Science & Technology (AUST), a subsidiary organ of DAM (The organization of Dhaka Ahsania Mission), Bangladesh in 2009. He has previously worked at ZTE Corporation as Transmission Engineer, which is one of the International Telecommunication Vendor. Besides, he has various research interests including Microwave Engineering, HCI, Renewable energy, Distributed Computing, Network Security and UNIX programming etc. He is also a member of IEB. E-mail: akm_niaz@yahoo.com