An Extensive Review on a Practical NoC Design for Parallel DES Computation

Nisha Tripathi[1], Prof. Aditi Bhatt[2]

[1]M Tech Research Scholar, [2]Research Guide, Deptt. of Electronics & Communication,

Sagar Institute of Science & Technology, Ganghi

Abstract: Network security is integral to every part of the system such as memory, chipsets, storage devices, controller devices, display devices, motherboards, and bus interconnects. In this review paper we study and understanding the on-chip bus security is a key component of hardware security because the bus is an essential link for the entire system. No matter how secure other hardware devices are, if the bus is insecure then the risk of data compromise is high. Hardware security requires careful management of the hardware and software, such that system components obey the overall system security policy.

Keywords:- The Network-on-Chip (NoC), Multi-Core NoC (MCNoC), Data Encryption Standard (DES) & FPGA.

I. INTRODUCTION

Integrated Circuit (IC) bus protocol analysis allows chip designers and computer security professionals to understand, identify, and mitigate security vulnerabilities with respect to confidentiality, integrity, and availability. To better understand the bus protocols that this research investigates, this study provides background on IC bus architecture concepts that includes bus communication components, System-on-Chip (SoC) designs, security policies, and threat models.

Core Concepts of IC Bus Communications

Buses are defined as a set of wires that acts as a shared, but common, data path to connect multiple subsystems within a system. Buses are frequently used to transmit information from one component to another and their architectures are based on one of two forms of information flow—parallel or serial form. Parallel buses carry data on multiple wires (several bits of data are sent at the same time along multiple paths), whereas serial buses carry data in a bit serial form (bits of data are sent one at a time along a single path). These design choices are based on platform requirements and the designers' needs.

Data is transferred from one device to another on a system. Each device is assigned a role of master, slave, or sometimes both, depending on the platform and particular bus architecture. A master controls the data traffic on the bus. It is responsible for initiating a session by making a read or write request to a device that is designated as a slave. Inversely, a slave performs a service at the behest of a master device. A practical example of a master and slave interaction is if main memory (master device) wanted to write data to a display device (slave).

Clock signals are used to inform master and slave devices when to start, stop, and what speed the data is travelling at. A clock line is a path for which systems' clock signals oscillate between high (logical one) and low (logical zero) states. The clock is responsible for speed, bit width, and coordinated actions of a circuit during a bus operation. A clock generator produces clock signals, which synchronize data movement on the data line.

Through any interaction between a master and slave, there has to be assurance that there is no loss of information through bus collisions, deadlocks, or unauthorized use of devices (i.e., depending on the platform, some devices are not permitted to interact with other devices). Every bus architecture has some form of arbitration to ensure that these policies are enforced. Arbitration is the process of determining which bus master will and can obtain access to the bus (e.g., to prevent more than one master from transmitting simultaneously to one slave). Methods of arbitration fall into the categories of either centralized or distributed. A centralized scheme uses a single hardware device, often called an arbiter, which is responsible for allocating time on the bus. The arbiter may be a separate device or part of the processor. In a distributed scheme, there is no central controller. Arbitration is managed through access control logic and all modules act together to share bus resources.

Next generations of general-purpose many-core processors point out to exploration of parallel programs in order to achieve high-performance computing. In this context, there are several problems related to different fields and levels, such as [1]: operating systems, algorithms, compilers and architectures. Focusing on architecture, on-chip interconnections are important to support, e.g., collective communication patterns [2]. For this reason, a good choice of network architecture can reduce the packet transmission time in order to increase the performance of parallel programs.

Traditional on-chip interconnection architectures such as buses and crossbar switches have scalability problems. In many-core processors a single and large interconnection reduces the performance due to the wire constraints [3], for instance, resistance and routing. The state-of-the-art points out the Network-on-Chip (NoC) [4] as the main alternative to support a large number of processing cores or on-chip devices.

The continued progress in VLSI technologies allows us to put more cores (from dozens to hundreds) on a single chip to build a system-on-a-chip (SoC) system [11]. This kind of SoC system is commonly found in embedded systems which are prevalent in every aspect of our daily life, such as mobile terminals, portable game consoles, personal media players, etc. It is also found in battlefield, for example, unmanned aerial vehicles (UAVs), killer robots, etc. It is gaining popularity that various services are expected to be integrated on an SoC-based embedded system, e.g., a smart phone can handle phone calls, play multimedia files, access to the Internet, etc., simultaneously. On the other hand, the growing development of multimedia applications (e.g., high quality video) and high data rate wireless technologies (e.g., UWB, WiMax) are driving the need for higher computational power in such system. This also requests for sustained long-period operation of the system (e.g., watching a live soccer match using a mobile terminal with high-quality Long-time monitoring using UAVs. However, high computational requirement and sustained long-period operation are often two contradicting goals. In essence, energy efficient operation of SoC systems is highly required.

More specifically, in a study conducted by MorphICs, it is concluded that the algorithmic complexity of future applications grows much faster than the processor performance, which is governed by the Moore's Law. However, the battery technology grows at a much lower rate. This means that an embedded system such as a mobile terminal is expected to execute dramatically more computation-intensive tasks, but the increase in battery capacity is simply more than offset by the increase in computational requirement. This suggests that, in order to conserve energy and hence to prolong the lifetime of the system, the computational tasks should be executed in an intelligent way.

To meet computational need and at the same time reduce energy requirement, the growing trend is to apply parallel processing by employing a number of relatively less capable processing cores, instead of one big, power-hungry core. In the embedded systems programming survey conducted in 2005, it was reported that nearly 50% of chips used multiple processors and over 100 projects used more than 10 processors. Moreover, nearly two-third of SoC's were heterogeneous multi-processor. The future trend is to employ heterogenous multi-core system-on-a-chip.

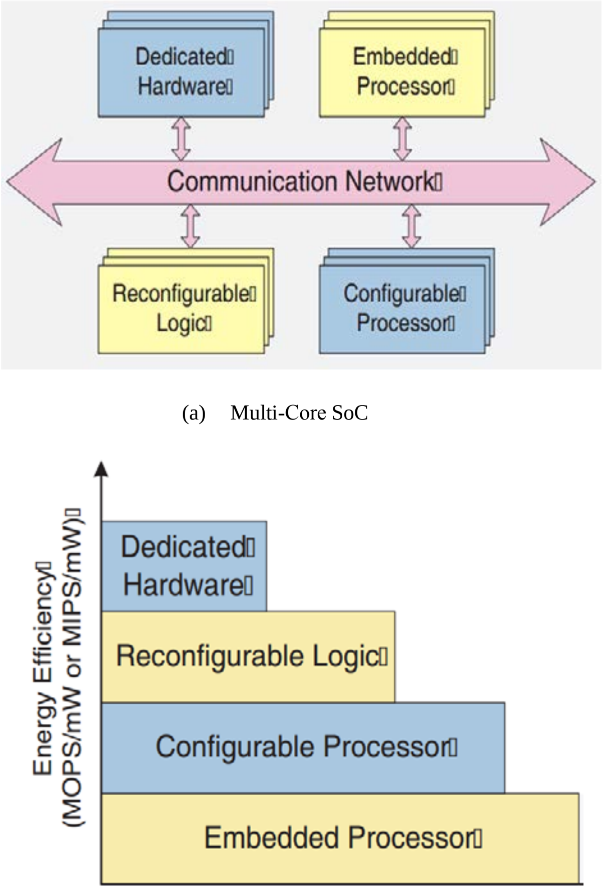
Figure 1 shows the system architecture block diagram a typical heterogeneous multi-core SoC. In a system, there is a set of heterogeneous cores (processing elements) of different flexibility and computation characteristics allowing optimal execution of different tasks in the system. Specifically, these processing elements can be reconfigured / utilized to save energy in various ways:

1. Embedded processor supports dynamic voltage scaling (DVS), on-demand shut-down, or power-up;

2. Reconfigurable logic (e.g., Field Programmable Gate Array and Field Programmable Object Array) acts as off loader to implement computation intensive algorithms in hardware;

3. Configurable processor (e.g., [7, 19, 21]) customized to execute specific tasks; and

4. Dedicated hardware (e.g., cryptographic cores, DSP cores) for most energy efficient executing of tasks.



(a) Multi-Core SoC

(b) Energy efficiency vs. Flexibility

Using a very popular FPGA in the research community the study was able to emulate 3x3 No architecture with 8b data paths using 27% of the FPGA resources. This NoC is not a virtual channel implementation, but does make for a light-weight router architecture which was the intention of the project. The open source NoC emulation project, NoCem is the NoC architecture based our work from as it is freely available.

II. LITERATURE SURVEY

R. Yuan, S. J. Ruan and J. Götze, [1] The Network-on-Chip (NoC) is considered to be a new SoC paradigm for the next generation to support a large number of processing cores. The idea to combine NoC with homogeneous processors constructing a Multi-Core NoC (MCNoC) is one way to achieve high computational throughput for specific purpose like cryptography. Many researches use cryptography standards for performance demonstration but rarely discuss a suitable NoC for such standard. The goal of this paper is to present a practical methodology without complicated virtual channel or pipeline technologies to provide high throughput Data Encryption Standard (DES) computation on FPGA. The results point out that a mesh-based NoC with packet and Processing Element (PE) design according to DES specification can achieve great performance over previous works. Moreover, the deterministic XY routing algorithm shows its competitiveness in high throughput NoC and the West-First routing offers the best performance among Turn-Model routings, representatives of adaptive routing.

H. Cota de Freitas, L. M. Schnorr, M. A. Z. Alves and P. O. A. Navaux, [2] Due to the multi-core processors, the importance of parallel workloads has increased considerably. However, many-core chips demand new interconnection strategies, since traditional crossbars or buses, common for current multi-core processors, have problems related to wires and scalability. For this reason, Networks-on-Chip (NoCs) have been developed in order to support the performance and parallelism focused on several workloads. Although a Network-on-Chip is a good option, most designs consist of a large number of routers. These routers are responsible for forwarding packets, and consequently, for supporting message-passing workloads. In this context, the NoC performance is a problem. Therefore, the main goal of this paper is to evaluate the impact of well-known parallel workloads on NoC architecture design. In order to achieve high performance, the results point out to parallel workloads with small packets and cluster-based NoCs with circuit switching and adaptable topologies.

Tyrone Tai-On Kwok and Yu-Kwong Kwok, [3] with the continued progress in VLSI technologies, authors can integrate numerous cores in a single billion-transistor chip to build a multi-core system-on-a-chip (SoC). This also brings great challenges to traditional parallel programming as to how authors can increase the performance of applications with increased number of cores. In this paper, authors meet the challenges using a novel approach. Specifically, authors propose a reconfigurable heterogeneous multi-core system. Under our proposed system, in addition to conventional processor cores, authors introduce dynamically reconfigurable accelerator cores to boost the performance of applications. Authors have built a prototype of the system using FPGAs. Experimental evaluation demonstrates significant system efficiency of the proposed heterogeneous multi-core system in terms of computation and power consumption.

Y. S. Yang, J. H. Bahn, S. E. Lee and N. Bagherzadeh, [4] The computational performance of network-on-chip (NoC) and multi-processor system-on-chip (MPSoC) for implementing cryptographic block ciphers can be improved by exploiting parallel and pipeline execution. In this paper, authors present a parallel and pipeline processing method for block cipher algorithms: data encryption standard (DES), triple-DES Algorithm (TDEA), and advanced encryption standard (AES) based on pure software implementation on an NoC. The algorithms are decomposed into task loops, functions, and data flow for parallel and pipeline execution. The tasks are allocated by the proposed mapping strategy to each processing element (PE) which consists of a 32-bit reduced instruction set computer (RISC) core, internal memory, router, and Network Interface (NI) to communicate between PEs. The proposed approach is simulated by using networked processor array (NePA), the cycle-accurate System and hardware description language (HDL) model platform. Authors show that our method has the advantage of flexibility as compared to previous implementations of cryptographic algorithms based on hardware and software co-design or traditional hardwired ASIC design. In addition, the simulation result presents that the parallel and pipeline processing approach for software block ciphers can be implemented on various NoC platforms which have different complexities and constraints.

G. Schelle and D. Grunwald,[5] The network on chip will become a future general purpose interconnect for FPGAs much like today psilas standard OPB or PLB bus architectures. However, performance characteristics and reconfigurable logic resource utilization of different network on chip architectures vary greatly relative to bus architectures. Current mainstream FPGA parts only support very small network on chip topologies, due to the high resource utilization of virtual channel based implementations. This observation is reflected in related research where only modest 2times2 or 2times3 networks are demonstrated on FPGAs. Naively it would be assumed that this complex network on chip architectures would perform better than simplified implementations. Authors show this assumption to be incorrect under light network loading conditions across 3 separate application domains. Using statistical based network loading, a synthetic benchmarking application, a cryptographic accelerator, and a 802.11 transmitter are each demonstrated across network on chip architectures. From these experiments, it can be seen that network on chips with complex routing and switching functionality are still useful under high network loading conditions. Additionally, it is also shown for our network on chip implementations, a simple solution that uses 4-5times less logic resources can provide better network performance under certain conditions.

III. PROBLEM IDENTIFICATION

The results show a high throughput DES computation design can be achieved with low-cost switching, packet format and routing algorithms in a 5 × 5 mesh-based MC NoC. Using large PE is area efficient to FPGA and having PE processing time longer than routing time is a key factor for PE architecture selection. This paper also shows the uneven PE utilization in XY routing and biased routing algorithms in Turn-Model causing none negligible performance loss. Finally, a NoC with considerations of DES architecture adds great throughput to the final design, 5 times faster than the best performance in previous works.

IV. CONCLUSION

The increased demand on computer systems and information sent over networks makes it essential to take steps to protect the systems and information from known risks. On vast networks such as the Internet with no central administrator, the risk is even greater as every computer along the route that the data traverses can attack what is being sent or received. Fortunately, numerous techniques have been developed to keep the data secure and private. The essential technology underlying virtually all automated network and computer security applications is known as encryption. Encryption was primarily used for military and espionage use.

REFRENCES

[1] R. Yuan, S. J. Ruan and J. Götze, "A practical NoC design for parallel DES computation," VLSI Design, Automation, and Test (VLSI-DAT), 2013 International Symposium on, Hsinchu, 2013, pp. 1-4.

[2] H. Cota de Freitas, L. M. Schnorr, M. A. Z. Alves and P. O. A. Navaux, "Impact of Parallel Workloads on NoC Architecture Design," 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing,

[3] Tyrone Tai-On Kwok and Yu-Kwong Kwok, "On the design, control, and use of a reconfigurable heterogeneous multi-core system-on-a-chip," Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on, Miami, FL, 2008, pp. 1-11.

[4] Y. S. Yang, J. H. Bahn, S. E. Lee and N. Bagherzadeh, "Parallel and Pipeline Processing for Block Cipher Algorithms on a Network-on-Chip," Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on, Las Vegas, NV, 2009, pp. 849-854.

[5] G. Schelle and D. Grunwald, "Exploring FPGA network on chip implementations across various application and network loads," 2008 International Conference on Field Programmable Logic and Applications, Heidelberg, 2008, pp. 41-46.

[6] J. Duato, A. Robles, F. Silla, and R. Beivide. “A Comparison of Router Architectures for Virtual Cut-Through and Wormhole Switching in a NOW Environment”. In Proceedings of the 13th International Symposium on Parallel Processing and the 10th Symposium on Parallel and Distributed Processing, pages 240–247, 1999.

[7] N. Banerjee, P. Vellanki, and K.S. Chatha. “A Power and Performance Model for Network-on-Chip Architectures”. In Design, Automation and Test in Europe Conference and Exhibition. Proceedings, volume 2, pages 1250–1255 Vol.2, Feb. 2004.

[8] A. Kumar, A. Hansson, J. Huisken, and H. Corporaal, “An FPGA Design Flow for Reconfigurable Network Based Multi-Processor Systems on Chip,” Proc. De sign, Automation and Test in Europe (DATE 2007), pp. 33–38, Apr. 2007.

[9] T. T.-O. Kwok and Y.-K. Kwok, “On the Design of a Self-Reconfigurable SoPC Based Cryptographic En gine,” Proc. ICDCS 2004 International Workshop on Embedded Computing, pp. 876–881, Mar. 2004.

[10] T. T.-O. Kwok and Y.-K. Kwok, “Practical Design of a Computation and Energy Efficient Hardware Task Scheduler in Embedded Reconfigurable Computing Systems,” Proc. 13th Reconfigurable Architectures Workshop (RAW 2006), Apr. 2006.

[11] P. Magarshack and P. G. Paulin, “System-on-Chip Be yond the Nanometer Wall,” Proc. 40th Design Au tomation Conference (DAC 2003), pp. 419–424, June 2003.

[12] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, “Performance Evaluation and Design TradeOffs for Network-on-Chip Interconnect Architectures,” IEEE Transactions on Computers, vol. 54, no. 8, pp. 1025–1040, Aug. 2005.

[13] A. Patel, C. A. Madill, M. Saldana, C. Comis, R. Pomes, and P. Chow, “A Scalable FPGA-based Multiprocessor,” Proc. 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2006), pp. 111–130, Apr. 2006.

[15] P. G. Paulin, C. Pilkington, M. Langevin, E. Bensoudane, D. Lyonnard, O. Benny, B. Lavigueur, D. Lo, G. Beltrame, V. Gagne, and G. Nicolescu, “Parallel Programming Models for a Multiprocessor SoC Platform Applied to Networking and Multimedia,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 7, pp. 667–680, July 2006.