

Possibilities of using reconfigurable devices for improving computational efficiency

László Kazup, János Végh, Ádám Kicsák, Zsolt Bagoly

University of Debrecen

kazup.laszlo@gmail.com, janos.vegh@unideb.hu, pixels@vipmail.hu,
ba_zsolt@yahoo.com

Abstract

The today's computers are much more powerful than their predecessors in the heroic age of information technology. They comprise several CPU cores and large working memory with high clock frequency and lots of additional controllers which perform many complicated functions for example I/O functions, storage management, direct memory access for peripherals, etc. There are much more intelligent devices within a computer than John von Neumann could have imagined in the 40's. But the basic architecture of the system is still the same: the CPU is the most intelligent device and it has to transact all of important operations and decisions. Other controllers operating on non-Neumann principles get only secondary and less important functions. However, devices implementing the necessary judgments in software by the operating system are available. Unfortunately this method requires a lot of instructions to execute.

In this paper we describe a few possibilities supporting a single- or a multi-core CPU by reconfigurable devices, typically FPGAs. We show more special cases when a Neumann type architecture is extended by non-Neumann controllers, such as supporting the CPU by handling semaphores, software interrupts, etc., that is replacing traditionally hardware functions by software of the operating system. This method works for the traditionally hardware controlled processes (for example serial communication, direct memory access, interrupts, etc.) perfectly for a long while. By this way, hardware tasks will be created which will replace the computationally expensive software operations. There are already systems which demonstrate how hardware and software tasks can work together. Not least, these implementations result in a computationally highly effective and deterministic systems. This is a significant benefit, especially in case of real-time operation systems.

Keywords: reconfigurable, Neumann, FPGA

1. Introduction - Trends of today's computing

As it is wellknown, the silicon-based technology reached its physical limitations a few years ago. Hardware manufacturers can not produce computing hardware with higher clock frequency, but there are needs for more computational power nowadays. There are two significant trends for improving the efficiency of computing.

1.1. Improving of computational efficiency by using multicore systems

The computational efficiency can be improved by the parallelisation of instruction's execution. This method needs more CPU cores within a system. By using more CPU-s the unrelated instructions can be executed in the same time. However, the Neumann-principles assume "in-order" running. Therefore, this improving method needs the follow-up ordering of results. This fact is the bottleneck of today's multi-core systems.

1.2. Improving of computational efficiency by using hybrid computers

In the recent years the hybrid architectures have been improving dramatically. These systems contain reconfigurable fabrics for realizing more and more classic software functions. Modern computers include also special ASICs for implementing communication protocols, handling interrupts, managing the direct memory access or accelerating the graphical visualization, however, reconfigurable devices operating in cooperation with the CPU have more benefits. Software tasks can be also implemented in hardware by using these special flexible hardware components. It can be also feasible to create standard operating system functions by using FPGA blocks. This possibility is due to the ability of reconfiguration and it results an improved computational efficiency. More and more hardware manufacturers devote time and money for the development of hybrid computers.

2. Basics of reconfigurable devices

There are two significant types of reconfigurable devices. Programmable Logic Devices (PLDs) and Complex Programmable Logic Devices are the first generation of reconfigurable logic arrays which can be realized in various logic networks by using programmable interconnections between logic gates. However, the complexity of these networks is strongly limited. Due to this limitation these devices can perform only the most necessary user-defined functions, and the standard peripherals such as communication interfaces, SRAM blocks, etc. have to be added as external components.

The next type of programmable devices is the Field Programmable Gate Array (FPGA). In contrast to PLD/CPLD devices the FPGAs includes static RAM blocks

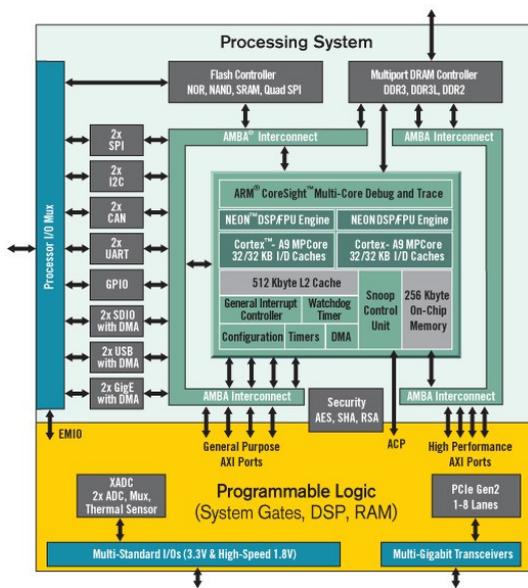


Figure 1: Functional block diagram of Xilinx Zynq 70xx “All-Programmable” SoC

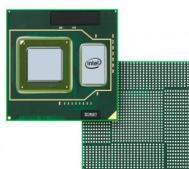


Figure 2: Intel Atom E600C series – the first configurable Atom-based processor

for generating sum-of-product terms containing the truth table in the RAM cells. These units are named as “Logic Cells”. Since this structure these logic cells can produce the output variable within a single cycle time of reading an SRAM cell which is only a few nanoseconds. So, every logical unit realizes a bit of the whole hardware and every little unit can work in parallel with each other. This gate-level parallelisation is in opposite the standard Neumann-architecture. In addition, thousands (or millions nowadays) of logic cells, very complex logic functions can be implemented on a mass-produced, cheap integrated circuit, it means, FPGAs fill the gap between CPLDs and Application-Specified Integrated Circuits (ASICs). Due to the flexibility and the complexity these devices are ideal for realizing reconfigurable hardware accelerators in a Neumann-system.

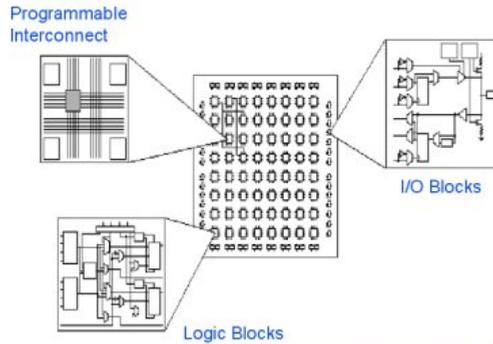


Figure 3: Basic architecture of FPGA devices

3. Accelerators for the CPU by using reconfigurable devices

For decades the Neumann-based computers contain lots of non-Neumann accelerators such as interrupt controller, DMA controller, communication controllers, etc. These devices are closely integrated with the CPU and these are made up as application-specified integrated circuits (ASICs). Using this method reconfigurable (RC) devices can be utilized as system accelerators. Due to the ability of reconfiguration lots of different functions can be realized by using RC fabrics. However, when the interface between the CPU and the accelerator isn't enough effective, it results a poor improvement of computational efficiency. There are different methods for connection of RC fabrics to the Neumann-system.

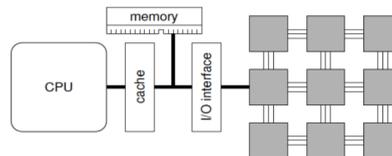


Figure 4: RC fabrics are connected to the I/O interface

In the first case (see on Figure 3.) the reconfigurable device is connected to an I/O port. This possibility is the easiest way to connect the RC fabric to the system, but it has the worst performance among of all because the I/O interface provides limited bandwidth.

The following arrangement (see on Figure 4.) offers the maximum integration between the software components and the reconfigurable accelerators. This lay-

out integrates the RC devices directly into the data path of the CPU, thus the accelerator possess each important information of the running CPU.

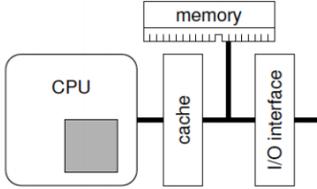


Figure 5: RC fabrics are connected into the data path

The latest and the most trendiest option of integration between RC devices and the Neumann-system is shown on Figure 5. These systems embed the CPU (or CPUs) within the reconfigurable fabrics. This is the only possibility with advances in the reconfigurable hardware technologies. The CPU can either be implemented physically or as a soft processor. Among other things, this layout has much higher performance potential than the previous compositions and it breaks away from the processor-centric compute model. Today’s hybrid computers are based on this topology.

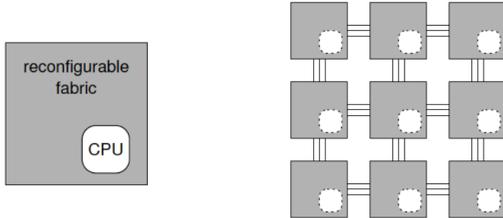


Figure 6: CPU within RC fabrics

4. Compatibility with the operating system

In the first case the operating system can communicate with the accelerator by using standard I/O commands. The software system have to be extended by special device drivers only for using the hardware accelerator because this one operates as an add-on card. For example, this protocol is utilized for using hardware accelerators on Xilinx Zynq70xx. However, by using the second or the third architecture

both the basic architecture and the operating system have to be modified. Additional to the above described phenomena user-defined CPU instructions are needed for increasing the efficiency of reconfigurable accelerators performance. When it's possible, frequently-used and computationally expensive operating system functions can be realized as hardware tasks.

5. Summary

The combination of a Neumann-CPU and reconfigurable systems results a powerful hybrid system which produces more performance than the standard systems due to the ability of hardware task's gate-level parallelisation. It seems to be a good idea to realize a really effective hybrid architecture, but the manufacturers are still looking et for the bestway of this combination. The most difficult task of this research is to find an architecture which is compatible with the most popular operating systems and the existing applications. It isn't impossible but it needs deep knowledge about the RC and computing technologies and complex toolchain.

References

- [1] Xilinx Homepage – <http://www.xilinx.com>
- [2] Altera Corp. Homepage – <http://www.altera.com>
- [3] Intel Homepage – <http://www.intel.com>
- [4] Hayden Kwok-Hay So, Robert W. Brodersen: BORPH: An Operating System for FPGA-Based Reconfigurable Computers (Electrical Engineering and Computer Sciences University of California at Berkeley, 2007)
- [5] Altera Cyclone II Specification – Altera Corporation
- [6] Epiphany Architecture Reference – Adapteva Inc.