Aquarius II – A platform for dynamic reconfigurable systems prototyping

Abstract

The programmable logic devices, FPGAs (Field Programmable Gate Arrays) have been used for a long time, especially as tools for digital circuits prototyping. However, this reality has changed and new applications are possible, since the computational power of these devices increased, cost decreased and their capacity of run-time reconfiguration (dynamic reconfiguration) were implemented.

However, despite all advantages presented, this technology has not been largely used on computational applications mainly because of the complexity of these systems development.

This article presents Aquarius II, a platform able to efficiently manage the resources provided by FPGAs, through an operational system for reconfigurable systems prototyping.

1. Introduction

The technology of programmable logic devices has evolved considerably and presented several advantages over conventional architectures: performance gains of around 10 to 100 times, compared with conventional CPUs; reduced consumption of energy, important feature both in embedded applications, as in computers clusters; significant decrease in the price of these devices; possibility of dynamically reconfiguration [1][2][3].

However, despite the many advantages offered, this technology is not yet widely used to perform computing due to the complexity of the development process of these systems [4][5].

Even with this difficulty, the FPGAs are becoming an extremely promising technology in cost-benefit ratio, especially in applications that require high performance computing. Several companies have begun to provide supercomputers with FPGAs incorporated into their systems [3][6][7][8].

In this context, this paper presents a platform based on an operating system capable of managing the computing resources offered by FPGAs in a transparent manner to the developer in order to alleviate the problem presented. This platform aims to allow the prototyping of reconfigurable systems that exploit the parallelism provided by the FPGA architecture.

2. Aquarius II Platform

The first version of the Aquarius platform was developed at the Center for Informatics of the Federal University of Pernambuco and includes the functions of total and partial dynamic reconfiguration of a Xilinx Virtex-II FPGA [10] through a control platform. However, the Aquarius did not allow any kind of communication between the control platform and the IPCore configured in the FPGA. This limitation prevented that more complex data processing applications were implemented.

The second version, presented in this article, solved this limitation, through an architecture that enables communication between the control platform and the memory of the reconfigurable core through resources offered by an operating system.

2.1. Architecture Overview

The Aquarius II platform is a hybrid platform composed of an Altera development board [12], with a Stratix II FPGA [11], which is the control platform, and a Xilinx development board [13], called reconfigurable platform, which is equipped with a Virtex-II FPGA II [10] which represents the co-processor or the system's reconfigurable device.

In the control platform are mapped: the soft-core processor, NIOS II; the IPSelectMap core (already implemented in the first version of Aquarius [9])), responsible for total and partial dynamic reconfiguration of the Virtex-II FPGA; and IPCommCore, implemented in Aquarius II. This one is responsible for communication between the control platform and the core's memory in the reconfigurable platform. All control platform IPs communicate through the Avalon bus [14]. Also are members of this platform, the operating system uCLinux [15], ported to Altera NIOS II processor [16], and the device drivers for IPSelectMap and IPCommCore. The Figure 1 presents the platform architecture.

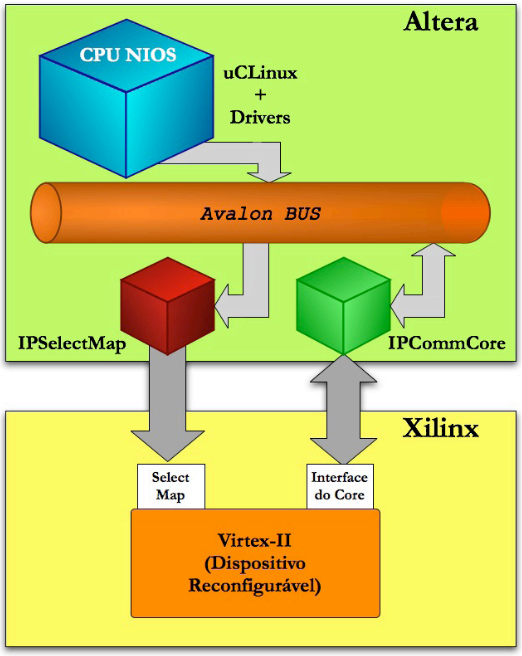

Figure 1 - Aquarius II Architecture Overview

2.2. Hardware Support

2.2.1. IPCommCore

The IPCommCore is the core that effectively makes the communication between the control platform and the reconfigurable platform. It allows both data and commands to be sent between the two platforms.

For the IPCommCore communicate with the core mapped in the co-processor device, was defined a standard interface for communication with the reconfigurable cores (Figure 2).

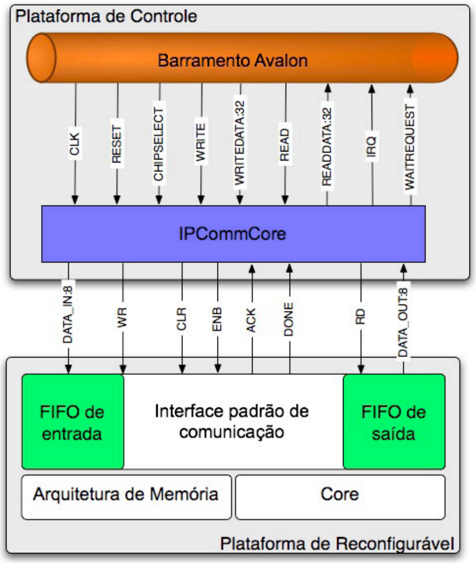

Figure 2 - Interfaces between Avalon Bus, IPCommCore and The Reconfigurable Core Interface

2.2.2. Standard Communication Interface

The communication interface is the entry point of data coming from the control platform to the reconfigurable cores memory of the application. This interface receives data in its entry FIFO and distributes in the internal memory architecture of the reconfigurable core. The signals of this interface are shown in the Figure 2.

2.3. Software Support

2.3.1. uCLinux

Four main factors influenced the choice of uCLinux for the platform: it is free and open source which allows complete freedom in carrying out changes; use of prior knowledge of the Linux operating system in the implementation of device drivers [17]; has a large community of users who provide a constant support, and finally has a port to NIOS II processor used in Aquarius II platform [16].

2.3.2. Device Driver

For the uClinux access the device, it was necessary the implementation of a device driver responsible for its control. On Linux all devices are seen as a file of the file system and an operating system API provides a number of functions for access these files. These API functions have been redefined inside the device driver to allow access to the communication resources provided by IPCommCore.

3. Experimental results

To validate the proposed architecture, was developed an application of 8 bits integer numbers multiplication, from a reconfigurable core that implements the multiplication.

The core, in fact, is composed, in addition to the multiplication core, a memory architecture and an interface that defines how access is performed as mentioned previously in Section 2.2.2.

The experiments are carried out by repeated integer numbers multiplications in hardware. The processing time of these multiplications are presented in Table 1.

Table 1 - Mutiplication Processing Time

Number of Multiplications	Processing Time
1	7,76 ms
2	10,64 ms
4	17,10 ms
8	25,9 ms

The experiments presented here were used to validate the entire flow of design and implementation on the Aquarius II platform. Due to existing bottlenecks in the operating system and the communication between the platforms, we need to develop cores that intensively explore the parallelism in FPGAs. Cores with parallel architectures are already being implemented by the research group and will be prototyped in Aquarius II.

4. Conclusions

In this article was presented Aquarius II, a hybrid platform for dynamically reconfigurable systems prototyping, controlled by the uCLinux operating system. The platform enables algorithms that require acceleration in their implementation use an FPGA as an embedded co-processor. The Aquarius II also implements total and partial reconfiguration of the FPGA. This feature can be exploited in several future applications.

The uCLinux operating system, used in the platform creates an abstraction of how the tasks are running, providing to developers transparency in the use of algorithms implemented in hardware. However, this requires the prior development of a hardware components library that can be used by the developer, similar to the reality present in software libraries.

However, for these libraries provide a significant performance, it is necessary to exploit the parallelism present in the algorithms. The research group is now studying these algorithms.

5. References

[1] W. Rosenstiel, *"Reconfigurable systems – a new era in digital system design has just begun"*, University of Tübingen WSI – Dep. Computer Engineering, 2001.

[2] Patrick Schaumont, Ingrid Verbauwhede, Kurt Keutzer, and Majid Sarrafzadeh, *"A quick safari through the reconfiguration jungle"* in DAC '01: Proceedings of the 38th conference on Design automation, New York, NY, USA, 2001, pp. 172–177, ACM Press.

[3] Ander Dellson, Göran Sandberg, and Stefan Möhl, "Turning FPGAs Into Supercomputers: Debunking the Myths About FPGA-based Software Acceleration," in CUG Proceedings, 2006.

[4] W.H. Mangione-Smith, B. Hutchings, D. Andrews, A. DeHon, C. Ebeling, R. Hartenstein, O. Mencer, J. Morris, K.

Palem, V.K. Prasanna, and H.A.E. Spaanenburg, "Seeking solutions in configurable computing," Computer, vol. 30, pp. 38–43, December 1997.

[5] S. Hauck and A. Agarwal, "Software technologies for reconfigurable systems," Submited to Proceedings of the IEEE, 1997.

[6] "FPGA Acceleration in HPC: A Case Study in Financial Analytics", Hosted in <http://www.xtremedatainc.com/pdf/FPGA_Acceleration_in_HPC.pdf>. Access in 07/26/2007.

[7] "Site of CLC-Bio", www.clcbio.com.

[8] Volodymyr V. Kindratenko, Robert J. Brunner, and Adam D. Myers, "Mitrion-C Application Development on SGI Altix 350/RC100," in Proceedings of IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'07), Napa, California, April 2006.

[9] Jordana L. Seixas, "Aquarius - uma plataforma para desenvolvimento de sistemas digitais dinamicamente reconfiguráveis," M.S. thesis, Federal University of Pernambuco (UFPE), Informatic Center, February 2007.

[10] Xilinx, Virtex II Platform FPGA User Guide, User Guide UG002 (V1.5), December 2002.

[11] "Stratix II - Altera", http://www.altera.com/products/devices/stratix2/st2-index.jsp.

[12] Altera, Nios Development Board Reference Manual, Stratix II Edition, 2005.

[13] Xilinx, Virtex II V2MB1000 - Development Board User's Guide (v.3.0), Memec Design, December 2002.

[14] Altera, "Avalon Bus", www.altera.com/literature/manual/mnl_avalon_spec.pdf.

[15] "uCLinux", www.uclinux.org.

[16] Microtronix, "uCLinux NIOS Port", www.microtronix.com.

[17] Daniel P. (Daniel Pierre) Bovet and Marco Cesati, Understanding the Linux kernel, O'Reilly, third edition, November 2005.