University of Business and Technology in Kosovo
UBT Knowledge Center

UBT International Conference 2018 UBT International Conference

Oct 27th, 9:00 AM - 10:30 AM

FPGA multi-core processors power consumption: soft- core vs. hard-core

Marsida Ibro
Aleksander Moisiu University, marsidaibro@uamd.edu.al

Gerti Kallbaqi
Aleksander Moisiu University, gertikallb@gmail.com

Recommended Citation

Ibro, Marsida and Kallbaqi, Gerti, "FPGA multi-core processors power consumption: soft- core vs. hard-core" (2018). *UBT International Conference*. 151.
https://knowledgecenter.ubt-uni.net/conference/2018/all-events/151

FPGA multi-core processors power consumption: soft-core vs. hard-core

Marsida Ibro[1], Gerti Kallbaqi[2]

[1] "Aleksandër Moisiu" University of Durrës, Albania
marsidaibro@uamd.edu.al

[2] "Aleksandër Moisiu" University of Durrës, Albania
gertikallb@gmail.com

Abstract. Actually multi-core processors designs are limited in power consumption and performance. Consequently, it is not possible to optimize further the performance without increasing power consumption. The main challenge in multi-core processors is the fact that they have heterogeneous hardware components. This article will study different technologies for implementing multi-core processors in FPGA devices. The minimum requirement to ensure low power consumption is parallelism. The purpose of this study is to highlight the latest methodologies used in terms of environment, clock signal, testing, flexibility, cost, availability and power consumption.

Keywords: processor, FPGA, soft-core, hard-core, power consumption.

1. Introduction

Since digital signal processing (DSP) is integrated into many devices, the need arises to realize the most optimal design to meet market demands. Software’s enable design flexibility, allowing continuous changes even after the design is over. Software’s are executed sequentially while hardware allows execution in parallel. Also, creating integrated circuit for specific applications (ASICs) takes a lot of time and after the completion it is impossible to change the design. In this case logic programs come to our aid, which provide a good solution by combining hardware and software.

Signal processors have found many applications because of the short time, low power and low cost development. Due to the requirements for designing DSP systems, logic programs have become very necessary. Due to the development of fabrication technologies, FPGAs feature highly programmable logic (CLB) and have become a platform for a wide range of applications. Processors typically perform arithmetic actions through computer programs, and the idea to carry out these actions through hardware has taken a long time to come to fruition. The FPGA development platforms make possible the best possible combination of both cases.

Configurable hardware’s, like FPGAs, offer very high performance and are therefore much faster than ordinary microprocessors. Software multiprocessor technologies utilize valuable resources on programmable devices. Based on their suitability and ability to support parallelism, they serve as excellent platforms for rapid prototype development and provide ample space for multiprocessor design. Often, these microprocessors can be implemented using FPGAs as they enable reconfiguration whenever new functionality is needed. The number of processors of the signals realized through the software must have a simple architecture that provides good performance mainly for not very critical calculations.

A single-core processor consists of a processor, two or more memory levels, main memory, hard disk, and input/output (I/O). Consequently, using cache memories reduces the Time Access Memory (MAT) resulting in a better performance. According to Moor's law, which was declared in 1965, the number of transistors in a chip would double nearly every year. Moor's law is often cited because it says the performance computer wills double every 18 months. The problem of adding more transistors to a chip in the amount of heat generated that exceeds the rate of advancement of cool technology.

2. Related Work

Many of multi-core platforms have been proposed lately [1]. Also, various multi-core processor architectures with soft-core NIOS II are presented [2]. These platforms are designed to increase GOPS efficiency for Watts. One of the famous performance architectures is GPGPU. Fermi processor is an example. It consists of 512 cores. Such architectures present some boundaries in terms of narrowing access memory as all cores share the same memory and high energy consumption, making them unsuitable in the embedded system [3].

In this paper authors present a reconfigurable parametric architecture. This work mainly focuses on providing a dynamic reconfiguration network by developing a generic connectivity link module. PEs used in the design has limited memory and can only perform specific instructions required for digital signal processing [4]. Power consumption reduction can be done by dynamically empowering idle time [5].

The multi-core accelerator for embedded SoCs, called Platform 2012, is presented in [6]. It is based on multiple processor sets and works in MIMD mode. A batch can hold a number of PE (STxP70-V4 processors) ranging from 1 to 16. P2012 is a Globally Asynchronous Locally Synchronous (GALS) structure of arrays linked through a global asynchronous NoC.

In their article, Martos and Baglivo showed the result of applying the soft-core processor Cortex M0 processor to a low FPGA at the end of Xilinx. The processor was simulated at the test bench and then successfully tested with an LED intervention application. Mondragon and Christman in their paper [7] compared a soft-core processor with a real micro controller. The paper emphasizes the trade that can offer both methodologies. Both methods are compared based on the environment, visibility to internal signal behavior, testability, design flexibility, cost and availability, energy consumption, and so on. Three different control systems apply to FPGAs based on soft-core and hard-core and are compared by Weber and Chin on their paper [8]. Anemologies & As [9] presented an assessment of the design methods and concepts of soft-core processors. A detailed overview of soft Xilinx Micro Kindle software, as well as soft-core applications of fixed fixed processors such as Intel and Pentium Z80. Also discussed are the pros and cons of FPGAs on ASICs. In the white paper by Sandia National Laboratories [10], the author compared three reconfigurable FPGA micro-processor software's, with Leon3.

Miney & Kukenska [11] study the application of soft-core processors in FPGAs and some of the decisions and trade designations to be made during the design process. It looks at the operational performance as well as the power needed to implement the functionality of the design system.

Salem, Othman & Saoud [12] implemented a real-time.The operating system on both hard-core and Soft-core processors and used them to control a DC engine car.

In his paper, Prado [13] presented a comparison of the speed, power, flexibility and cost between a microprocessor and its soft-core version. Soft-core developed by the University of Massachusetts compares with a powerful micro controller PIC16F84. The soft core was found to dominate the microcontroller with a velocity factor of 6.9 and energy consumption with a factor of 28.

3. Multi-core processors architecture

Nowadays, CPUs can be categorized by the number of cores in 3 types: single, multi and many. Based on these, it is expected that the number of cores between multi-core CPUs and many-nuclear CPUs will increase. At present, CPUs with a core can only be found in low power solutions, but also there, the minimum seems to be 2 (one for the tasks and one for the operating system). The main reason was the frequency clock, which made it impossible to get to 10 MHz and up processors. The I7-2600 from

2011 has a 1921 single point in the Passmark thread where the fastest CPU, Intel Core i7-7740X 2017, delivers 2652. The multi-core CPU, with 2 to 8 cores, now are the standard for high performance and fast calculation speed.

ARM came later with double commercial Cortex-A9. While operating systems could plan different tasks in different cores, the software package required years to reach. The program may be delayed because there were still improvements in the CPU by increasing the clock signal (many pipelines). 4-core CPUs arose only a few years ago and 8-core CPUs are still high cost CPU. We can expect the core calculations for computers, laptops, tablets, and smart phones to hold between 4 and 8 cores for years to come. 10-64 CPU Core Virtualization CPUs should address many topics at the same time. CPUs simply had a very high GFLOPS to be competitive in parallel code. That depends on the options that AMD, Intel and IBM see how this will develop exactly.

For example, the low-power multi-core processors, called grid processors perform very well tasks such as video encoding, signal processing, cryptography, and neural networks. To meet the high performance requirements of embedded multimedia applications, integrated systems are integrating multiple processing units. However, they are mainly based on the methodology of custom logic design. A multi-core processor which contains one, two or more processors to improve performance and more efficient processing of multiple tasks is a growing industry trend in core processors. A basic block scheme of a multi-core generic processor is shown in Figure 1.

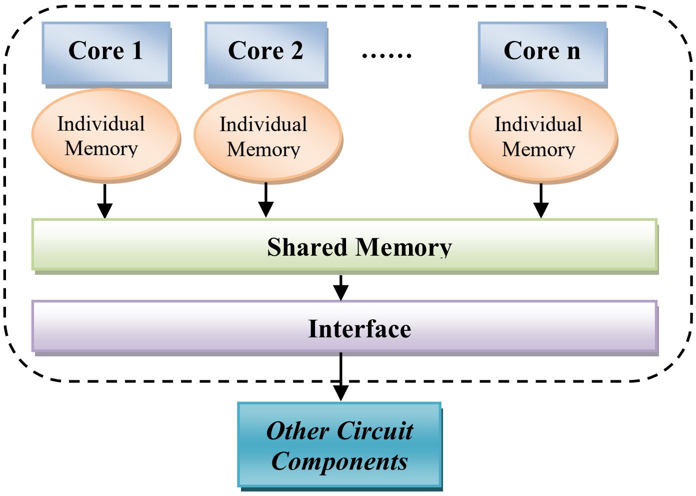
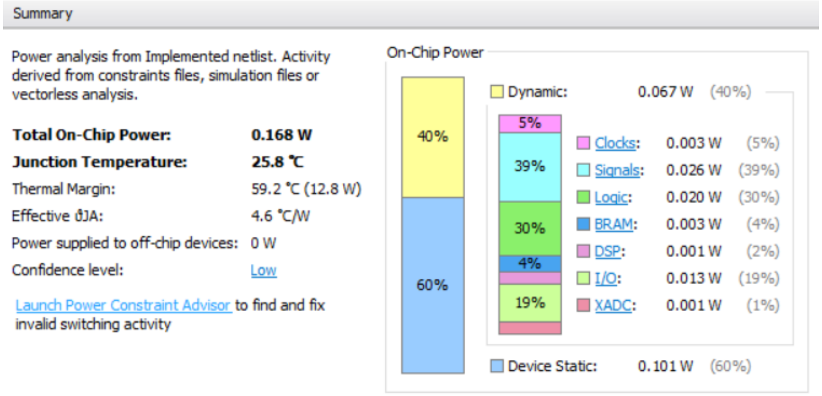


Fig. 1. Block diagram of a multi-core processor

4. Power consumption: hard-core vs. soft-core

The embedded system design is an issue between software and hardware. On one side, hardware is the part that really works, but on the other side, the operating system does functionality and tends to be software, with hardware that supports this effort.. FPGA board contents memories in different size, I/O and other parts of the support devices are provided to address the needs of the functions that need to be executed by the software. Some other IP are used to complete the designs on both soft-core and hard-core based designs. Once the design is completed in HDL, Xilinx Vivado is used to compile and convert the HDL program into a physical design that can be applied to FPGA devices. This includes several steps such as synthesis, simulation, implementation, and finally generation of bits current. Synthesis is the process of transforming a specified RTL design into a port level representation. After their synthesis, modules were simulated individually to verify their functionality. Implementation starts when the simulation is successful.

The flow diagram for the different stages throughout the design process: stage 1 was Hardware Identification, where development boards, soft-core processors, and hard-core processors were deployed. In stage 2, RTL design is done in Verilog/VHDL followed by functional verification by simulation. Phase 3 consists of the application software script (in programming C) that goes into the processor. Stage 4 is where we bring together hardware, software and testing equipment Cortex M0 and Cortex A9 to start testing for the application. Both development boards are tested individually and later side by side to compare performance. At stage 5 we collect data to compare and evaluate two designs. At stage 6, we use the data collected to analyze and find for and against any design (hard-core and soft-core). Xilinx Vivado has the ability to demonstrate design power rating after execution of the implementation phase. This is done on both Soft-core and Hard-core maps and the results are presented in Figure 2 and Figure 3.



Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

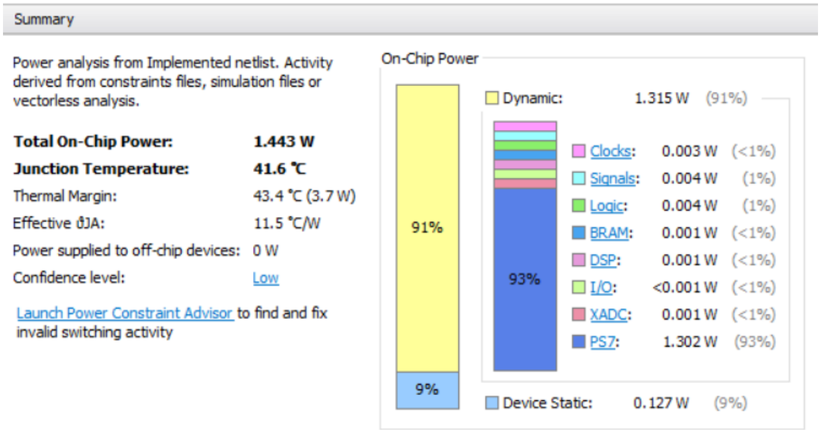
Total On-Chip Power:	**0.168 W**
Junction Temperature:	**25.8 ℃**
Thermal Margin:	59.2 ℃ (12.8 W)
Effective ϑJA:	4.6 ℃/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

□ Dynamic:	0.067 W	(40%)
□ Clocks:	0.003 W	(5%)
□ Signals:	0.026 W	(39%)
□ Logic:	0.020 W	(30%)
□ BRAM:	0.003 W	(4%)
□ DSP:	0.001 W	(2%)
□ I/O:	0.013 W	(19%)
□ XADC:	0.001 W	(1%)
□ Device Static:	0.101 W	(60%)

Fig. 2. Power consumption of soft-core based design



Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**1.443 W**
Junction Temperature:	**41.6 ℃**
Thermal Margin:	43.4 ℃ (3.7 W)
Effective ϑJA:	11.5 ℃/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

■ Dynamic:	1.315 W	(91%)
■ Clocks:	0.003 W	(<1%)
■ Signals:	0.004 W	(1%)
■ Logic:	0.004 W	(1%)
■ BRAM:	0.001 W	(<1%)
■ DSP:	0.001 W	(<1%)
■ I/O:	<0.001 W	(<1%)
■ XADC:	0.001 W	(<1%)
■ PS7:	1.302 W	(93%)
■ Device Static:	0.127 W	(9%)

Fig. 3. Power consumption of hard-core based design

5. Conclusions

In this article, we have introduced a soft-core and hard-core multi-core implementation on the FPGA device and power consumption of both of them tested over a VHDL design. In the future we will also study the effect of the different operating systems. In this work we have compiled an application to compare two types of FPGA embedded processors, such as soft-core and hard-core. Hard-core has exceeded soft-core in both the speed and resource utilization parameter. The hard-core processor is not limited by the FPGA speed as in the case of soft-core.

However, in the case of soft-core power consumption comes because the Cortex M0 in Nexys processor consumes more energy. Thus, we can see that the hard-core processor Cortex A9 is most suitable for applications where speed and resource minimization are an effort, while soft-core processors should be preferred when application flexibility is of major concern.

References

1. T. Dorta, J. Jiménez, J. L. Martín, U. Bidarte, and A. Astarloa, “Reconfigurable multiprocessor systems: a review,” International Journal of Reconfigurable Computing, vol. 2010, Article ID 570279, 11 pages, 2010.
2. A. Kulmala, E. Salminen, and T. D. Hamalainen, “Evaluating large system-on-chip on multi-FPGA platform,” in Proceedings of the International Workshop on Systems, Architectures, Modeling and Simulation (SAMOS '07), S. Vassiliadis, M. Berekovic, and T. D. Hamalainen, Eds., pp. 179–189, Springer, 2007.
3. C. M. Wittenbrink, E. Kilgariff, and A. Prabhu, “Fermi GF100 GPU architecture,” IEEE Micro, vol. 31, no. 2, pp. 50–59, 2011.
4. D. Kissler, F. Hannig, A. Kupriyanov, and J. Teich, “A highly parameterizable parallel processor array architecture,” in Proceedings of the IEEE International Conference on Field Programmable Technology (FPT '06), pp. 105–112, Bangkok, Thailand, December 2006.

5. V. Lari, A. Tanase, F. Hannig, and J. Teich, “Massively parallel processor architectures for resource-aware computing,” in Proceedings of the 1st Workshop on Resource Awareness and Adaptivity in Multi-Core Computing (Racing '14), Paderborn, Germany, May 2014.
6. D. Melpignano, L. Benini, E. Flamand et al., “Platform 2012, a many-core computing accelerator for embedded SoCs: performance evaluation of visual analytics applications,” in Proceedings of the 49th Annual Design Automation Conference (DAC '12), pp. 1137–1142, June 2012.
7. Martos, P., Baglivo, F. (2011). Implementing the Cortex-M0 Design Start processor on a low end FPGA
8. Minev, P. B., & Kukenska, V. S. (2007, November). Implementation of soft-core processors in FPGAs. In UNITECH'07 International Sceintific Conference (2007)
9. Mondragon, A. F., & Christman, J. (2012). Hard Core vs. Soft Core: A Debate. In American Society for Engineering Education.
10. Anemaet, P., & As, T. V. (2003). Microprocessor Soft-Cores: An Evaluation of Design Methods and Concepts on FPGAs. part of the Computer Architecture (Special Topics) course ET4078, Department of Computer Engineering.
11. Learn, M. Evaluation of Soft-Core Processors on a Xilinx Virtex-5. Sandia National Laboratories. SAND2011-2733.
12. Salem, A. K. B., Othman, S. B., & Saoud, S. B. (2008, June). Hard and soft-core implementation of embedded control application using RTOS. In Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on (pp. 1896-1901). IEEE.
13. Prado, D. F. G., (2006, December). Embedded micro-controller and FPGA soft-cores. In ELECTRÓNICA – UNMSM (No. 18). Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, USA.