International Journal of Computer Science, Communication & Information Technology (CSCIT)

Vol.1, pp.14-19, 2014

# Performance Evaluation of FPGA Soft Cores Configurations Case of Xilinx MicroBlaze

Imene MHADHBI[#1], Nejla Rejeb[#2], Slim BEN OTHMEN[#3], Slim BEN SAOUD[#4]

[#]*LSA Laboratory, INSAT-EPT, University of Carthage, TUNISIA*

[1]imene.mhadhbi@gmail.com

[2]rejebnejla2@gmail.com

[3]slim.BenOthman@ept.rnu.tn

[4]slim.bensaoud@gmail.com

***Abstract*— Field Programmable Gate Array (FPGAs) present an attractive choice for implementing embedded systems due to their fast processing speed, their intrinsic parallelism, their rising integration scale and their lower cost solution. The increasing configurable logic capacities of FPGA have enabled designers to handle Microprocessors (MP) soft-cores onto FPGA products. Evaluating the performance of these cores presents the great dial of embedded designers to face up the various problems related to the selection of the efficient FPGA soft-core configuration, against a specific software application. The purpose of this paper was to evaluate the effect of the Xilinx MicroBlaze soft-core configuration on the execution time and the FPGA area consumption using two complementary benchmarks.**

***Keywords*— Embedded Systems, FPGAs, Soft processor, Performance evaluation.**

## I. INTRODUCTION

Nowadays, embedded systems are present in practically all human activities such as cellular telephones, personal digit assistants (PDAs), digital cameras, Global Positioning System (GPS) receivers etc. Semiconductor markets have responded to this demand with a bewildering of other solutions for processing like Application-Specific Integrated Circuits (ASICs), microcontrollers (MCUs), Digital Signal Processors (DSPs) and Field Programmable Gate Array (FPGAs).

MP on FPGA chips are becoming an excellent choice for implementing embedded systems due to their coexistence on-chip with custom logic. This coexistence can reduce costs, board sizes and improve embedded system performance by reducing the communication time between MP and FPGA. In the late 1990s, FPGA vendors began introducing a single-chip MP/FPGA device including hard-core MP (implemented using transistors/gates) and an FPGA on a single Integrated Circuit (IC), with an efficient mechanisms for communication between the MP and the FPGA. Atmel [1] and Triscend [2] were the first to make hard-core MP device available. More recently, Altera [3] offers the Excalibur devices using ARM9 processor and a one million gate FPGA. Xilinx developed the Virtex II Pro device in comporting two or more PowerPC and an FPGA with tens of millions of gates [4].

MPs on FPGAs can be hard-core MP and soft-core MP. While hard-core MP/FPGA offers excellent packaging and communication advantages, soft-core MP/FPGA approach offers the flexibility and the lower cost. Soft-cores MP are synthesized onto the FPGA, like any other circuits. They have the advantage of using lower cost FPGA parts and enabling a custom number of MP per FPGA (over 100 soft-core MP/FPGA can fit onto modern high-end FPGAs).

Many FPGA vendors are offering soft-core MP that designer can implement using a standard FPGA: Altera [3] offers both NIOS and recently NIOS II soft-core, Xilinx offers the PicoBlaze and the MicroBlaze soft-cores [4] , OpenCore offers OpenRISC soft-core [5] and Gailer Research offers LEON and LEON2 soft-cores [6].

The great dial of embedded systems designers are faced up with the various problems in selecting soft-core MP architecture to implement complex applications into the most efficient MP. Performance evaluation of MP help designer to ask the flowing questions: Does a particular MP appropriate for our application? How fast is the used MP? Is it performed for a real-time application? What are the limits to the improvement in this MP? How does the cheeped memory? Etc.

The goal of this paper is to evaluation the performance of the different architectures of the Xilinx MicroBlaze soft-core. The major advantage of choosing MicroBlaze for our research is that we can immediately benefit from the high configurability of this kind of MP. The remaining parts of this paper are organized as follows: Section 2 illustrates the performance evaluation technique of MP. Section 3 presents our design methodology. Section 4 determines the performance evaluation results of the Xilinx MicroBlaze architectures. Finally, Section 5 summarizes the paper and gives our perspectives.

## II. PERFORMANCE EVALUATION TECHNIQUES

The performance evaluation of embedded MP has multiple aspects depending on the application that the system is made to. It will always be a great challenge for designers of this kind of systems, especially for MP/FPGA designers. Hence, performance analysis is involved in several stages of the

design, where going back in the process is very costly. Based on the classification showed in [7], performance evaluation can be classified into two categories: Performance modelling and performance measurements.

### A. Performance modelling

Performance modelling is concerned architecture-under-development. It can be employed at the early stage of the design process where the MP is not available or it is very expensive to prototype all possible MP architectures choices. Performance modelling can be classified into analytical-Based and simulation-Based [7].

1) *Analytical modelling:* is based on probabilistic methods, Marcov models, or Petri nets to create a mathematical model of the system. Results of analytical approach are not often easy to construct. It allows predict mainly user performance, time execution of tasks rapidly without compilation or execution steps. There are not been much study on analytic approach of MP because their structure is more than those analytical models can be provided. However, some researches efforts are present by Nooburg and Shen using Marcov model to model a pipelined processor [8], and by Sorin and al. which used probabilistic techniques to model a Multi-processor composed of superscalar processors [9].

2) *Simulation method:* presents the best performance modelling method in the performance evaluation of MP architectures. The model of the MP being simulated must be written in a high level language such as C or Java and running on some existing machine. Simulators provides performance information in terms of cycles of execution, cache bit ratios, branch prediction rates, etc. Many simulators exist: The SinOS simulator [10] which presents a simple pipeline processor model and an aggressive superscalar processor model and the SIMICS simulator [11] which simulates uni-processor and multi-processor model.

### B. Performance measurements

The performance measurement aims to implement and verify the architectural and the timing behaviours under a set of benchmark programs [7]. The best benchmark presents the application itself. However, in most cases we want a performance estimation of the end product at the initial phase of project. The optimal benchmark program for a specific application is the one who is written in a high-level language, portable across different machines, and easily measurable as well as having a wide distribution.

Several open source and commercial benchmarks are developed. We can find Mibench [13], Paranoia [14], LINPACK [15], SPEC (Standard Performance Evaluation Corporation) [16], EEMBC (Embedded Microprocessor Benchmark Consortium) [17]. Benchmarks can be divided into three categories depending on the application [18]:

3) *Synthetic Benchmarks*: developed to measure processor specific parameters. Synthetic benchmarks are created with the intention to measure one or more features of systems, processors, or compilers. It tries to mimic instruction mixes in real word applications. However, it is not related to how that feature will perform in a real application.

*4) Application Based Benchmarks or "real world" benchmarks*: developed to compare different processors architectures in the same fields of applications. Application based or "real world" benchmarks use the code drawn from real algorithms and they are more common in system-level benchmarking requirements.

5) *Algorithm Based Benchmarks:* (a compromise between the first and the second type) developed to compare systems architectures in special (synthetic) fields of application.

Several studies are based on this approach to evaluate the MP performances. Daniel Mattson and Marcus Christensson evaluated three soft-cores MP/FPGA namely LEON2, MicroBlaze and OpenRISC to measure the execution time and the area consumption, using Dhrystone and Standford benchmarks [7]. Berkly Design Technology Inc evaluated the performance of the Texas Instruments' DSCs processors to compute the execution time using the Fast Fourier Transform (FFT) algorithms using fixed-point and floating-point data precision [17]. Thomas Stolze, Klaus-Dietrich Kramer and Wolfgang Fengler examined the performance of both DSPs and MCUs basing on the execution time of a number of benchmarks codes included fixed-point and floating point math operations, logic calculation, digital control, FFT, conditional jumps and recursion tests algorithms [18].

In our paper, we have chosen to adopt the performance measurements method using freely benchmark solutions. We used the two Synthetic Benchmarks: Dhrystone and Whetstone to compute execution time and area consumption of the different configuration of the soft-core MP/FPGA Xilinx MicroBlaze.

## III. Benchmark Program Selection and Specification

In our work we have chosen to adopt freely available benchmark solutions. We used Synthetic Benchmarks based on the two complementary benchmarks : Dhrystone witch report the integer performance of the architecture in Dhrystone MIPS and Whetstone witch computes different algorithms and report the characteristics of the floating point units in whetstone MIPS.

### A. Dhrystone Benchmark

Dhrystone [19] is a synthetic computation benchmark program developed in 1984 by Reinhold P. Weicker in ADA and translated to C by Rick Richardson. It is intended to be representative of integer performance.

Dhrystone grow to become representative of general processor performance until it was outdated from Standard Performance Evaluation Cooperation. The recent version 2.1 of this benchmark is constituted by 103 high level statements within the main loop, which executes repeatedly during the benchmark execution. User can choose the number of iterations. As result, Dhrystone prints the absolutely time required per iterations through the loop, the performance

measured in number of Dhrystone per second (the number of iterations of the main code loop per second).

### *B. Whetstone Benchmark*

Whetstone benchmark [20] is a synthetic benchmark written in 1972 at the National Physical Laboratory in the United Kingdom. It was the first intentionally written benchmark ware to measure processors performance. It originally measured computing power in units of kilo-Whetstone Instructions per Second (kWIPS). This was later changed to Millions of Whetstone Instructions per Second (MWIPS).

Both Dhrystone and Whetstone are synthetic benchmark, meaning that they are simple programs that are carefully designed to statistically mimic the processor usage of some set of programs. It difficult stems for the fact that one benchmark cannot effectively represent the variety of embedded applications. To evaluate the performance of the Xilinx MicroBlaze soft-core MP configurations, we have to present a brief informal about the MicroBlaze soft-core design methodology.

## IV. MICROBLAZE SOFT-CORE DESIGN METHODOLOGY

### *A. MicroBlaze Soft-core Architecture*

MicroBlaze is a 32-bit Harvard Reduced Instruction Set Computer (RISC) architecture optimized for synthesis and implementation into Xilinx FPGAs with a separate 32-bit instruction and data buses to execute programs and access data from both on-chip and external memory at the same time. Fig. 1 presents a simple MicroBlaze soft-core [21, 7]. It has Harvard memory architecture and uses: Two Local Memory Busses (LMB) for instruction and data memory, two Block RAMs (BRAM) and two peripherals connected via On-chip Peripheral Bas (OPB).

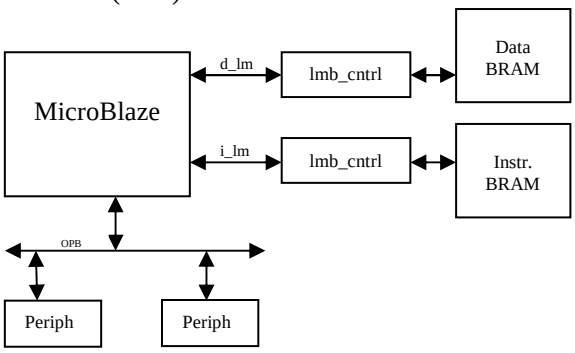


Fig. 1 Simple MicroBlaze soft-core MP/FPGA

The MicroBlaze soft-core offer designer tremendous flexibility during the design process, allowing the designers to configure the MP to meet the needs of their embedded systems with adding custom instructions, Intellectual Proprieties (IPs), particular coprocessor or MP, etc. To alleviate the performance of the MicroBlaze, designer can modify a number of features through the setting parameters. Configured parameters may include:

- Integer Multiplier Units (mul): Integer multiplication.
- Barrel Shifter Units (BS): Shift by bit operations.
- Integer Divider Units (ID): Division of integer numbers.
- Floating Point Units (FPU): Basic and Extended precision.
- Machine Status Register Units (MSRU): Set and clear machine status register.
- Pattern Compare Unit: String and pattern matching.

Performance evaluation was estimated on a first time by a basic measurement of the different MicroBlaze soft-core configurations, on Xilinx Virtex-5 development board (XUPV5-LX110T, xc5vlx110t, grade ff1136, speed -1), in terms of the number of Look Up Tables (LUTs) and Flip Flops (F-Fs) used. We used Xilinx Project Studio (XPS) for configuring the FPGA to include a MicroBlaze soft-core with a 64 KB (the maximum size possible), 125 MHz (the maximum frequency possible). Table 1 presents the area consumption by exhaustively examining some possible MicroBlaze configurations.

TABLE I
THE AREA CONSUMPTION OF SOME MICROBLAZE CONFIGURATIONS.

| **MB + Units** | **With Optimization** | | **Without Optimization** | |
|---|---|---|---|---|
| | LUTs | F-Fs | LUTs | F-Fs |
| Basic | 1210 | 1452 | 1657 | 1693 |
| BS | 1570 | 1247 | 1818 | 1727 |
| FPU | 1620 | 2153 | 2395 | 2105 |
| mul | 1456 | 1232 | 1714 | 1709 |
| ID | 1581 | 1326 | 1801 | 1805 |
| MSRU | 1458 | 1214 | 1675 | 1690 |
| BS+mul | 1571 | 1266 | 1805 | 1748 |
| BS+ID | 1729 | 1361 | 1995 | 1845 |
| BS+mul+ID | 1727 | 1380 | 1964 | 1867 |
| BS+FPU | 1204 | 1655 | 2519 | 2142 |
| BS+mul+FPU | 2307 | 1674 | 2511 | 2162 |
| BS+ID+FPU | 2433 | 1769 | 2668 | 2258 |
| BS+mul+ID+FPU | 2432 | 1788 | 2681 | 2278 |
| BS+ MSRU | 1608 | 1248 | 1829 | 1730 |
| BS+mul+ MSRU | 1609 | 1267 | 1830 | 1749 |
| BS+ID+ MSRU | 1734 | 1365 | 1966 | 1846 |
| BS+mul+ID+ MSRU | 1739 | 1384 | 1967 | 1866 |
| BS+FPU+ MSRU | 2313 | 1659 | 2533 | 2142 |
| BS+mul+ MSRU +FPU | 2355 | 1679 | 2546 | 2165 |
| BS+ID+ MSRU +FPU | 2477 | 1774 | 2680 | 2261 |
| BS+mul+ID+MSRU +FPU | 2479 | 1793 | 2680 | 2281 |
| mul+ID | 1621 | 1347 | 1868 | 1827 |
| mul+FPU | 1495 | 1233 | 1705 | 1712 |
| mul+ID+FPU | 2358 | 1755 | 2575 | 2241 |
| mul+ MSRU | 1504 | 1238 | 1717 | 1711 |
| mul+ID+FPU+MSRU | 2355 | 1679 | 2546 | 2165 |
| mul+ID+MSRU | 1628 | 1351 | 1864 | 1829 |
| mul+FPU+MSRU | 2207 | 1645 | 2418 | 2127 |
| ID+FPU | 2360 | 1736 | 2569 | 2221 |
| ID+ MSRU | 1623 | 1331 | 1883 | 1808 |
| ID+ MSRU+FPU | 2359 | 1740 | 2554 | 2224 |
| MSRU+FPU | 2202 | 1625 | 2417 | 2107 |

Results of the area consumption of the different MicroBlaze soft-core configurations prove that the average slices without using optimization option is very important. For each application, different MicroBlaze configuration is generated and the resulting system can be analysed in different metrics. The use of hardware area presents one of the metric in the choice of embedded systems which requires an optimal area. However, in real-time complex applications, both execution time, area and energy consumption determine the efficiency and the high performance of the configured embedded system. In our paper, we measure the execution time of these configuration using Dhrystone and Whetstone benchmarks using the following design flow.

### *B. Design flow*

To evaluate the performance of the configured MicroBlaze soft-core, the Embedded Development Kit (EDK) is used in our design flow showed in the Fig. 2. EDK enables the integration of both Hardware and Software components of an embedded system. For the hardware side, the architecture is first synthesized into a gate-level netlist, and then translated on the specific device resource such as Look-up tables, flip-flops and block memories.

The interconnections and locations of these resources are, then, placed and routed to meet with the timing constraints. A downloadable .bit file is created for the whole architecture hardware design. For the software side, benchmarks are compiled into an executable and linkable file (ELF) format. The MP software specification (MSS) file and the MP hardware specification (MHS) file are used to define software structure and hardware connection of the embedded architecture. EDK uses these files to implement the design flow and eventually merge the system into a single downloadable file ready to be implemented on FPGA.

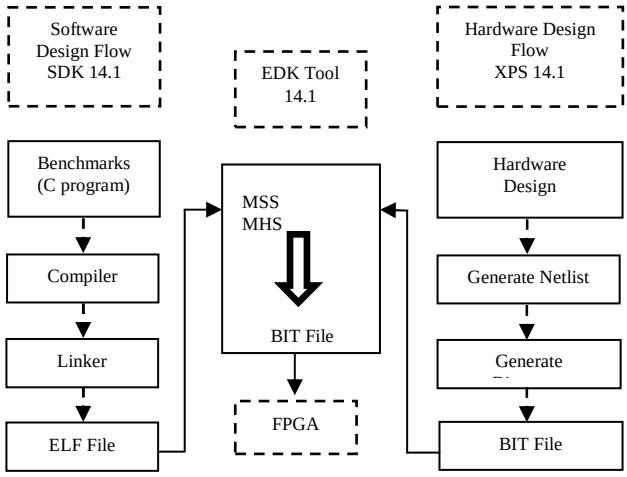


Fig. 2 FPGA Design Flow

## V. PERFORMANCE EVALUATION RESULTS

Processor performance can be measured in many ways. The most common metric is the time required for a processor to accomplish defined task. Some architecture use internal CPU clock driver. The total execution time for the benchmark is the clock driver multiplied by the total instruction cycle count. This clock divided is not reflected in the total instruction cycle count number presented.

All performance measurements are done on a MicroBlaze soft core processor implemented on Virtex 5 Pro FPGA device, presented on the Fig. 1, using Xilinx Embedded Development Kit (EDK).

Fig. 3 Xilinx Virtex-5 FPGA Evaluation Platform

In our case, execution time is measured using a Logic Analyzer to have a high precision measurement. Clearly each benchmark can only be compared to itself, as the resulting values are meaningless outside of that benchmarks context. The executed time for each benchmarks is very small, so a number of loops where used to get a time in microsecond range.

### *A. Dhrystone Benchmark results*

Dhrystone benchmark is used to measure the performance of processors in handling pointers, structures and string. It is dominated by simple integer arithmetic, string operations, logic decisions, and memory accesses intended to reflect the processors activities in most general purpose computing applications. Results of the Dhrystone benchmark are based on the speed time: The number of microseconds that Dhrystone program takes to run. Dhrystone MIPS (DMIPS) and Dhrystone MIPS per MHZ are calculated using the following formulas*:*

$$\mathrm{DMIPS} = (\mathrm{Loop}\ /\mathrm{Run\_Time})/1757)$$

Where:

- 1757 is the number of Dhrystones per Second obtained on the VAX 11/780 (Virtual Address extension), nominally a 1 MIPS machine.

It is interesting to compute the Dhrystone score as a function of the frequency to show the effectiveness of the processor core rather and how fast it can run. DMIPS/MHz is computed using the following formulas.

$$\mathrm{DMIPS/MHZ} = \mathrm{DMIPS/Frequency})$$

Performance was estimated in term of execution time and DMIPS using the different configuration of the MicroBlaze. Results of Dhrystone benchmark are shown in the Table 2.

TABLE II
RESULTS OBTAINED USING DHRYSTONE BENCHMARK USING OPTIMIZED AREA

| **MB + Units** | DMIPS |
|---|---|
| Basic | 0,219 |

| | |
|---|---|
| Barrel Shifter (BS) | 0,220 |
| FPU | 0,211 |
| Multiplier (mul) | 0,237 |
| Integer Divider (ID) | 0,182 |
| MSRU | 0,184 |

Dhrystone benchmark does not use huge values. It is dominated by single integer arithmetic, string operations, logic decisions, and memory accesses intended to reflect the CPU activities in computing applications. Using MicroBlaze soft-core configuration has an effect on the performance of MP. Results prove that MicroBlaze MP is not intended to execute string operations. It takes a huge time to memory access operations. To evaluate the performance of Xilinx MP, we have to estimate the hardware area consumption and the execution time in order to choose the efficient configuration which takes the minimum execution time onto the smaller hardware area. Fig. 3 demonstrates the benefits of the Xilinx MicroBlaze soft-core for the Dhrystone benchmark on 32 configurations.

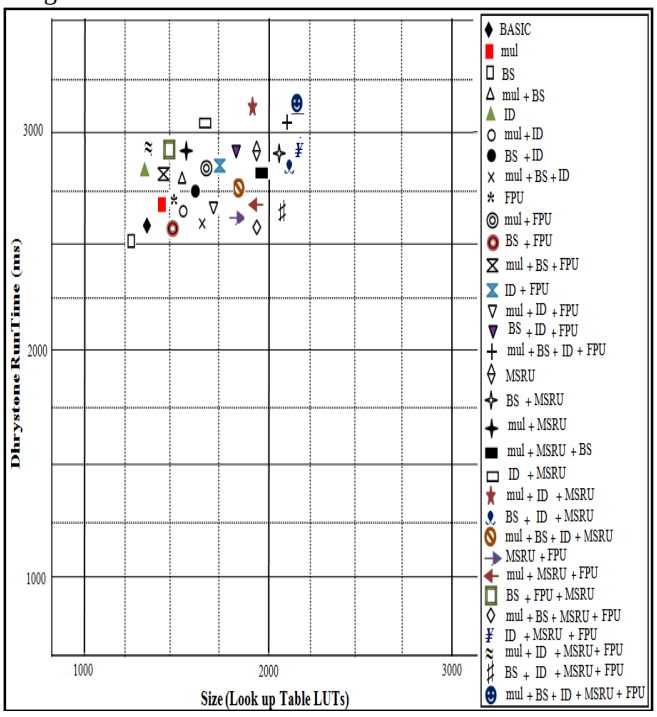


Fig. 4 The Run Time and the size of MicroBlaze soft-core MP configurations executing Dhrystone Benchmark

Figure demonstrates the benefits of configuring soft-core MP to evaluate the embedded MP in terms of execution time and area size constraints. Results prove that the efficient configuration of the MicroBlaze for the Dhrystone benchmark is the using of the multiplier, the barrel shifter and the machine status register units.

### *B. Whetstone Benchmark results*

Whetstone benchmark attempts to measure the performance of both fixed-point and floating-point arithmetic in a variety of scientific functions. These functions are divided into modules: Computation with simple identifiers, computation with array elements, passing an array as parameter, performing conditional Jump, performing integer arithmetic, computation of trigonometric functions, procedures call, array reference and procedure call, integer arithmetic and computations standard functions. The number of Whetstone Instruction per second (WIPS) can be measured for all Whetstone benchmarks. It is calculated as follows:

$$WIPS = (100.0 * Loop)/Run\_Time$$

Performance analysis was estimated in term of execution time and KWIPS using the different configuration of the MicroBlaze soft-core. Table 3 presents the evaluation results of the Xilinx MicroBlaze soft-core MP of the Whetstone benchmark using 6 basics configurations.

TABLE III
RESULTS OBTAINED USING THE TWO DATA PRECISION FOR WHETSTONE BENCHMARK USING OPTIMIZED AREA OPTION

| **MB + Units** | **WIPS** |
|---|---|
| Basic | 323,729 |
| Barrel Shifter (BS) | 471,031 |
| FPU | 324,675 |
| Multiplier (mul) | 502,260 |
| Integer Divider (ID) | 260,416 |
| MSRU | 323,301 |

Results of the Whetstone benchmark prove that the efficient configuration, in term of time execution, is when we used the barrel shifter option. In embedded systems design process, designer have to get idea about the execution time and the hardware area consumption of a specific application, to choose the best configuration. Fig. 4 illustrates the benefits of the Xilinx MicroBlaze soft-core for the Whetstone benchmark on 32 configurations.

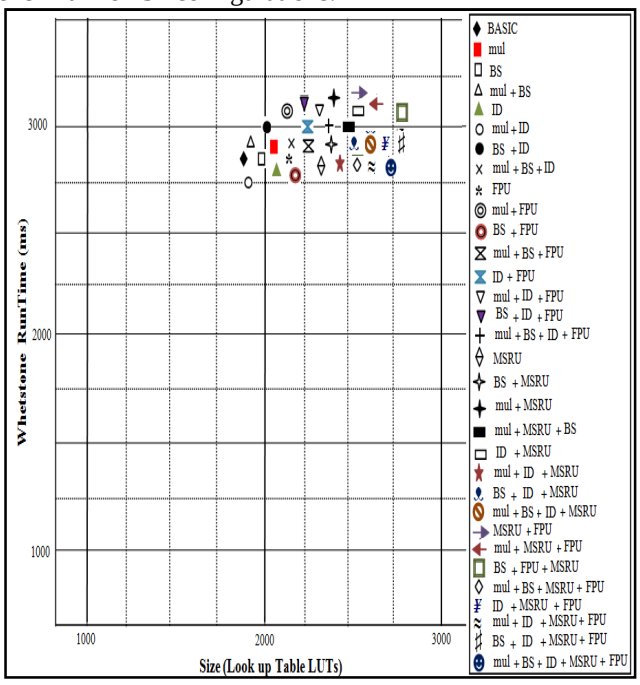

Fig. 5 The run Time and the size of MicroBlaze soft-core MP configurations executing Whetstone Benchmark

Whetstone benchmark is very simple. It contains ten small modules, three executed procedure calls which required the most of the executed time; four loops carry out arithmetic point calculations, two functions, and branching statements. Results demonstrate that the optimal MicroBlaze soft-core configuration for the Whetstone benchmark is the using of the multiplier, barrel shifter, machine status register units and FPU precision.

Performance evaluation of the Xilinx MicroBlaze soft-core MP indicate that customizing soft-core FPGA/MP have an effect of the execution time and the hardware area consumption.

## VI. Conclusions

FPGA based soft-cores present one of the attractive MP for implementing embedded applications. These soft-cores MP/FPGA, such as MicroBlaze, typically require longer execution times with a high energy consumption compared to the hard-core MP/FPGA, which reduce the number of potential application using soft-core MP/FPGA. However, they give designers the flexibility to be configured and enable those designers to quickly build/implement/verify and FPGA systems. The purpose of this paper was to show the effect of the MicroBlaze configuration on the execution time for two complementary benchmarks: Dhrystone benchmark used to compute fixed-point operations and Whetstone benchmark used to characterize the floating-point operations. Results demonstrate that the choice of the good configuration have a significant impact on the system performance. However, obtaining these results require approximately 20 minutes per configuration (40% of the time spent on synthesis). The same approach can be used to evaluate the performance of other embedded systems or other architectures.

The increasing capacity of FPGAs has fuelled their growth in areas of research into soft-core MP architectures. Today's multi-core architectures provide many challenges in the embedded systems area. Multiple MicroBlaze can be implemented on a single FPGA and MicroBlaze Debug Module allows debugging of 8 MicroBlaze MP at a time. The combination of multiple soft-cores MP enables a range of performance optimizing options for parallel processing applications. This work can be extended by evaluating the multi-cores architectures performance.

## References


[1] Atmel Corp. http://www.atmel.com, available at 2012

[2] Triscend Corp. http://www.triscend.com. available at 2012

[3] Altera Corp. http://www.altera.com. available at 2012

[4] Xilinx Inc. http://www.xilinx.com. available at 2012

[5] Daniel Mattsson, Marcus Christensson, "Evaluation of synthesizable CPU, Master ' s Thesis, Computer Science and Engineering Program,. Charmers University of Technology, Departement of Computer Engineering, Ghthenburg 2004.

[6] Xiang LI, Master Thesis, "Open Core Platform based on OpenRISC Processor and DE2-70 Board", Royal Institute of Technology, School of Information and Communication Technology, 2011, Stockholm, Sweden

[7] Lizy Kurian John, "Performance evaluation: Techniques, tools and benchmarks", Electrical and computer engineering department, The university of Texas at Austin

[8] D. B. Noonburg and J.P. Shen, "A framework for statical modeling of superscalar processor performance", the 3rd International Symposium of High Performance Computer Architecture (HPCA), 1997, pp. 298-309.

[9] D. J. Sorin, V. A. Wood, "Analytic evaluation of shared memory systems with ILP processors", The International Symposium on computer architecture, 1998, pp. 380-391.

[10] SimOS complete system simulator, http://simos.stanford.edu, available at 2012.

[11] SIMICS, www.simics.com, available at 2012.

[12] M. R. Guthaus, J. S. Ringenberg and D. Ernst, "MiBench M. R.. A free, commercially representative embedded benchmark suite". In Proc. of 4th Annual IEEE Workshop on Workload Characterization, 2001

[13] R. KARPINSKI, "Paranoia: A floating-point benchmark". Byte Magazine 10, pp. 223–235, 1985

[14] J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart. "LINPACK Users Guide", SIAM Pub, Philadelphia, PA, 1979.

[15] Henning J. L., "SPEC CPU2000: measuring cpu performance in the new millennium, IEEE Computer", vol.7, pp.28–35, July, 2000.

[16] EDN Embedded Microprocessor Benchmark Consortium, available : http://www.eembc.org

[17] Berkeley Design Technology Inc., "Evaluating DSP Processor Performance", 2002.

[18] T. Stolze, Kramer, Fengler: Benchmarking: ClassicDSPs vs. Microcontrollers, Paper IMCIC, 2010, Orlando (Fl.)

[19] G. Stitt, R. Lysecky and F. Vahid, "Dynamic Hardware/Software Partitioning: A First Approach", in Proc. of the 40th Design Automation Conference, pp. 250-255, 2003

[20] H. J. Curnow and B. A. Wichmann, "Whetstone benchmark: A synthetic benchmark", The Computer Journal, vol. 19, pp. 43-49. 1976

[21] Xilinx. Microblaze processor reference guide embedded development kit edk 7.1i. 2012