

Direct FPGA-based Power Profiling for a RISC Processor

Cosmin Cernazanu-Glavan, Marius Marcu,
Alexandru Amaricai, Stefan Fedea, Madalin Ghenea,
Computer and Software Engineering
Politehnica University of Timisoara, Romania

Zheng Wang, Anupam Chattopadhyay,
Jan Weinstock, Rainer Leupers
MPSoC Architecture Research Group
RWTH-Aachen University, 52074 Aachen, Germany

Abstract—This paper investigates the possibility of creating an energy profile of a RISC processor instruction set in the prototyping phase, using FPGA implementation and physical measurements. In order to determine the power consumption at instruction-level, several programs have been developed and run on the processor implementation on FPGA. The experiments have focused at the following groups of instructions: arithmetic and logic (ALU) instructions, memory access instructions, control instructions, compare and move instructions. The main goal of our work is the investigation of the correlation between dynamic power consumption of a RISC processor design implemented in different technologies (FPGA vs. ASIC) and manufacturing processes, called power technology gap. The achieved correlation coefficient between the FPGA 45nm physical power measurements and ASIC 45nm power estimation is 86.39%.

Keywords—power profiling, RISC processor, dynamic power consumption, instruction set, FPGA technology gap

I. INTRODUCTION

Improving the energy efficiency of computing devices is one of the main challenges of our times. For a better evaluation of a computing device in terms of power consumption, instruction-level power consumption analysis has particular importance in the early stages of the design. However, accurate energy profiling can be performed only on a physical prototype, which is available during the final development stages. Instead, model-based energy profiling is usually performed, but it lacks the physical measurements accuracy and calibrations. Reconfigurable devices are being used in a wide range of applications, such as prototyping, digital signal processing, military and space applications, automotive and avionics, telecom, etc. Therefore, reconfigurable devices are good candidates to perform physical energy profiling during prototyping. As power consumption is one of the most important parameters in modern digital systems, evaluating, analyzing and monitoring power and energy consumption of FPGA devices represents an important objective in the design of FPGA based systems.

This paper investigates the possibility of RISC processor power and energy profiling using FPGA prototyping. Instruction level power profiling of a microprocessor is already known for 20 years [1]. The novelty of our work is the investigation of the correlation between dynamic power consumption of a RISC processor design implemented in

different technologies (FPGA vs. ASIC) and manufacturing process (e.g. 90 nm vs. 45 nm). The comparison between FPGA and ASIC is problematic due to the heuristic nature of the design synthesis. On FPGA, a full logical block might be used for one single bit register, which is a single flip-flop in ASIC. However, if the FPGA logical blocks fits the design very well, it translates to more complex combinational logic and flip-flops on an ASIC. Therefore, there is a high degree of uncertainty in terms of area and timing. On the other hand, dynamic power consumption is highly correlated with the gate level switching activity, which is utilized by existing solutions for power estimation models. Therefore, dynamic power consumption correlation between technologies is worth investigating, considering the benefits of its existence. The main benefit is the possibility to achieve physical measurements and study the design behavior using hardware emulated implementations.

A comparison between a 90-nm FPGA and 90-nm standard-cell ASICs in terms of logic density, circuit speed, and absolute power consumption for core logic has been done in [2]. We are correlating the dynamic power consumption measurements of RISC processor instruction set on FPGA board (45nm) with the ones estimated for the same processor implemented for various ASIC technologies (45nm, 65nm, 90nm and 130nm). The main goal of our work is to investigate the possibility of determining parameters of dynamic power consumption for a RISC processor using physical measurements of the same processor implemented on a FPGA board. We are motivated by two advantages: (i) we can estimate dynamic power consumption parameters by physical measurements; (ii) we can validate and thereafter, increasingly refine the power consumption models. In our experiments, we are collecting online physical measurement, using built-in current sensors, performed on Digilent Atlys board, with Xilinx Spartan-6 device. This particular board is needed for physical measurements because the power estimations models that have to be validated, use 45nm technology. Xilinx Spartan-6 device is implemented in 45nm technology while providing built-in current sensors.

This paper is organized as follows: Section II presents other approaches regarding energy evaluation and profiling on FPGA with and without a processor core; Section III is dedicated to the proposed methodology of evaluation and profiling the energy consumption; Section IV described the

performed experiments and correlation results and the last Section is dedicated to the concluding remarks.

II. RELATED WORK

At present, there are many techniques and tools used for monitoring an FPGA. Most of these tools have integrated power monitoring and energy consumption estimation solutions capable of doing most of the work. A very good survey of tools used for energy estimation can be found in [3]. Many of these tools used a soft-processor in order to perform the evaluation for a given design. The estimation is done based on code analysis and other external instruments. LEAP is presented as an FPGA-based hardware profiler being developed as a non-intrusive and area-efficient tool [4]. In order to analyze an executed program, LEAP monitors the instruction bus of the processor together with the program counter. Its objective is to find a perfect correlation between the routines and the counters, based on the starting address for each function.

In [5], the authors developed two profiles for each software application: a power estimation-based profile and a power measurement-based profile. The estimation-based profile is less accurate because it's based on power modeling, but it provides higher flexibility. The other profile, provides extremely accurate results, but requires additional equipment.

A framework made by many abstraction layers is presented in [6]. This framework is useful for the design space exploration process. Inside the framework many units are presented, like: control unit, instruction management unit, data processing unit or memory management unit. Each unit has different metrics and builds a separate power profile.

Another energy estimation method is via instruction level energy profiling [7,8]. There, every instructions' power cost is broken into multiple costs (e.g. base cost, interaction cost, operand costs, etc.). This approach tried to capture all the dependencies between instructions. However, this method is difficult to implement because of the multiple dependencies between instructions (e.g. data dependencies, processing features, cache type, multiplier design, etc.)

Our approach is based on physical power measurement of the FPGA device. A similar approach was used in [9], for several types of embedded systems. However, our method is not restricted on designs which use soft processors and can be performed on any FPGA device. Similar work has been performed on Zynq development board for power control and monitoring [10]. Power monitoring and estimation do not target only processing cores but also memory and interconnect components [11].

III. ENERGY PROFILING METHODOLOGY

Establishing an energy profile for a processor is a difficult task: it requires many physical measurements which must be done with high precision in a strict controlled environment. The profiling test bench should be simple enough so that the power consumption of target component is not masked by other components. The used processor in our tests is a RISC processor provided by Synopsys, connected with two small

instructions and data memories. The PD_RISC core [12] contains 6 pipeline stages with fully bypassed functionality (Fig. 1). It supports mixed 16/32 bits ISA and implements 5 groups of instructions which are ALU, multiply, load/store, compare and branch. The core itself does not contain any caches, which is possible to be integrated through memory interfaces. In this work, we omit caches to prevent prolonged delay of memory access, which may cause variation on accuracy of power measurement. Both program and data memories are implemented using block RAM on FPGA.

The processor instruction set architecture (ISA) was designed using an Architecture Description Language (ADL) - LISA [15]. Using Synopsys Processor Designer technology, software toolchain e.g. assembler, linker are automatically generated. C compiler is also designed using LLVM framework. Arbitrary C code that follows C99 standard is executable on PD_RISC processor. However, due to the simple RISC-style architectural features, embedded applications with small algorithmic kernels are typically executed on this core.

Many FPGA-based development boards provide support for direct measurement of the current consumption by monitoring the voltage drop on very small shunt resistors (milli-ohms) that are placed on the existing power rails powering individual FPGA modules. Accuracy of these measurements may differ from one vendor to another as there are different values used for shunt resistors, as well as for the ADC resolution and sampling rate. The Digilent's Atlys [13] development boards based on Xilinx FPGAs are using INA219 devices [14] for the current monitoring on 3.3V line (which powers FPGA I/O, video, USB ports, ROM), 2.5 V line (FPGA aux, VHDC, GPIO), 1.2V line (FPGA core, Ethernet), 1.8V line (DDR, FPGA DDR, I/O) and 0.9V line (DDR). The built-in sensors provide a 2mA accuracy for the current measurement and 16 Hz sampling rate. This constraint is addressed by the proposed instruction benchmark programs.

In our tests we used Digilent Atlys circuit board based on Xilinx Spartan-6 LX45 FPGA, having the following resources: 6822 slices, each of them containing 4 look-up tables with 6 inputs and 8 flip-flops, 2.1 Mbits of fast block RAM, 4 clock tiles, 6 phase-locked loops, 58 DSP slices and 500 MHz+ clock speeds.

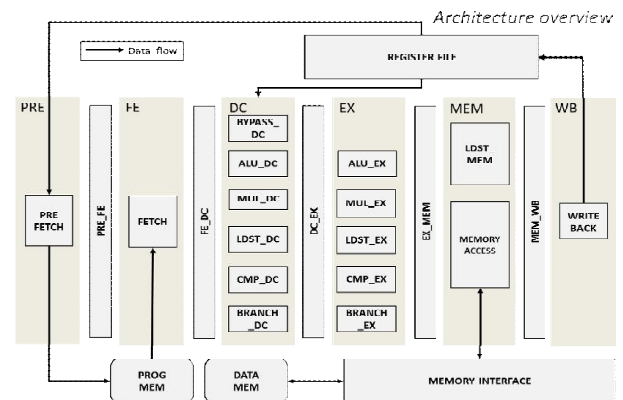


Fig. 1. PD_RISC processor core

The main intended use-case of the proposed measurement infrastructure is to profile the power consumption for each of the processor instruction. Once the power consumption of each instruction is established, we can calculate the total power consumption for a running program.

A. Profile the Power Consumption for an Instruction

For each instruction, a program has been written in order to capture the power consumption variation induced by the instruction. The program consists of 2 main loops: one loop which contains *nop* instruction and one loop which contains the instruction for the profile is made. (Fig. 2)

```

main_loop:
  xor R2,R2
  lw R1,R2
  add R2,1

loop_nop:
  nop
  nop
  nop
  ....
  b loop_nop

loop_instruction:
  add R3,R8
  add R4,R9
  add R5,R10
  add R11,R6
  ....
  b loop_instruction

```

Fig. 2. Program example for *add* instruction

All tests were conducted simultaneously on two identical Digilent Atlys boards. The temperature in the room was maintained at 26 degree Celsius using an internal air conditioning installation. For all test cases we have performed three series of measurements. A series of measurements means that we load into FPGA the PD_RISC processor with a program and, after we have reset the board, we monitor the power lines of the entire board for 5 minutes. After 2 minutes break, we resume the experiment with the same configuration. Thus, for each test we obtained 6 time series. To ensure the validity of the data we made a statistical analysis on all series obtained from one test case. The value of standard deviation we obtained 0.0015.

B. Power Simulation using Standard Cell Library

In order to emphasize the power of our method (based on direct measurements on FPGA) compared with a simulation-based method, we done a simulation of power consumption for the same instruction set based on method described in [12]. The standard-cell based power simulation is performed on the PD_RISC processor from the Synopsys Processor Designer IP libraries. We performed standard cell based power evaluation for following technologies 130nm, 90nm, 65nm (Faraday) and 45nm (Nangate). The architecture is synthesized at 10 MHz using technology libraries by Synopsys Design Compiler. The gate level netlist is simulated by using VCS simulator where the generated VCD file is processed by Synopsys PrimeTime for gate level power simulation.

IV. ENERGY PROFILE FOR PD_RISC PROCESSOR INSTRUCTION SET

The PD_RISC processor instruction set has a number of 71 instructions. From all these instruction we have chosen the most significant instructions grouped on the type of instruction.

A. ALU instructions

The chosen instructions for testing the ALU unit of the processor are: **and16**, **or16**, **and**, **or**, **xor**, **add16**, **add**, **sub** and **mul**. Each of these instructions use the ALU unit of the processor in the own way, but there are many similarities between them. The measured power could vary, but only with minor variations. In time, there are absolutely no variations.

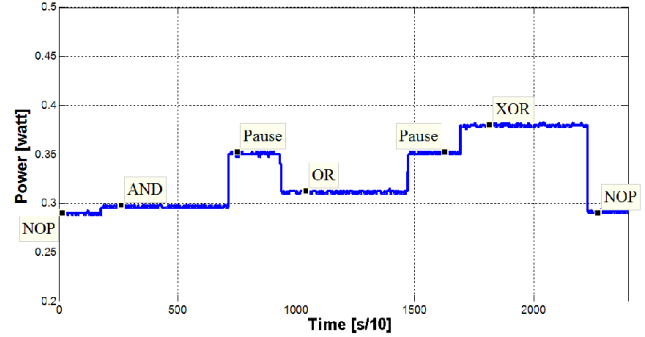


Fig. 3. Power consumption for a program sequence consists of following instructions: **nop**, **and**, **or** and **xor**.

Fig. 3 shows the power variation for a testing program for following instructions: **and**, **or** and **xor**. For each of these instructions we have a large loop filled with only one instruction, followed by a pause loop. These pauses allow us to see the differences between the consumption of each instruction.

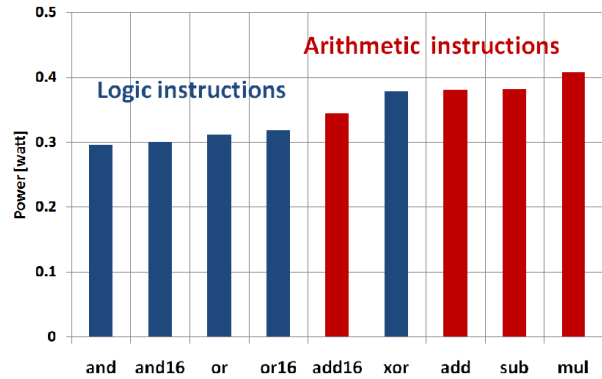


Fig. 4. Power consumption for ALU Instructions

The power consumption for ALU instructions can be seen in Figure 4. The logic instructions have a lower consumption than the arithmetic instructions. The instruction with the most power consumption is the **mul** instruction, and if we consider only the logic instructions we have the **xor** instruction.

B. Memory access instructions

To access the memory, the PD_RISC processor has the following instructions: **lw**, **lh**, **lb**, **sw**, **sh** and **sb**. These instructions permit to store/load information on 32 bits, 16 bits and 8 bits. Consumption for all store instructions doesn't depend on the information stored in memory, but for load instructions, the consumption could be different. Because of that, 3 types of experiments were made, which differ only by the loaded information.

In the first experiment, the information loaded from the memory is changed every time, so that each bit of the registry must switched from 1 to 0 or from 0 to 1 every time (e.g. suppose we load 8 bits of information, one time we load 10101010, the next time we load 01010101 and so on). In the second experiment we loaded only 0 from memory, and in the third experiments we loaded only 1.

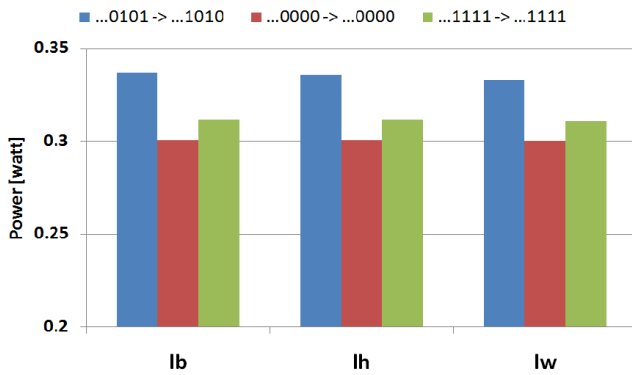


Fig. 5. Differences in power consumption of load instructions due to the loaded information. For 16 bits information we have: (blue) - one time load AAAA, the next time load 5555, and so on. (red) - load only 0000 words from memory. (green) - load only FFFF words from memory

The power consumption for all 3 experiments can be seen in Figure 5. Regardless of the length of the information loaded from memory (8, 16 or 32 bits), the power consumption is higher when the bits of the register must switch when new information is loaded (first experiment).

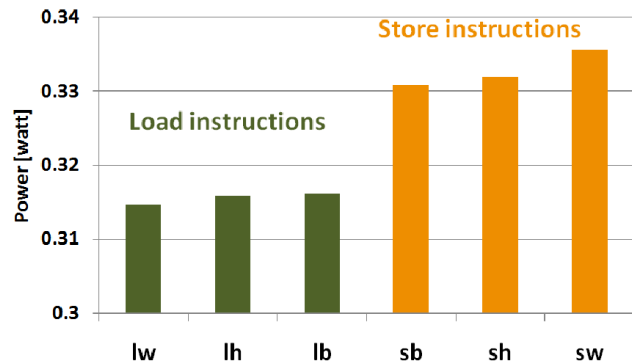


Fig. 6. Power consumption for memory access instructions

To establish a level of consumption for the load instructions, the obtained values for each instruction, in all of the 3 experiments, have been averaged. Putting together the

store and the load instructions it can be easily seen that the level of power consumption for load instruction is lower than the level of power consumption for store instruction (Fig. 6).

In Fig. 6, different modes of load instructions show very close values of power consumption, but small differences can be observed. To investigate on this effect, standard cell based power simulation is performed to check power consumption of individual architectural components. Evaluation shows that the extra power values of lb and lh are consumed by the data masking logic in the MEM pipeline stage. Once the processor loads data word in 32 bits, the lb instruction masks 24 bits data out of 32 bits which consumes extra power. In contrast, the lh instruction masks only 16 bits data, which causes less overhead than lb. The lw instruction does not need any data masking, which consumes no additional power for data loading.

C. Control instructions

A number of 3 instructions existing in PD_RISC processor and used for programming control have been analyzed. These instructions are: **b**, **bcc** and **call**. The difference between **b** and **bcc** instructions is that the **bcc** instruction executes the branch based on a condition.

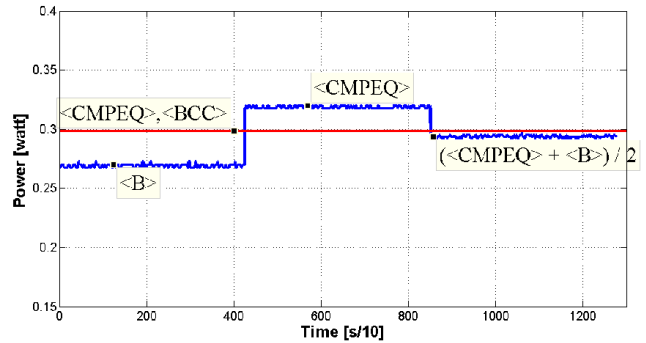


Fig. 7. Power consumption of **cmpeq** followed by **bcc** instruction is approximately equal with power consumption of **cmpeq** followed by **b** instruction

Internally, the **bcc** instruction means an analysis prior of the condition (resulted from another instruction), and after that it could make the jump. We wanted to verify this assumption, so it has been made the following experiment: we measure the power consumptions for **b** (unconditional jump), **bcc** instructions (conditional jump) and for a compare instruction **cmpeq**. The values for the consumption of **b** instruction and for the consumption of **cmpeq** instruction, together with the average of the two values, can be seen in Figure 7 (the blue line). The red line is the consumption of one compare instruction followed by a **bcc** instruction. The red line is approximately equal with the end of blue line (the average value of **b** and **cmpeq**), the little difference represents the power consumption due to the prior analysis (which is part of a **bcc** instruction). This experiment validated the correctness of the measurements and also, gave us important details regarding how instructions are executed internally in the processor.

The last tested control instruction was the call instruction. Like **b**, it is an unconditional branch instruction, but, in addition, the call instruction stores the address of the following instruction into a register.

Fig. 8 shows the consumption power values for all control instruction. As a reference, the power consumption for **nop** instruction was added.

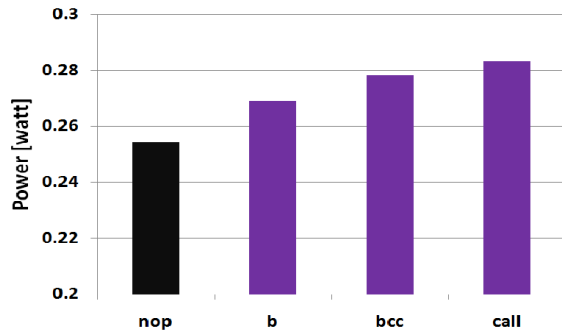


Fig. 8. Power consumption for control instructions

D. Compare and move instructions

The last set of experiments was to determine the power consumption for compare and move instructions. Regarding compare instructions, it have been used instructions which compare register with an immediate signed/unsigned number and instructions which compare 2 registers. The chosen instructions were: **cmpltu**, **cmpeq** (register-register and register-number), **cmplt**, **srl** and **sll**.

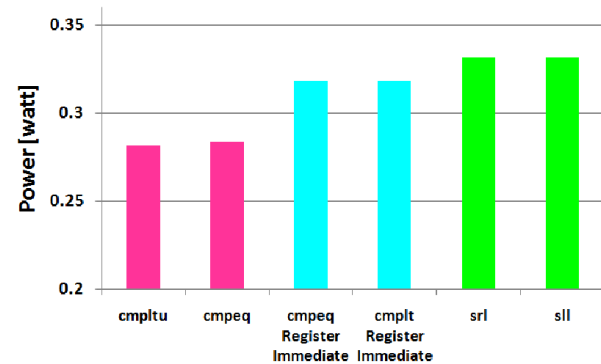


Fig. 9. Power consumption for compare and move instruction

The power consumption for all these instructions can be seen in Fig. 8. It can be observed three levels of power consumption: the lowest level is for compare instructions between 2 register, the medium level is for compare instructions between one register and one number, and the higher level is for move instruction.

E. Overview of power consumption of all tested instructions

The power consumption for all tested instructions can be seen in Fig. 10. The instruction with the most power consumption is **mul** instruction, and with the lowest power consumption is **nop** instruction. Another observation is that

the instructions which are part of a group of instructions (e.g. arithmetic, logic, memory access, etc.) are grouped together in this chart.

For benchmarking purpose with FPGA power measurement, the test benches used on FPGA are simplified for gate-level simulation due to the long simulation time. Typically, each test bench contains one type of instruction with random operands, while the test bench is simulated for only 1,000 clock cycles. Compared to test benches used in FPGA measurement, such timing trade-off leads to inaccuracy in simulated power values, which is caused by the less switching activities in operand values. Fig. 11 shows the estimated power values for instructions using standard cell library. Running at the same frequency, the power values between standard cell and FPGA differ significantly in absolute values due to the higher power efficiency of ASIC library and the different implementation technologies (as discussed also by [8]).

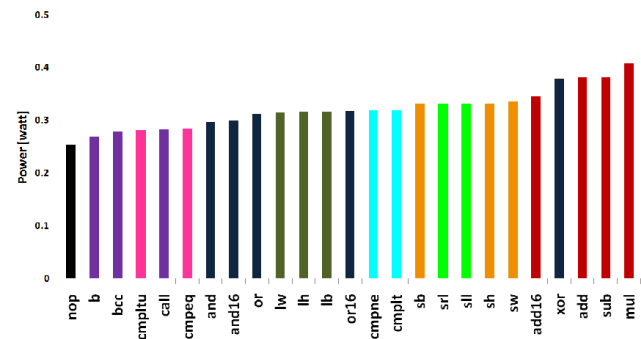


Fig. 10. Power consumption for all tested PD_RISC instructions

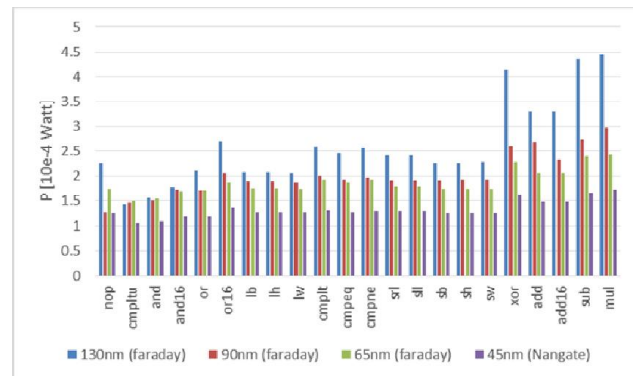


Fig. 11. Simulated Power Values Using Standard Cell Libraries

One observation is that the average power for multiply instruction is significantly less due to large multiplied values are fast out of scale in 32 bits registers, which round to '0' in around 10 loop iterations, for the following million iterations almost no switching activities exist for multiplier itself.

The correlation coefficient between physical measurements of a RISC core implementation on FPGA board (45nm) and simulated ASIC (45nm) power values, is 86.39%. Even though this value shows a strong correlation between the two series, the method based on physical measurements is

recommended because of the reduced execution time and its higher accuracy and validation. The complete correlation matrix of the instruction level power consumption series achieved for different technologies and manufacturing processes is presented in Table I. Very strong correlations between ASIC estimated series are observed, while good correlation between FPGA and ASIC exists too.

TABLE I. POWER CONSUMPTION SERIES CORRELATION MATRIX

	130nm (faraday)	90nm (faraday)	65nm (faraday)	45nm (Nangate)	45nm (Atlys)
130nm (faraday)	1				
90nm (faraday)	92.11%	1			
65nm (faraday)	98.95%	92.53%	1		
45nm (Nangate)	98.67%	93.16%	98.37%	1	
45nm (Atlys)	85.78%	95.47%	84.34%	86.39%	1

Finally, the power technology gap between the RISC core FPGA implementation and ASIC simulations for 45nm manufacturing process is between 200 to 1000 times larger. This is confirmed by the results in [8].

V. CONCLUSIONS

The main purpose of this article has been to measure the power consumption for a RISC processor instruction set. The processor has been implemented on FPGA and different benchmarks programs have been executed. The goal of these measurements has been to investigate the correlation between dynamic power consumption measurements of RISC processor instruction set on FPGA board with the ones estimated for the same processor implemented in a different technology.

The processor instruction set is divided into several groups: arithmetic-logic instructions, memory access instructions, control instructions, compare and move instructions. An interesting observation is that the instructions from a group of instructions have similar power consumption. If we chart the power consumption for all the instructions we can observe that each group of instruction has a certain level for power consumption (distinct from other groups). The instruction with the highest power consumption is **mul** instruction, and with the lowest power consumption is **nop** instruction.

The correlation coefficient between the FPGA 45nm power measurements and ASIC 45nm power estimation is 86.39%. The benefits of this work are two folds. On one hand we can estimate dynamic power consumption by physical measurements, while on the other hand, we can validate power consumption models using real measurements.

As a future work, it is important to determine how the power estimation accuracy scales with increasing program complexity. Furthermore, 28nm manufacturing process implementations on Zynq 7020 evaluation board will be investigated.

ACKNOWLEDGMENT

This work was partially supported by the research grant CHIST-ERA/1/01.10.2012 – “GEMSCCLAIM: GreenEr Mobile Systems by Cross LAYer Integrated energy Management”

REFERENCES

- [1] V. Tiwari, S. Malik, A. Wolfe, “Power analysis of embedded software: a first step towards software power minimization”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 2, 1994.
- [2] I. Kuon and J. Rose, “Measuring the Gap Between FPGAs and ASICs”, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 26, No. 2, 2007.
- [3] Jason G. Tong, Mohammed A.S. Khalid, “Profiling Tools for FPGA-Based Embedded Systems: Survey and Quantitative Comparison”, published in Journal of Computers, Vol.3, No. 6(2008), pp. 1-14, June 2008.
- [4] Mark Aldham, Jason Anderson, Stephen Brown, Andrew Canis, “Low-Cost Hardware Profiling of Run-Time and Energy in FPGA Embedded Processors”, published in 2011 IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP), pp. 61-68, September 2011.
- [5] Andreas Genser, Christian Bachmann, Joseph Haid, Christian Steger, Reinhold Weiss, “An Emulation-Based Real-Time Power Profiling Unit for Embedded Software”, published in International Symposium on Systems, Architectures, Modeling, and Simulation SAMOS '09, pp. 67-73, July 2009.
- [6] Adam Powell, Christos Savvas-Bouganis, Peter Y.K. Cheung, “High-level power and performance estimation of FPGA-based soft processors and its application to design space exploration”, published in Journal of Systems Architecture: the EUROMICRO Journal, Vol. 59, Issue 10, November 2013, pp. 1144-1156.
- [7] Th. Laopoulos, P. Neofotistos, C. Kosmatopoulos, S. Nikolaidis, “Measurement of Current Variations for the Estimation of Software-related Power Consumption”, published in IEEE Transactions on Instrumentation and Measurement, Vol.52, Issue 4, Aug. 2003.
- [8] N. Kavvadias, P. Neofotistos, S. Nikolaidis, K. Kosmatopoulos and Th. Laopoulos “Measurements Analysis of the Software-Related Power Consumption in Microprocessors”, published in IEEE Transactions on Instrumentation and Measurement, Vol. 53, Issue 4, Aug. 2004.
- [9] Z. Nakutis, “Embedded Systems Power Consumption Measurement Methods Overview”, published in MATAVIMAI Academic Journal; Measurements, Vol. 44, Issue 2, pp. 29-35, Kaunas, 2009.
- [10] Arash Farhadi Beldachi, Jose L. Nunez-Yanez, Accurate Power control and monitoring in ZYNQ boards, 24th International Conference on Field Programmable Logic and Applications (FPL), Sep. 2014
- [11] Stefan Schürmans, Diandian Zhang, Dominik Auras, Rainer Leupers, Gerd Ascheid, Creation of ESL Power Models for Communication Architectures using Automatic Calibration, 50th ACM / EDAC / IEEE Design Automation Conference (DAC), Jun. 2013
- [12] Zheng Wang, Lai Wang, Hui Xie, Chattopadhyay Anupam, “Power modeling and estimation during ADL-driven embedded processor design”, Energy Aware Computing Systems and Applications (ICEAC), 2013 4th Annual International Conference on , 16-18 Dec. 2013
- [13] Digilent, “Atlys Board rev. C Reference Manual”, 28 Feb. 2011
- [14] Texas Instruments, “Zero-Drift, Bi-Directional CURRENT/POWER MONITOR with I2C Interface”, Sep. 2011
- [15] Anupam Chattopadhyay, Heinrich Meyr and Rainer Leupers, “LISA: A Uniform ADL for Embedded Processor Modelling, Implementation and Software Toolsuite Generation”, in Processor Description Languages, Morgan Kaufmann, 2008.