*American Journal of Advanced Research*, 2022, 6–2
December. 2022, pages 01-04
doi: 10.5281/zenodo.7265966
http://www.ibii-us.org/Journals/AJAR/
ISBN 2572-8849 (Online), 2572-8830 (Print)

# Performance Comparison of Processor for Energy Efficiency

Muhammad Z Hasan*, Nicholas Trevino, Jacob Flores

School of Engineering, Texas A&M International University, 5201 University Boulevard, Laredo, TX 78041.

*Email: muhammad.hasan@tamiu.edu

Received on June 6, 2022; revised on September 11, 2022; published on October 30, 2022

**Abstract**

Embedded microprocessor systems are used every day by millions of people, but these systems are not seen because (as the name implies) they are buried inside the product or the equipment. They are incorporated into products such as cars, fridges, ovens, traffic lights, industrial equipment, and so on. It was reported that as far back as 1997, close to 2 billion chips were used in numerous embedded systems applications. Embedded processors were expected to grow worldwide by 11 per cent back in 2020.

These embedded processors consume more power than their earlier generations. Focus of this project is to investigate the dynamic power consumption when the processor is running. Two processors are chosen for experiment. The Very Simple Central Processing Unit (VSCPU) has only four instructions whereas the Relatively Simple Central Processing Unit (RSCPU) has sixteen instructions. As such, VSCPU is expected to consume less power. However, for a particular task, the code written for VSCPU is expected to be larger than that of RSCPU. So, the execution time for VSCPU is expected to be longer as well. In this paper, both CPU design is presented and simulation with floating point operations to estimate its power and energy consumption is discussed. The power consumption figures of a Mod 6 counter is presented, as an example. Eventually, the same will be extended to the CPUs and will be compared. This work will demonstrate the effect of instruction-set size on dynamic power and energy consumption of processors.

## 1 Introduction and Problem definition

Embedded microprocessor systems are computer chips that are incorporated into products such as cars, fridges, ovens, traffic lights, industrial equipment, and so on. Although, these embedded systems have shrunk in size and operates at higher speed, they consume more power than their earlier generations [1][2][3]. It is necessary to employ cooling mechanism to disperse the heat generated from this additional power [4][5]. One way to minimize this power consumption is to shut down the internal hardware components that are in idle state at any specific time. They can be powered up again when it is required [6]. Another way to address this problem is to design several versions of the same (homogeneous functionality) component with different power ratings. These versions can be used as needed [7][8][9]. For example, a high power version can be used when high performance is needed and a low power version can be used when the performance demand is less. We have developed methodology to support the above concepts. We have simulated the methodology in appropriate framework and have observed more than 50% reduction in static power consumption (when idle) for two example microprocessors. We would like to investigate the dynamic power consumption when the processor is running. Embedded microprocessor systems are used every day by millions of people, but these systems are not seen because, as the name implies, they are buried inside the product or the equipment. It was reported that as far back as 1997, close to 2 billion chips were used in numerous embedded systems applications. The Very Simple Central Processing Unit (VSCPU) has only four instructions whereas the Relatively Simple Central Processing Unit (RSCPU) has sixteen instructions [10]. However, for a particular task, the code written for VSCPU is expected to be larger than that of RSCPU. As such, the execution time for VSCPU is expected to be longer as well. In this project, RSCPU will be designed and simulated with floating point operations to estimate its power and energy consumption. These figures will be compared with that of VSCPU. When completed, this work will demonstrate the effect of instruction-set size on dynamic power and energy consumption of processors. The first objective is set to measure the dynamic power and energy of RSCPU. The next objective is to compare them with that of VSCPU.

## 2 Hardware Development of VSCPU and RSCPU

Static power consumption is the power used when the processor is idle. Dynamic power consumption is the power used when the processor is executing code. In this project, we would like to investigate dynamic power consumption when the processor runs benchmark code.

As shown in the following Table 1, the VSCPU has only four instructions (ADD, AND, JMP, and INC) implemented through four execution path. In total, it has six states.

Table 1: Execution States of VSCPU

| Fetch 1 | | | |
|---|---|---|---|
| Fetch 2 | | | |
| Fetch 3 | | | |
| (IR=00) | (IR=01) | (IR=10) | (IR=11) |
| ADD 1 | AND 1 | JMP1 | INC1 |
| ADD 2 | AND 2 | | |

These four instructions need to be repeated extensively in order to carry out any complex task. For example, a multiplication will be carried out as repeated addition. As such, the size of the code is expected to be larger (and the execution time will be longer). However, the amount of hardware resources needed for implementing VSCPU is less.

As compared to VSCPU, RSCPU has sixteen instructions as shown in Table 2. Conditional instructions have two additional execution paths. It results in 54 states. These instructions provide more flexibility in order to carry out any complex task and need to be less repeated. As such, the size of the code is expected to be smaller. However, the amount of hardware resources needed for implementing RSCPU is more.

Table 2: Possible Instruction Execution of RSCPU

| Fetch 1 | | | |
|---|---|---|---|
| Fetch 2 | | | |
| Fetch 3 | | | |
| NOP | LDAC | STAC | MVAC |
| MOVR | JMP | JMPZ (Z=1) | JMPZ (Z=0) |
| JPNZ (Z=1) | JPNZ (Z=0) | ADD | SUB |
| INAC | CLAC | AND | OR |
| XOR | NOT | | |

## 3 Implementation Methodology

In order to measure dynamic power, the VSCPU and the RSCPU need to be designed. It was designed using VHSIC Hardware Description Language (VHDL). The language is supported in Quartus II Integrated Development Environment (IDE) tool / platform. IDE with a hardware platform that can accommodate the CPUs is available. The IDE also provides facility to analyze power consumption of the designed processor. Table 3 shows the registers of VSCPU.

| | |
|---|---|
| AR | Address Register |
| PC | Program Counter |
| DR | Data Register |
| AC | Accumulator |
| IR | Instruction Register |

Table 3: Very Simple CPU Registers

In addition, it has a hardwired control unit consisting of a counter, state decoder, and control logic generator. Also the VSCPU has an ALU that can ADD and AND two operands. It was designed and verified on the hardware platform.

Table 4 shows the registers of RSCPU.

| | |
|---|---|
| AR | Address Register |
| PC | Program Counter |
| DR | Data Register |
| TR | Temporary Register |
| AC | Accumulator |
| IR | Instruction Register |
| R | Register |

Table 4: Relatively Simple CPU Registers

It was designed and tested earlier. First, the individual blocks of Table 4 was modeled using VHDL and tested thereupon (as done for VSCPU). In addition, it also has a hardwired control unit consisting of a counter, state decoder, instruction decoder, timing decoder, and control logic generator. Also the RSCPU has an ALU that can perform addition, subtraction, and several logical operations on two operands. Then these models were hierarchically synthesized to form the RSCPU and tested as well. Then the memory was modeled in VHDL with appropriate executable code embedded in the memory. This code represents floating point addition and / or multiplication. These arithmetic operations are considered as benchmarks. RSCPU and the memory will be joined to form the complete system. This complete system will be simulated in Modelsim simulator. The simulator is available from the vendor of Quartus II IDE tool. Both Quartus II IDE and Modelsim simulator run on personal computer.

## 4 Benchmark Application

Benchmark Application consists of floating-point number addition and multiplication. Such 32-bit numbers are typically represented by IEEE 754 format as shown below.


The very simple processor has two separate program instruction configurations set to execute either a simulated IEEE 754 single precision floating point addition operation or multiplication operation depending on which program is loaded to the processor's memory. The IEEE 754 single precision floating point addition operation executes 8 AND, 8 JMP, and 3 ADD instructions to simulate the 8-bit exponent comparison and 23-bit mantissa addition. The simulated IEEE 754 multiplication is accomplished through a series of 16 ADD operations which represent the exponent additions and multiplication of the mantissa bits.

Due to the additional instructions implemented in the Relatively Simple processor, it was possible to more accurately simulate the floating-point operations. The IEEE 754 single precision floating point addition program begins by executing a loop that compares the two exponent values of the number to be added. The first exponent is loaded using the LDAC instruction and is then moved to the storage register R using the MOVR instruction. The second exponent value is then loaded using the LDAC instruction and compared to the exponent value in the R register using the XOR instruction. If the result of the XOR operation is not zero, the program jumps, using the JPNZ instruction, to a set of instructions that first loads the second exponent value from memory using the LDAC instruction. This value is then incremented using the INAC instruction and the new value is stored back into memory using the STAC instruction. After these instructions, the program returns to the beginning instructions using the

JUMP instruction and re-executes the exponent comparison loop. This loop iterates three times before the exponents are equalized. On the fourth iteration of the loop, the exponents are compared and the result of the XOR instruction is zero therefore the JPNZ instruction is not executed. The program then begins adding the mantissa bytes of the floating-point numbers. The first mantissa byte of one number is loaded into the accumulator using the LDAC instruction; it is then moved to the storage register R by the MVAC instruction. The first mantissa byte of the second number is then loaded to the accumulator and added with the first byte of the first number using the ADD instruction. The result is then stored in memory using the STAC instruction. This process is repeated for the remaining two bytes of mantissa data.

The IEEE 754 single precision floating point multiplication routine begins by first loading the exponent byte to the accumulator using the LDAC instruction. This byte is then moved to the storage register R using the MVAC instruction. The exponent of the second number is then loaded to the AC using the LDAC instruction and added to the first exponent using the ADD instruction. The result of this addition is then stored in memory using the STAC instruction. The first mantissa byte of the first number is then loaded to the AC using the LDAC instruction. The ADD instruction is then performed four times to simulate a small multiplication. The result of these additions is then stored to memory using the STAC instruction. This process is then repeated for mantissa bytes two and three of the first number.

## 5 Benchmark Execution and Test Results

Modelsim simulation generates a signal activity file that can is used as an input to Quartus II IDE tool for power calculation. The tool gives the static and dynamic power (p) consumed by the CPU when executing floating point arithmetic code. Also, it is possible to measure the execution time (t) in the simulator. These values would enable us to calculate the energy ($p \times t$) consumed.

Both the CPUs can be operated at various clock frequencies. Three operating clock frequencies will be chosen (10 MHz, 20 MHz, and 40 MHz) and the energy data will be collected for each. These energy data of RSCPU will be compared against the energy data of VSCPU (which was collected in an earlier study) [11].

To familiarize with the process of measuring power, the Mod 6 counter (that counts from 0 to 5) design was chosen. The design worked correctly and was verified by checking the LED counter display with the Altera FPGA board. The modulus 6 counter program, which included the modification of an added clock divider, was downloaded to the FPGA board and compiled in Quartus II for design verification and testing. After the Mod 6 counter was proven to work, the design was opened in ModelSim to check the signal activity during the CPU's operation since the signal activity can be used to analyze the power consumption of the hardware design. The waveform created as a result of the signal activity in ModelSim can be seen in Figure 1.

The signal activity of a Modulus 6 counter generated from ModelSim can be saved to a VCD file and be used to analyze power consumption with Quartus II's Power Analysis Tool. A test bench was used for the Mod 6 counter where design inputs were initialized, and the clock frequencies were also included. Simulation of the test bench generates signal activity, which is fed to Quartus II for power consumption calculation and the results of the power consumption for the Modulus 6 counter can be seen in Table 5 where 5 MHz to 50 MHz clock frequencies were used.

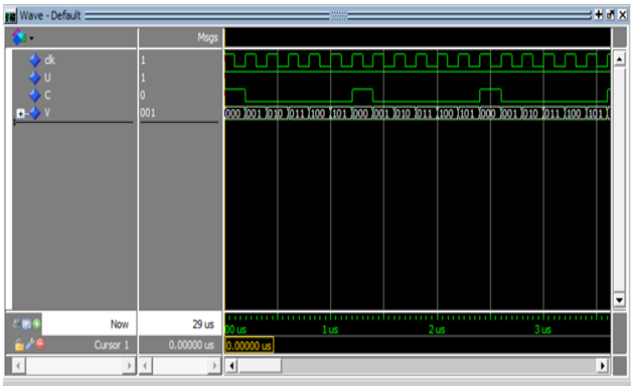


Figure 1: Modulus 6 Counter Input / Output Waveform

| Clock Frequency | Power Consumed |
|---|---|
| 50 MHz | 68.25 mW |
| 40 MHz | 67.53 mW |
| 20 MHz | 66.07 mW |
| 10 MHz | 65.35 mW |
| 5 MHz | 64.96 mW |

Table 5: Modulus 6 Counter Clock Frequency and Power Consumption

The relationship between the clock frequency of the counter and the power consumed is further depicted in Figure 2.

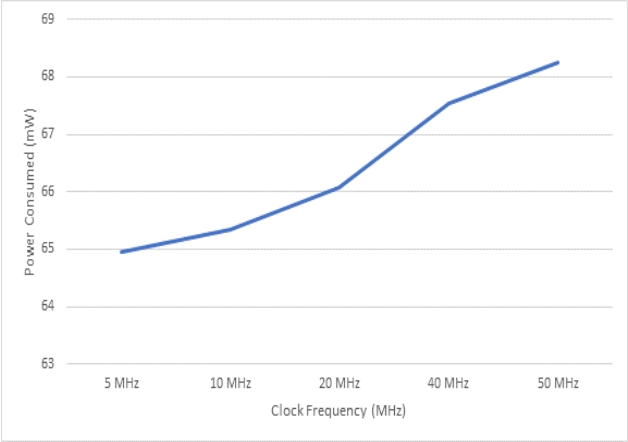


Figure 2: Modulus 6 Counter Clock Frequency and Power Consumption Relationship.

## 6 Conclusion and Further Research

From the results gathered after simulating the Modulus 6 counter, it reveals that almost a linear relationship exists between the performance of the system (in terms of the amount of power the system consumes) and the operating clock frequency. This process of measuring the power consumption would be extended to both VSCPU and RSCPU with benchmark applications. Results would be collected at various frequencies. Then,

those results would be analyzed and compared in order to find the effect of instruction set size on power (and energy) consumption of processors.

## Funding

This work was awarded a University Research Grant by the Texas A&M International University.

*Conflict of Interest:* none declared.

## References

A. Telikepalli, "Power-Performance Inflection at 90 nm Process Node - FPGAs in Focus" Chip Design Magazine, October 9, 2005.

J. Butts, G. Sohi, "A Static Power Model for Architects," *33rd Annual International Symposium on Microarchitecture*, December 2000.

N. Kim, T. Austin, D. Blaauw, T. Mudge *et al*., "Leakage Current: Moore's Law Meets Static Power," *IEEE Press*, December 2003.

B.S. Deepaksubramanyan, A. Nunez, "Analysis of Subthreshold Leakage Reduction in CMOS Digital Circuits," Proceedings of the 13th NASA VLSI Symposium, June 2007.

W. Elgharbawy, M. Bayoumi, "Leakage Sources and Possible Solutions in Nanometer CMOS Technologies," *IEEE Circuits and Systems Magazine*, Fourth Quarter, 2005.

M. Z. Hasan and Mathew Bird, "Energy Reduction for Processors in Reconfigurable Logic," IEEE International Conference on Electro/Information Technology (EIT), Minnesota State University, Mankato, MN, USA, May 15-17, 2011.

P. Abusaidi, M. Klein, B. Philofsky, "Power Consumption Areas in FPGAs," in *Virtex-5 FPGA System Power Design Considerations*, Xilinx Inc. February, 2008.

Texas Instruments Inc, "CMOS Power Consumption and $C_{pd}$ Calculation," Texas Instruments Inc, June 1997.

Umer Misgar and M.Z. Hasan, "Performance Analysis of Different Multiplication Strategies in Reconfigurable Hardware," IASTED International Conference on Parallel and Distributed Systems, Dallas, December 14-16, 2011.

J. Carpinelli, Computer Systems Organization and Architecture, Reading, Massachusetts: Addison-Wesley, October 2000.

Muhammad Z Hasan, Nicholas Trevino, "Performance Comparison of Processor for Energy Efficiency", International Conference of Advanced Research in Applied Science, Engineering and Technology (ICARASET'22), Dallas, TX, April 1, 2022.