

Low Power Microprocessor Design

Sanchith V M, Prashanth K S, Madhvish M Ballal
Department of Electronics and Communication
NIEIT
Mysuru, Karnataka, INDIA

Pawan Bharadwaj
Assistant Professor, Dept. of ECE
NIEIT
Mysuru, Karnataka, INDIA

Abstract— A high performance, low power 32 bit microprocessor is presented in this paper. This processor is based on the RISC architecture which uses an optimized instruction set, large number of general purpose registers and the load/store architecture. The processor uses Harvard architecture which has separate buses for data and code. The design consists of a fast adder module, a fast multiplier module and a fast shifter module, to increase the processor's throughput. The fast adder module is based the look-ahead carry adder. The multiplication module is based on the addshift algorithm which can multiply with a maximum of 32 clock cycles. The shifter is purely based on combinational logic to increase speed. The ALU is made of purely combinational logic which increases the performance. To reduce power consumption, gating clock signals are used. This ensures that the modules which are not in use will not be supplied with the clock signal thereby reducing the power.

Keywords—Low Power; Fast Multiplier; RISC; clock gating;

I. INTRODUCTION

Electronic gadgets have been used since 1900 with the invention of wireless telegraphy and the radio. But their uses have risen meteorically since human kind began using the television, telephones and the smart phones. When these devices were invented, all the different devices had different circuits with discrete parts, but now all of these devices have a microprocessor or a microcontroller built in it. A microprocessor is an electronic device that performs arithmetic and logical computations.

These microprocessors are the brain of every electronic device. They perform all the computing activities. With the astronomical use in electronic devices, the microprocessors have evolved a long way from taking up an entire room and needing hours to perform a computation to now being smaller than the palm and performing millions of computations in a second. The demand for faster processors is still growing rapidly with the advent of smartphones. Statistics show that smartphone users have grown from 76 million to 700 million from 2013 to 2020.

The microprocessors are broadly classified into complex instruction set (CISC), reduced instruction set (RISC), superscalars and digital signal processors (DSPs). CISC processors have thousands of complex instructions whereas RISCs support a smaller set of instructions. Superscalar processors can perform multiple computations at a single

time. The RISC architecture was developed by David Patterson in 1981. This architecture had an optimized set of instructions and a large number of general purpose registers. The ARM processor, used in most smartphones, was initially designed from the RISC architecture.

Some of the popular design methodologies for low power design are power gating, clock gating and voltage reduction. We have used clock gating and improvements in Instruction Set Architecture (ISA) as suggested in [1], to lower the power consumption in our microprocessor.

In this paper, we are designing a RISC microprocessor which is efficient and consumes less power. Fast divider and fast multiplication modules can increase the throughput by decreasing the clock per instruction (CPI). All the modules in a microprocessor run all the time even if not in use, this increases its power consumption. To avoid this, gating clock signals are used, which can decrease power consumption by turning off sub-modules which are not required. This project is built, tested and analysed in Xilinx ISE, the ISim Simulator will be used for testing and the XPower Analyzer is used for the measurement of power consumption and analysis.

II. DESIGN

In this section, the various aspects in the design of the microprocessor are discussed.

A. Instruction Encoding

Instruction encoding is the process of converting the assembly level language instructions into binary in a format that the microprocessor understands. The design has two types of instructions – N-type and I-type. The N-type instructions are for instructions that contain the data in registers. The I-type instructions are for instructions that contain immediate data.

The different fields in the instruction are:

1. Mode – This is a 2-bit field. It selects the Arithmetic and Logic Unit (ALU) or shifter or multiplier/divider. 01 selects the shifter, 10 selects the multiplier/divider and 11 selects the ALU.
2. Function – This is a 14-bit field. This contains the specific function that has to be performed. For example, if the mode is 11 and the function is 14'd0, then it selects the ALU and performs the bitwise AND operation. If the mode is 01 and function is 14'd06, then the shifter is selected and left shift operation is performed.

3. JMP ADDR – This is a 6-bit field containing the address of the jump. If the instruction is not a jump, then it is filled with zeroes.
4. IMM – This is an 8-bit field for immediate data. For load upper immediate and load upper immediate and add instructions, the upper 8-bit data is located in the function.
5. SRC2-sel – This is a 1-bit field used to specify the type of data for source 2. It can specify either register data or immediate data. It will be 0 for register data and 1 for immediate data.
6. SRC2 – This is a 3-bit data field containing the address of the source 2 if the data has to be taken from the register bank.
7. SRC1 – This is also a 3-bit field that contains the address of source 1 register.
8. DST – This is a 3-bit field containing the address of the destination register.

B. High Speed ALU Design

The ALU is a purely combinational circuit to reduce delays caused by sequential circuits. The ALU has 2 inputs, X and Y which are fed into it. There are 8 control signals for the ALU. These are zX , zY , nX , nY , f , L , Cs and $nOut$. According to these control signals, the correct operation is chosen and performed and the result is obtained at ALU out.

The ALU is made of a series of multiplexers with different functionality modules in between. The control signals are fed to each of the multiplexers which select the correct pathway for the data.

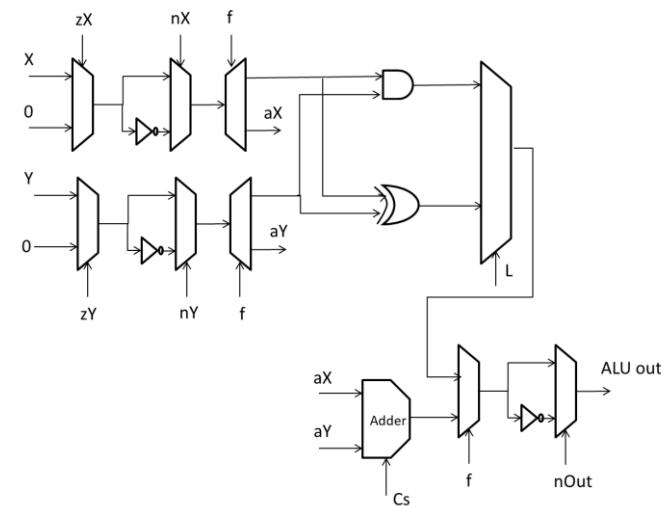


Fig. 1. ALU Circuit

C. Shifter Design

The shifter is a purely combinational circuit which is based on a barrel shifter. Rajalakshmi proposed a universal barrel shifter for low power applications [2]. Barrel shifters are a type of shifter which do not need sequential circuits to operate. The shifter contains 2 stages of multiplexers. In the

1st stage, circularly shifted versions of the input 1 as the actual input and input 2 as select signal, are fed to the multiplexer. A rotated version of input 1 is obtained at the output of the first stage. This is fed to the 2nd stage, where some of the bits are masked to obtain the shifted version of the input. The mask bits are generated in the Zeros module, which is built from truth-table. There are 3 control inputs SRn , LRn and ALn respectively control the shifter to perform either shift/ rotate, left/ right and arithmetic/ logical operation.

D. Independent Multiplier and Divider Module

Independent multiplier and divider module enhances the performance. If both the operations are in the same module, a set number of clock cycles are required to obtain the output. Since our ALU is combinational, all operations other than ALU operations have independent modules. This type was proposed in [3], as the number required clock cycles were large.

E. High Speed Multiplier Design

The paper presents an enhanced version of add/ shift algorithm. In the normal version, if the second input bits are high, shift operation is performed, else the cycle is wasted. In the proposed version, the cycles are not wasted and only the cycles where shift operation needs to be performed are considered. This reduces the time required to obtain the output by a huge margin. This is made possible due to the feedback loop, containing the decoder, NOT and AND gates, and the priority encoder. The number of clock cycles required to obtain the output is variable and depends on the number of 1's present in the second input.

The intermediate values are stored in registers and are updated during every shift operation. The multiplier has a signal (EOP) to inform the CPU of the end of operation.

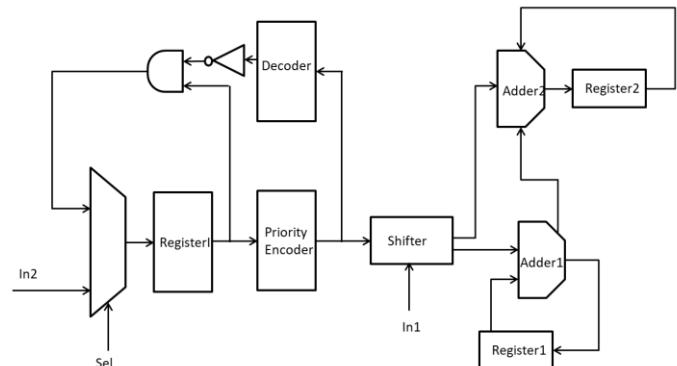


Fig. 2. Enhanced addshift multiplier circuit

Let us take an example where 6 (110 in binary) has to be multiplied by 5 (101 in binary). The number 6 is directly fed to shifter as $In1$. The number 5 is fed to $Register1$ through $In2$, since Sel is HIGH. The priority encoder outputs a value of 2. Hence the shifter left shifts 6 by 2 times and the result

24 is stored in the register. In the next cycle, *Sel* becomes LOW, hence the output of feedback loop i.e. 1 is stored in *Register1*. Now, the output of priority encoder is 0. Hence the 6 at the shifter is not shifted and is added to the intermediate value (24) and the result 30 is stored in the register. Now the EOP signal is made high to indicate that the final result is ready.

F. Clock Gating

Clock gating is a method used to reduce power consumption by not supplying clock signal to a module. In this method control signals are used to gate clock signals, in sequential circuits, to achieve active or power-down states.

In a CPU, clock, registers and memory consume the most power. When implemented on the chip, all circuits transform into an interconnected layout of transistors. The power is consumed, in a transistor, whenever the state of the transistor changes. Since the clock is always active and changing states continuously, stopping the clock reduces power consumption.

In this paper, gating clock is added to multiplier, divider and stack modules. Whenever the instruction requires the use of a module, the module is given clock supply and at other times, the modules are in power-down state.

III. RESULTS

The microprocessor was synthesized and simulated in Xilinx ISE. The Spartan 3 series XC3S400 was chosen as the FPGA board.

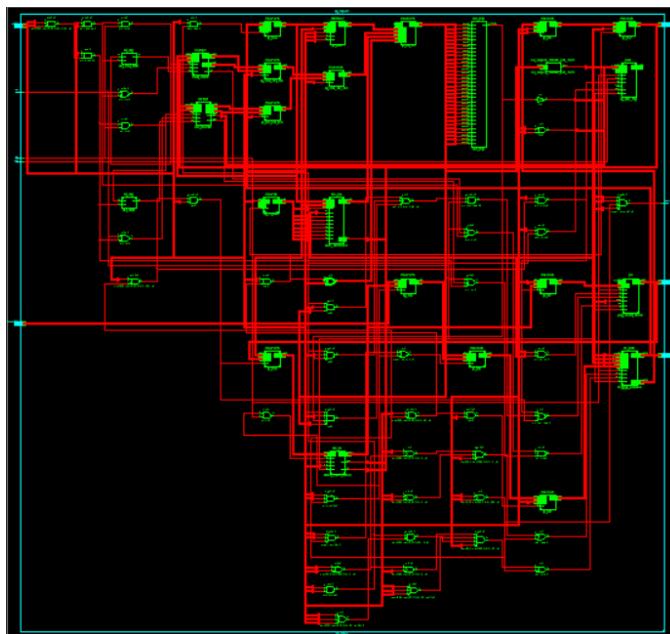


Fig. 3. Schematic Diagram of the processor

TABLE I: RESULTS OF THE SYNTHESIZE PROCESS

Resources	Utilization (in percentage)
Slice LUT	43
Slice Flip-flop	4
Input LUT	42
IO	69
BRAM	18
GCLK	62

Once the design was finished, the individual modules were created and tested. Finally all the modules were integrated and tested for the functionality by writing test benches. The test benches included most of the instructions performed by the processor and verified all the different data paths and control signals. After the simulation was done, the designs files were loaded into Xilinx XPower Analyzer to estimate the power consumed. The results showed that the microprocessor consumes 93mW of power at 24.89 MHz. This result is an estimation of resources of both the microprocessor and RAM unit.

IV. CONCLUSION

The paper presents a 32-bit low power microprocessor. The design uses techniques such as clock gating to reduce power. To reduce the number of clock cycles required, different algorithms were implemented for CPU operations. The processor designed has very low power consumption than that of processors available in the market. The processor has been successfully synthesized and simulated in Xilinx ISE. Test results show that the microprocessor functions correctly and can be applied to the design and development of System-on-Chip (SoC) systems as an IP core and FPGA.

REFERENCES

- [1] T. Hattori, "Design methodology of low-power microprocessors," *Proceedings of the ASP-DAC Asia and South Pacific Design Automation Conference*, 2003., 2003, pp. 390-393, doi: 10.1109/ASPDAC.2003.1195046.
- [2] R. Rajalakshmi and P. Aruna Priya, "Design and analysis of a 4-bit low power universal Barrel-shifter in 16nm FinFET technology," in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, May 2014, pp. 527–532, doi: 10.1109/ICACCT.2014.7019141.
- [3] Hu Yue-li, Cao Jia-lin, Ran Feng, and Liang Zhi-jian, "Design of a high performance microcontroller," in *Proceedings of the Sixth IEEE CPMT Conference on High Density Microsystem Design and Packaging and Component Failure Analysis (HDP '04)*, Jul. 2004, pp. 25–28, doi: 10.1109/HPD.2004.1346667.