International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Review Article

Review Paper on Parallel Processing Single Precision Floating Point Multiplier based RISC Processor

Omkar A. Shastri †*, Shubhangini Ugale† and Vipin Bhure†

†Electronics and Telecommunication, GHRAET, Nagpur, India

Accepted 10 March 2016, Available online 12 March 2016, **Vol.6, No.2 (April 2016)**

Abstract

This paper proposes a 32 bit RISC processor with a parallel processing floating point multiplier for high speed operations. The processor consists of blocks namely, Instruction fetch block, Instruction decode block and the execution block. The execution block will comprise of the parallel processing floating point multiplier so that high speed inputs can be provided thereby improving the accuracy of the system. As the processor is 32 bit, single precision floating point format will be used. Furthermore power gating technique will be used to lower the power consumption of the processor. We use 3 stage pipelining which involves instruction fetch module, instruction decode module and execution module. All the blocks are designed using VHDL hardware description language.

Keywords: *RISC, Floating point multiplier, Power gating, VHDL.*

1. Introduction

John Cocke of IBM Research in Yorktown, New York, originated the RISC concept in 1974 by proving that about 20% of the instructions in a computer did 80% of the work. RISC stands for Reduced Instruction Set Computer. Today RISC is considered to be the basis for designing high performance processors.

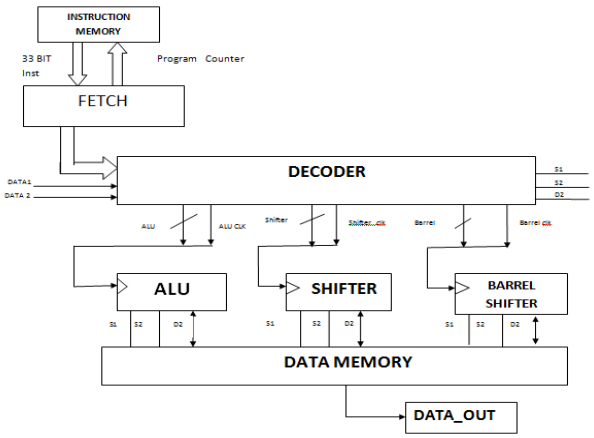

Figure1 Block diagram of RISC processor

The RISC processor have reduced number of Instructions, fixed instruction length, more general purpose register which are organized into register file, load-store architecture and simplified addressing modes which makes individual instruction execute faster, achieve a net gain in performance. The RISC processor requires less number of transistors hence the area required on the chip is less as compared to CISC. Only the load-store instruction access memory, no arithmetic or logic or IO instruction operates directly on memory content which is the key to single clock execution of instructions. It is easier to produce powerful optimized compilers since there are fewer instructions in the instruction set. Figure 1 shows the overview of the RISC processor.

A. Floating point number representation

The concept of floating-point representation over integer fixed-point numbers, which consist purely of significant that expanding it with the exponent component, achieves greater range. The term floating point actually refers to the fact that a radix point of any number, (decimal point, or, more commonly in computers, binary point) can "float"; meaning that it can be laced anywhere in relation to the significant digits of the number. From the past few years a variety of floating-point representations have been utilized in computers, The most commonly referred representation is that which is defined by the IEEE 754 Standard. Floating point numbers are one possible way of representing real numbers in binary format; the IEEE 754 standard presents two different floating point formats, Binary interchange format and Decimal interchange format. Fig. 1 shows the IEEE 754 single precision binary format representation; it consists of a one bit sign (S), an eight bit exponent (E), and a twenty

*Corresponding author: **Omkar A. Shastri**

three bit fraction (M or Mantissa). If the exponent is greater than 0 and smaller than 255, and there is 1 in the MSB of the significant then the number is said to be a normalized number; in this case the real number is represented by (equation 1).

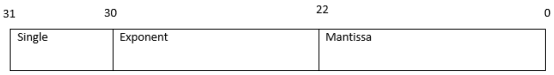

Figure 2 IEEE single precision floating point format

$Z = (-1)^{S} * 2^{(E\text{-}bias)} * (1.M)$ (Eq.1)

Where M= m22 2-1 + m21 2-2 + m20 2-3 +.....+m12-22+ m0 2-23; Bias=127

B. Parallel Processing

Parallel processing and pipelining techniques are duals to each other. If a computation can be pipelined; it can also be processed in parallel. Both of them exploit concurrency available in the computation in different ways. Parallel processing system is also called block processing, and the number of inputs processed in a clock cycle is referred to as the block size. In parallel processing the inputs are processed simultaneously so that the speed of the operation for a system with a high speed inputs can be sufficiently increased further. Parallel Processing requires knowledge of algorithms, languages, software, hardware, performance evaluation and computing alternatives.

2. Literature Review

Mr. S. P. Ritpurkar, Prof. M. N. Thakare, Prof. G. D. Korde used VHDL to describe the system and top-down design method in which initially they designed Instruction Fetch unit, then Decoder unit, then Execution unit, finally memory unit. They analyse Instruction fetch module, Decoder module, Execution module which includes 32bit Floating point ALU, Flag register of 32 bit, MIPS Instruction Set, and 32 bit general purpose registers and design theory based on 32 bit MIPS RISC Processor. Furthermore, they used pipeline concept which involves Instruction Fetch, Instruction Decode, Execution, Memory and Write Back modules of MIPS RISC processor based on 32 bit MIPS Instruction set in a single clock cycle. The pipelined design has been achieved with less clock cycle per instruction and the proposed processor has a delay of 0.741ns and maximum operating frequency of 1.350 GHz (S. P. Ritpurkar *et al*, 2014).

Samiappa Sakthikumaran, S. Salivahanan, V. S. Kanchana Bhaaskaran proposes a 16-bit non-pipelined RISC processor consisting of the blocks, namely, program counter, clock control unit, ALU, IDU and registers. Modifications have been made in the incremental circuits used in the program counter and carry select adder unit in the ALU. A modified low power Wallace tree multiplier has been introduced in the ALU design. They have extended the utility of the RISC processor in the convolution application. The RISC processor has been designed for executing 29 instructions and expandable up to 32 instructions. The processor has been realized using Verilog HDL, simulated using Modelsim 6.2 and synthesized using Synopsys (Samiappa Sakthikumaran *et al*, 2011).

Neenu Joseph, Sabarinath.S, Sankarapandiammal K proposes a low power RISC processor which consumes less power as compared to conventional processor. They have applied a technique in the front hand process to lower the power consumption and have proved that there method is more efficient than the back hand process. The processor is designed using 5 stage pipelining which are fetch, decode, execute, memory, write back. The power is reduced by using the technique called the clock gating technique and is compared with the back end process which resulted in power dissipation in microwatt which is less as compared to conventional one (Neenu Joseph *et al*, 2009).

Pravin S. Mane, Indra Gupta, M. K. Vasantha proposes a 16 bit RISC processor using VHDL. The basic units are modeled using behavioral programming by using the hierarchical approach and are combined using the structural programming. Four stage pipelining consisting of instruction fetch stage, instructions decode stage, execution stage, memory/IO writeback stage is used to improve the overall clock cycles per instruction. The proposed architecture consists of pipelined unit, control unit, hazard detection unit, branch forwarding unit, execution forwarding unit, interrupt and exception unit and pre fetch unit. The maximum throughput is achieved through pipelining and there are no stalls because of the hardwired approach. To resolve the data hazard result forwarding is used while structural hazards are handled using the prefetch unit (Pravin S. Mane *et al*, 2006).

Mrs. Rupali S. Balpande, Mrs. Rashmi S. Keote have proposed an MIPS instruction format, instruction data path decoder module function and design theory based on RISC CPU instruction set. Furthermore, they propose a design of instruction fetch (IF) module of 32-bit CPU based on RISC CPU instruction set. Through analysis of function and theory of RISC CPU instruction decoder module, they also propose a design of instruction decoder (ID) module of 32-bit CPU by pipeline theory. The instruction decoder includes register file, write back data to register file, sign bit extend, relativity check, and it is simulated on QuartusII successfully. They have also design an instruction set, dataflow and pipeline design of RISC CPU based on MIPS. In this research, they have adopted top-down design method and use VHDL to describe system. At first, they design the system from the top, and do in-depth design gradually. The structure and hierarchical of design is very clear. It is easy to edit and debug. Design of instruction fetch (IF) stage simulates, integrate and routes on Quartus II 4.3(Mrs. Rupali S. Balpande *et al*, 2011).

3. Proposed Work

In the existing system, the main issue with the RISC processor is the delay incurred during the multiplication. The multiplication operation in floating point format consumes the most clock cycles in the entire system. This slows down the operation of the system and also is a cause of system errors in case of high speed inputs. This paper proposes a parallel processing based 32 Bit RISC architecture which will reduce the time or clock cycles required to perform the RISC processor operations. The multiplication on the RISC processor will be developed in a parallel processing architecture so that it performs at a faster rate, mostly in 1 clock cycle. Thus faster speed inputs can be provided to the processor thereby improving the accuracy of the system. As this is a parallel processing architecture, the area and the power consumption might increase than the conventional approach. Therefore power gating technique will be used which will help to reduce the power consumed by the processor to some extent.

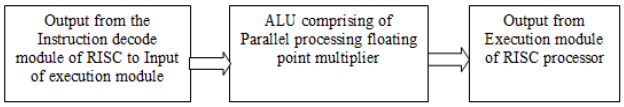

Figure 3 Block Diagram

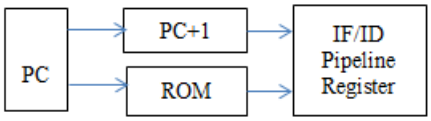

Figure 4 Instruction Fetch Stage

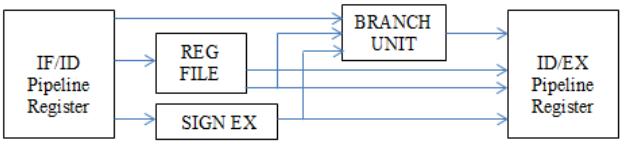

Figure 5 Instruction Decoder Stage

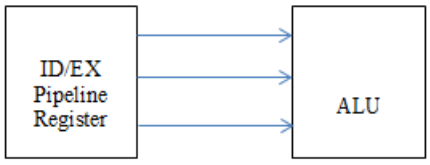

Figure 6 Execution Stage

Conclusion

We will be designing the arithmetic unit of the RISC processor using parallel processing floating point multiplier to increase the speed and accuracy of the system so that high speed inputs can be processed faster. The RISC processor will consist of three modules, namely, Instruction fetch module, instruction decode module, execution module. Furthermore the power gating techniques will be used which may reduce the power consumption by the processor to some extent. All the modules will be designed in Xilinx and Modelsim using VHDL language.

References

S. P. Ritpurkar, Prof. M. N. Thakare, Prof. G. D. Korde (2014), Synthesis and Simulation of a 32 Bit MIPS RISC Processor using VHDL, *IEEE International Conference on Advances in Engineering & Technology Research,* August 01-02, pp.46-50

Samiappa Sakthikumaran1, S. Salivahanan, V. S. Kanchana Bhaaskaran (2011), 16-Bit RISC Processor Design for Convolution Application, *IEEE- International Conference on Recent Trends in Information Technology*, June 3-5, pp.394-397.

Neenu Joseph, Sabarinath.S, Sankarapandiammal K (2009), FPGA based Implementation of High Performance Architectural level Low Power 32-bit RISC Core, *IEEE International Conference on Advances in Recent Technologies in Communication and Computing*, 978-0-7695-3845-7, pp.54-57.

Pravin S. Mane, Indra Gupta, M. K. Vasantha (2006), Implementation of RISC Processor on FPGA, *IEEE,* 1-4244-0726-5/06, pp.2096-2010.

Rupali S. Balpande, Mrs. Rashmi S. Keote (2011), Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor, *International Conference on Communication Systems and Network Technologies,* 978-0-7695-4437-3 DOI 10.1109/CSNT.2011.91, pp.409-413.

Preethi Sudha Gollamudi, M. Kamaraju (2013), Design of High Performance IEEE- 754 Single Precision (32 bit) Floating Point Adder Using VHDL, *International Journal of Engineering Research & Technology*, Vol. 2 Issue 7, pp.2264-2275

Jinde Vijay Kumar, Boya Nagaraju, Chinthakunta Swapna and Thogata Ramanjappa (2014), Design and Development of FPGA Based Low Power Pipelined 64-Bit RISC Processor with Double Precision Floating Point Unit, *International Conference on Communication and Signal Processing*, Vol. 3 Issue 5, pp.1054-1058.

Naresh Grover, M.K.Soni (2014), Design of FPGA based 32-bit Floating Point Arithmetic Unit and verification of its VHDL code using MATLAB, *I.J. Information Engineering and Electronic Business*, 1, pp.1-14

Mohit N. Topiwala, N. Saraswathi (2014), Implementation of a 32-bit MIPS Based RISC Processor using Cadence, *IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT),* ISBN No. 978-1-4799-3914-5/14, pp.979-98

Priyanka Trivedi, Rajan Prasad Tripathi (2015), Design & Analysis of 16 bit RISC Processor using low Power Pipelining, *International Conference on Computing, Communication and Automation (ICCCA),* ISBN: 978-1-4799-8890-7, pp.1294-1297