Prathamesh Berde, Dr.R.D.Daruwala,Sachin kharude / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.2159-2161

Reconfigurable RISC Processor Design and Implementation

Prathamesh Berde[1]*, Dr.R.D.Daruwala,Sachin kharude*****
*(Department of Electrical Engineering,V.J.T.I., Mumbai-19)
** (Department of Electrical Engineering,V.J.T.I., Mumbai-19)
** (Department of Electrical Engineering,V.J.T.I., Mumbai-19)

ABSTRACT
For any computing process we have to perform specific operations in sequence so as to operate and process the data in required sequence. The basic motto here is to design implement & evaluate the reconfigurable processor. It is uses software hardware co-design environment. This reconfigurable processor works similarly like special purpose processors but at much lesser cost than in compared to them as the cost of the processor is important aspect in designing any system. The components are designed in VHDL in top level architecture. It is simulated **and synthesized using Active HDL/Xilinx foundation series.**

Keywords - **VHDL, microprocessor verification, monitor, simulation based, trigger, arithmetic logic unit, address generation logic.**

I. INTRODUCTION

For any computing process we have to perform specific operations in sequence so as to operate and process the data in required sequence. The basic motto here is to design implement & evaluate the reconfigurable processor. It is uses software hardware co-design environment. This reconfigurable processor works similarly like special purpose processors but at much lesser cost than in compared to them as the cost of the processor is important aspect in designing any system. The components are designed in VHDL in top level architecture. It is simulated and synthesized using Active HDL/Xilinx foundation series.

II. CPU (CENTRAL PROCESSING UNIT)

The processor consists of Arithmetic and Logic Unit, First in first out queue, Register array, Multiplier, Shifter, Instruction Controller; it is designed in HDL Language Algorithm for addition multiplication and filtering are also used. Transposed form of FIR filters for cascaded pipelined adders is chosen[6]. The complete design is simulated and synthesized. Prior to synthesis and implementation, control logic is functionally simulated using the Active Xilinx foundation series. We can design the prototype of development board. The flexible JTAG interface has to be designed using Xilinx CPLD[3].This flexibility in JTAG interface can give us basic building platform for dedicated prototype boards within various industries.

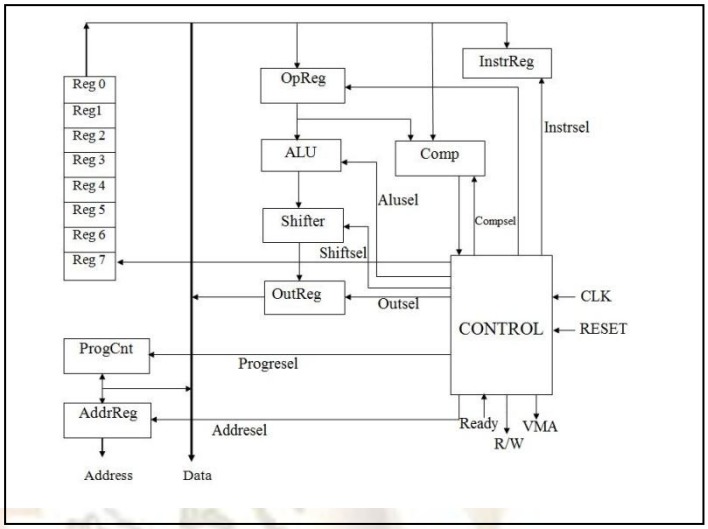

Fig.1 Block Diagram of RISC Processor Architecture

III. FORMAT OF INSTRUCTION

The instructions are executed by processor. It fetches them from memory. Instructions are stored in instruction register and then they are decoded in control unit. The processor consists of eight 16 bit registers ,ALU[1], program counter, instruction register, control unit. They are operated by the use of 32 bit Tristate data bus[4]. The arithmetic operation such as subtraction is executed in following manner. The first register value is written in operational register temporarily. The other register value is stored in data bus.ALU executes subtraction operation and output value is stored in Output register. In the above operation address of current instruction is stored in program counter. The program counter gets incremented and it goes to memory location of the next instruction once the execution of current instruction is completed. When branch instruction is to be executed then despite of the current content of the program counter gets memory location of the instruction to be executed directly. When the data has to be placed on the data bus, the memory sets the READY signal to a 1 value indicating that the memory data is ready for transfer. In branching address register giver new address to address bus. The R/W signal is set low by control unit and if the address is valid then Valid Memory Address is set high. The memory decodes the address and places the data on the data bus. Memory data is written into Instruction register by control unit. And then instruction is decoded and executed by control unit.

IV. PROCESSOR SPECIFICATIONS

The processor has following features:

- 32 bit processor
- 32 bit wide register array
- 32 bit data bus
- 32 bit address bus
- 32 bit ALU
- 32 bit Instruction Register (8 numbers)
- Four Memory modules each of size 16K
- 32 bit First-In-First-Out [FIFO]
- 32 bit multiplier

V. MEMORY UNIT

Memory Module consists of four modules Module 0, Module 1, Module 2 and Module 3.Each module has specific address from address 0 to address 3 to module 0 to module 3 respectively. The memory system is operated using queue. The data transfer is done using handshake signals Request and Grant. The CPU is connected to Memory subsystem via Queue and memory controller. The operation of data like reading and writing of data is done by push and pop operation in the Queue. The memory Sub-system is shown in the Fig.2

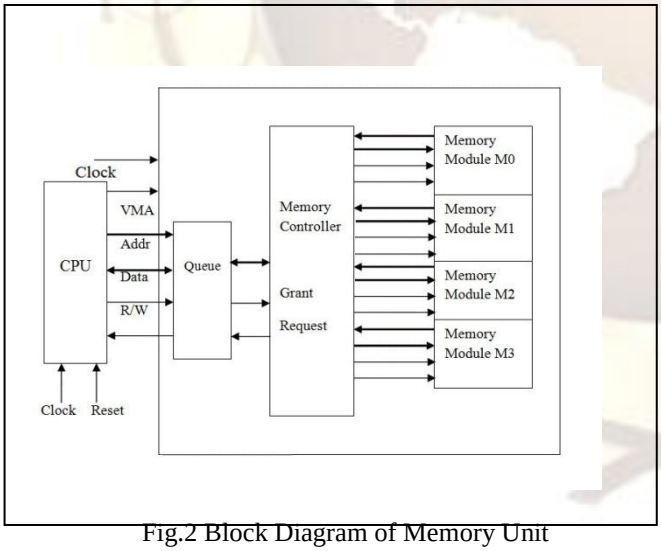

Fig.2 Block Diagram of Memory Unit

VI. FIFO UNIT

If push is active it stores data in the queue and if pop is activated then it gets data from the queue. The block diagram is as shown in Figure 3 and the simulation result of FIFO unit has shown in Figure 4.

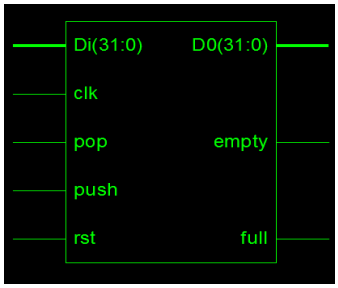

Fig.3 Block Diagram of FIFO

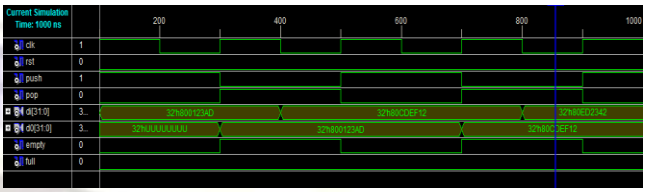

Fig.4 Simulation of FIFO

VII. MULTIPLIER UNIT

Multiplication is an arithmetic operation required in signal processing applications. The multiplier can be designed in various manners. Block diagram of multiplier is as shown in Fig.5 It has two 32 bits data inputs which has to be multiplied. The clock is used for synchronization, reset and start input.

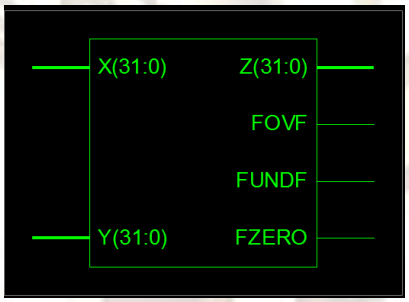

Fig.5 Block Diagram of Multiplier

VIII. SIMULATION RESULTS

Various blocks of the processor are designed using VHDL. Simulation results are shown below.

Fig.6 Block Diagram of Clock signal

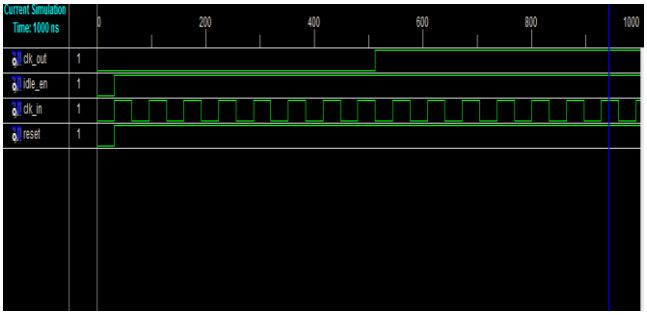

Fig.7 Simulation of Clock signal

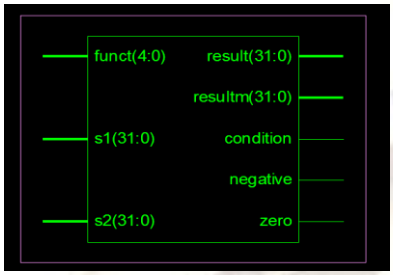

Fig.8 Block Diagram of ALU

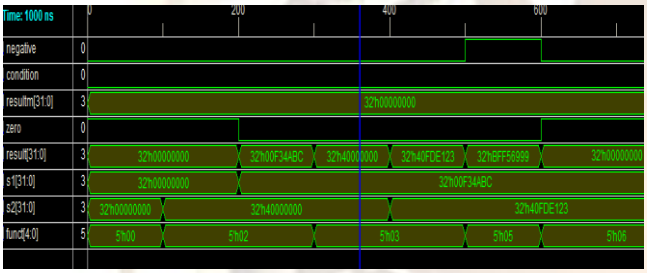

Fig.9 Simulation of ALU

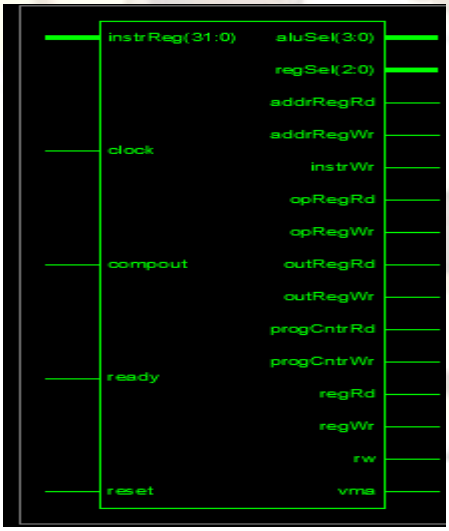

Fig.10 Block Diagram of Control Unit

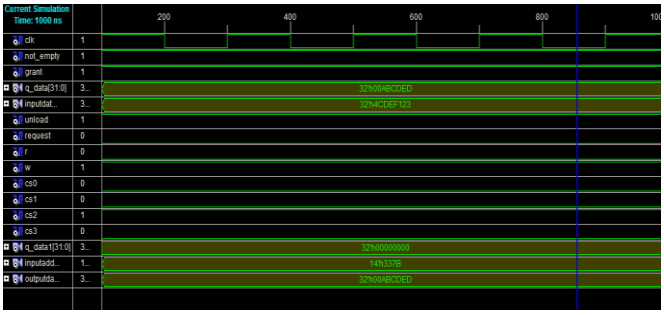

Fig.11Simulation of Control Unit

IX. FUTURE MODIFICATIONS

Reconfiguration provides lot of scope for modifications. Not only instruction set but also complete architecture can be modified. This is the first stage of my project .In next stage of my project we shall design UART and FIR filter which can be used for various applications. In future following modifications are expected in the design.:

1. Cache can be implemented.

2. Test bench for all VHDL codes should be developed.

3. Performance of processor needs to be tested by standard benchmarks like SPEC.

X. CONCLUSION

Since processor is based on reconfigurable programmable logic, user can modify them at any time depending on the need of the application. Simulation results are found to be encouraging and they show that algorithms implemented are working properly. Development board can also be used as a development platform to prototype other HDL designs.

ACKNOWLEDGEMENTS

I want to thank my guide Dr. Daruwala sir for his valuable guidance.I also want to thank my collogue, Sachin Kharude for his help.

REFERENCES

Books:

D. L. Perry, "VHDL: Programming by example", Fourth edition, Tata McGraw Hill, p.p. 289-367

Proceedings Papers:

[1] Kui YI and Yue-Hua DING," 32 bit Multiplication and Division ALU Design Based on RISC Structure", 2009 International Joint Conference on Artificial Intelligence..

[2] Michael Gschwind, "FPGA prototyping of a RISC processor core for embedded applications", *IEEE transaction*, Vol. 9, No. 2, April 2001.

[3] Mamun Bin Ibne Reaz,"A Single Clock Cycle MIPS RISC Processor Design using VHDL" ***ICSE2002*** *Proc. 2002, Penang, Malaysia* Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.