ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

Implementation Of FPGA Based 32 Bit CISC CPU Design

SARASWTHI P [1], M K CHANDRASEN [2]

M.Tech (Student), Department of ECE, Avanthi Institute of Engineering & Tech, Visakhapatnam, India [1]

Assistant Prof, Department of ECE, Avanthi Institute of Engineering & Tech, Visakhapatnam, India [2]

Abstract: Complex Instruction Set Computer (CISC) processors are primarily used in work stations and personal computers. CISC processors with integrated graphics and display systems can be used in car navigation systems, driver information systems etc. It is also applicable in high speed data transmission and real time controlling applications. Taking into consideration all these advantages and applications of CISC processor this work is adopted to design an 8 bit CISC CPU using FPGA.

Keywords: FPGA, CISC, CPU

I. INTRODUCTION

Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors. As of early 2008, billion-transistor processors are commercially available, an example of which is Intel's Montecito Itanium chip.

This is expected to become more commonplace as semiconductor fabrication moves from the current generation of 65 nm processes to the next 45 nm generations (while experiencing new challenges such as increased variation across process corners). Another notable example is NVIDIA's 280 series GPU. While we will concentrate on integrated circuits, the properties of integrated circuits-what we can and cannot efficiently put in an integrated circuit-largely determine the architecture of the entire system. Integrated circuits improve system characteristics in several critical ways.

Electronic systems now perform a wide variety of tasks in daily life. Electronic systems in some cases have replaced mechanisms that operated mechanically, hydraulically, or by other means; electronics are usually smaller, more flexible, and easier to service. In other cases electronic systems have created totally new applications.

II. VHDL

VHDL is an acronym for Very High Speed Integrated Circuits Hardware description Language. The language can be used to model a digital system at many levels of abstraction ranging from the algorithmic level to the gate level. The complexity of the digital system being modeled could vary from that of a simple gate to a complete digital electronic system.

The VHDL language can be regarded as an integrated amalgamation of sequential, concurrent, net list and waveform generation languages and timing specifications.

To meet this challenge, teams of engineers from three companies - IBM, Texas Instruments and Intermetrics — were contracted by the department of defense to complete the specification and implementation of a new language based design description method. The first publicly available version of VHDL, version 7.2 was released in 1985. In 1986, the IEEE was presented with a proposal to standardize the language, which it did in 1987 and academic representatives. The resulting standard, IEEE 1076—1987 is the basis for virtually every simulation and synthesis product sold today. An enhanced and updated version of the language, IEEE 1076-1993, was released in 1994, and VHDL tool vendors have been responding by adding these new language features to their products.

To get around the problem of non-standard data types, an IEEE committee adopted another standard. This standard numbered 1164, defines a standard package (a VHDL feature that allows commonly used declaration to be collected into an external library) containing definition for a standard nine-value data type. This standard data type is called standard logic, and the IELL 1164 package is often referred to as the standard logic package.

The IEEN 1076-1987 and IEEE 1164 standards together form the complete VHDL standard in widest use today (IEEE 1076-1993 is slowly working its way into the VHDL mainstream, but it does not add significant number of features for synthesis users).

In the search for a standard design and documentation tool for the Very High Speed Integrated Circuits (VHSIC) program the United States Department of Defense (DOD) in the summer of 1981 sponsored a workshop on HDLs at Woods Hole, Massachusetts.

The conclusion of the workshop was the need for a standard language, and the features that might be required by such a standard in 1983.DoD established requirements for a standard VHSIC hardware description language(VHDL), based on the recommendation of the "Woods Hole" workshop. A contract for the development of the VHDL language, its environment, and its software was awarded to IBM, Texas instruments and Intermetrics.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

VHDL 2.0 was released only six months after the project began. The language was significantly improved hereafter and other shortcomings were corrected leading to the release of VHDL 6.0. In 1985 this significant developments led to the release of VHDL 6.0. In 1985 these significant development led to the release of VHDL 7.2 language reference manual. This was later on developed as IEEE 1076/A VHDL language reference manual.

III. CISC ARCHITECTURE

Since the development of the stored-program computer, there has been remarkably few true innovations in the areas of computer organization and architecture. In this list, one of the most interesting and potentially one of the most important innovations is Complex Instruction Set Computer (CISC) architecture. The CISC architecture is a dramatic departure from the historical trend in CPU architecture and challenges the conventional wisdom expressed in words and deeds by most computer architects. The aim of the project is to:

- Design the architecture of simple 32- Bit CISC processor .
- Implementation of individual modules of the CISC processor using VHDL .
- Integration of individual modules.
- Their Simulation and Synthesis.

A CISC system offers a large menu of features which implies a larger and more complicated decoding subsystem and complex logic which is not in the case of CISC processors. One of the ways to increase the speed of execution on any computer is to implement pipelining which is difficult to implement in CISC systems. But in CISC processors pipelining can be efficiently implemented.

Due to this opposition to the traditional CISC design, in early eighties there emerged a new trend of computer design, i.e CISC, which became popular thereafter.
Several advantages of CISC processors are:

- VLSI realization
- Increase in computing speed
- Less time to complete the design
- Reduction in overall design cost
- High Level Language (HLL) support

IV. FLOW CHART OF VARIOUS BLOCKS IMPLEMENTATION

The ALU performs both arithmetic and logical operations and as well as control of transfer instructions. It takes data and acc as inputs to generate output according to the opcode. An execlk is given as input for synchronization and the output is available at positive edge of the execlk.

It performs arithmetic and logic instructions directly and control of transfer instructions are performed with the help of control and logic decoder. The block diagram is as mentioned in the Fig.1.

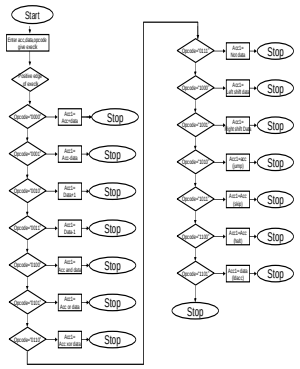


Fig.1. Block Diagram of ALU implementation

The implementation of Accumulator is given as shown in the Fig.2.

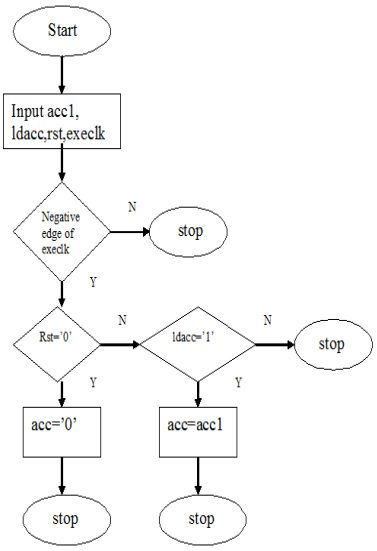


Fig. 2. implementation of ACC

V. TOP ORDER MODULE IMPLEMENTATION

Let us consider an instance when some information is stored in the memory. Now when the system is switched on, CPU is initialized. In order to fetch an instruction, as a result the program goes to the location in the memory that is pointed out by the program counter. After some instance, the instruction from the memory is put on the data bus. This cycle is called the instruction fetch cycle. The instruction is now available at the data bus. at next instance; the instruction is loaded into the instruction register. This is called the instruction load. In this cycle the 4 msb's of the instruction are separated and put in the opcode register and are loaded to control unit as well as ALU. The rest of the bits are sent out as Irout.

The outputs of the instruction register and the program counter are connected to a mux. During the negative edge of the fetch signal, the output of the instruction register is selected, while the output from the program counter is selected during the positive edge of fetch cycle. Now when the fetch signal goes low the mux selects the output from the instruction register and it points to the location of the operand. Now the operand present in the location is placed on the data bus. After an instruction is fetched the

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

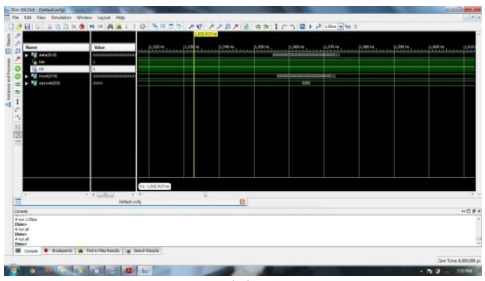
program counter is incremented. It points to the next location. Now the operand is available at the ALU. The operand is taken in by the ALU and operates on it. Now the result is available at acc1 at positive edge of execlk. During the negative edge of execlk, the result at the Acc1 register is placed on the data bus, which is sent and loaded into the accumulator for any further operations. If the data has to be stored into the memory, then during this clock cycle, Rd and Wr has to be 0 & 1 respectively. As a result the accumulator is connected to the memory and the value in the accumulator is sent back to a location in the memory through a module named Buffer.A characteristic of CISC processor is their ability to execute one instruction per clock cycle.

V. RESULTS

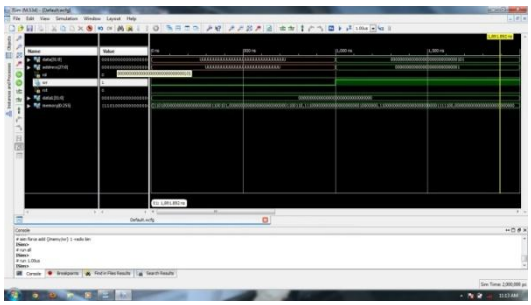
FPGA implementation of the 32 bit CPU involves in implementation of

a) Arthematic Logic Unit
b) Accumulator
c) 32 bit Memory Unit
d) 32 bit MUX
e) Instruction Register
f) Program Counter

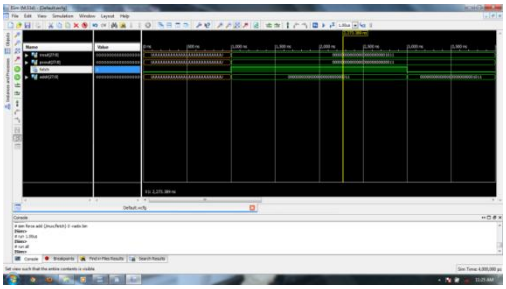
In this results sub section the implementation of some above the blocks are presented for specific data set which is 32 bit. Fig. 3(a)-(d) are the screen shots of the results of implementation of 32 bit Memory unit, MUX, Instruction Register and Program Counter.



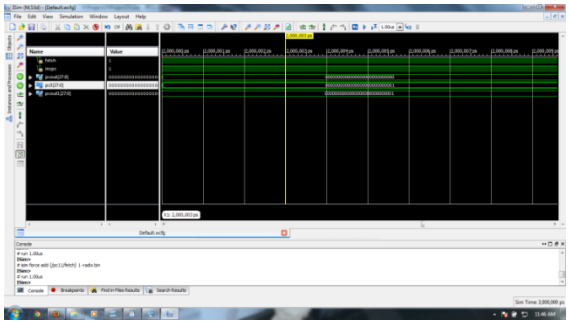
(a)



(b)



(c)



(d)

Fig.3 FPGA implementation of (a) Instruction Register (b)Memory (c) MUX (d) Program Counter

V. CONCLUSION

A 32 bit full CPU implementation using FPGA is performed and the results are presented regarding some the important blocks of the CPU and verified with a 32 bit data.

REFERENCES

[1] Mrs. Rupali S. Balpande, Mrs.Rashmi S. Keote, Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor, 2011 International Conference on Communication Systems and Network Technologies, 978-0-7695-4437-3/11,
[2] 2011 IEEE
[3] Wang-Yuan Zhen, IBM-PC Macro Asm Program, Huazhong University of Science and Technology Press, 1996.9.
[4] MIPS Technologies, Inc. MIPS32™ Architecture For Programmers Volume II: The MIPS32™ Instruction Set June 9, 2003
[5] Zheng-WeiMin, Tang-ZhiZhong. Computer System Structure (The second edition), Tsinghua University Press, 2006
[6] Pan-Song, Huang-JiYe, SOPC Technology Utility Tutorial, Tsinghua University Press, 2006.

BIOGRPHIES

Saraswathi.P is cureently pursuing her Masters in Technology ith specialization in VLSI from Avanthi Institute of Engineering & Technology, Visakhapaatnam. Her research interest include FPGA, VHDL and Verilog Programming.