8 Bit Microprocessor Using VHDL

Pallavi Deshmane[1], Maithili Lad[2] & Pooja Mhetre[3]
Electronics Department
TKIET,Warananagar
[1]pallavideshmane82@gmail.com,
[2]maithililad@yahoo.com

Sharan Kumar
Assistant Professor, Electronics Department
TKIET,Warananagar
kumar.sharan87@gmail.com

***Abstract*— Microprocessors are the heart of any electronic device we use in our everyday life. The efficiency of these processors depend on the speed of execution and simplified and reconfigurable programming. The design of such reconfigurable processors can be done using FPGA, as FPGA devices are reconfigurable, which make them unique than any other type of devices. The paper gives realization of 8 bit FPGA processor. The system was implemented on Xilinx Spartan 3 using ISE and Modelsim simulator. The language used was VHDL. % of slices were used.**

***Keywords: Microprocessor, FPGA, ALU, CPU*.**

I. INTRODUCTION

Nowadays, we seldom see any mobile or smart phone ,any PC or any electrical home appliance without a processor. In fact, it is a heart of any embedded system. A processor mainly works on basis of Central Processing Unit, supported by other discretecomponents like registers, memory, control unit, etc.It is the Central Processing Unit that determines capability of the processor.

The designer of processor has many alternatives like DSPs, ASICs and FPGAs for the designing process. ASICs customized for a particular use are very expensive even though they provide the highest performance. DSP based designs, on the other hand, are cost efficient and low in power consumption and heat-emission. However, they only provide a limited speed for data processing because using special memory architectures that are able to fetch multiple data and/or instructions at the same time, they are susceptible to arithmetic saturation. Whereas FPGAs however have the advantage of shorter time to market, ability to be re-programmed in the field for errors correction, reconfiguration, flexibility, and low-cost. Due to the cost of ASIC design and the speed of DSPs processors that is involved in the development of flexible devices, many engineers are now turning to FPGA. As a result, many significant works have been based on FPGA.

About the hardware Description Language (HDL), VHDL is a general, modern, and powerful language. It is readable, the code developed is portable to any technology at any time, abundance of models available from different sources can be used with ease. VHDL as it is a Platform independent and device independent language. Designers are not put to hard work in changing the tools or environment or devices and hence it has attained a lot of fame among them.

The design of processor in FPGA and VHDL consist of bank of 8 bit general registers, ALU, CU, memory and other special registers.

II. MICROPROCESSOR ARCHITECTURE

The proposed architecture of processor shows its building blocks. The register bank has 7 registers, 7 special purpose registers (each having its own assigned special task), 8 bit ALU and a control unit. All the arithmetic operations are carried out by ALU. The implementation of instruction after they are fetched from memory into instruction register is done by control unit. As the name suggests, control unit controls all the input and output signals, and the steps to be carried out for implementation of any programme loaded in memory.

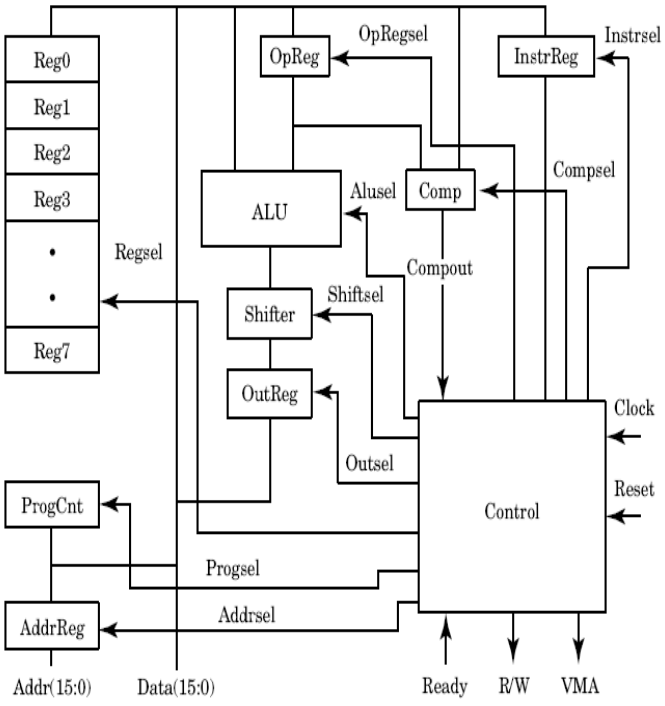


Fig 1. Architecture of designed microprocessor

III. PROCESSOR DESIGN

A) CPU Top-Level Design:

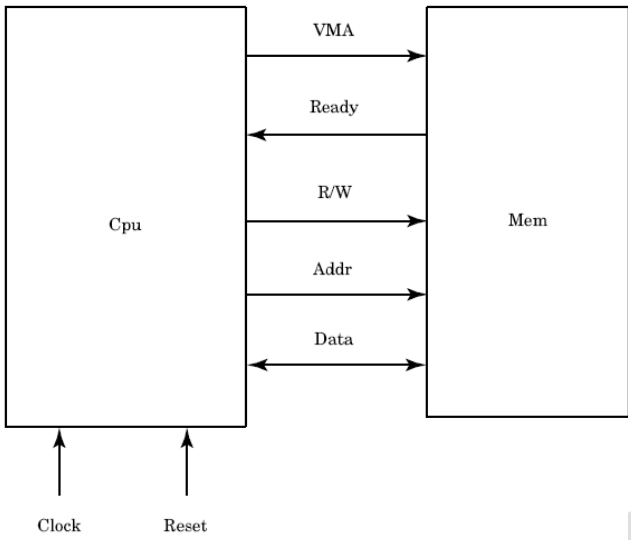


Fig 2. CPU and Memory interface to form top model

"Mem" is a large array with a simple bus interface to allow reading and writing to the memory. A memory location is selected for 'read' by placing the appropriate address of the location on signal "addr", setting input "rw" (read write) to a **'0'** and putting the value '1' on signal "sel" (select). The value of the memory location appears on signal "data", and signal "ready" is set to a '1' value signaling that the memory information is available. To write a location in the memory, the address is placed on signal "addr", set signal "rw" to a '1' value, set signal "sel" to a '1' value, and put the data to be written on input data.

The memory is divided into two separate sections. The first section is the instruction area, and the second is the data area. The instruction area contains the instructions to be executed, and the second section contains the data area for the instructions to manipulate. The CPU instructions start at location **00** and end at location **0D**. The data area starts at location **10** and ends at location **3F**, the end of the array.

B) Control Unit:

control unit provides the necessary signal interactions to make the data flow properly through the CPU and perform the expected functions. Architecture contains a state machine that causes all appropriate signal values to update based on the current state and input signals and produce a next state for the state machine. The control block provides all of the control signals to regulate data traffic for the CPU.

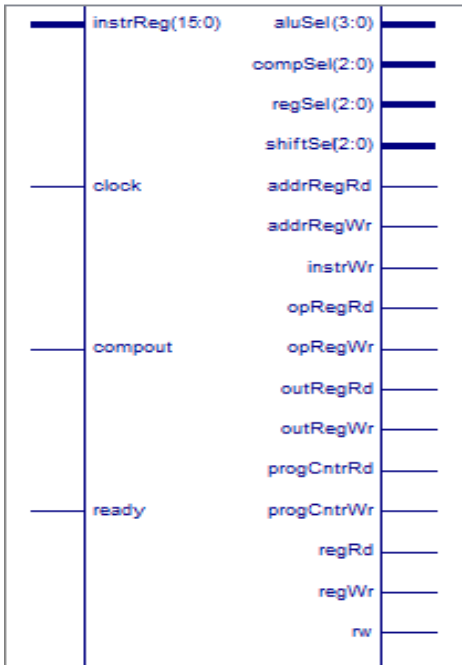
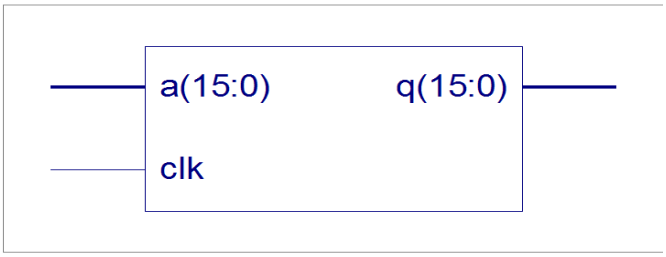


Fig 3. RTL schematic of control unit

Register:

The **register** is used for the address register and the instruction register. These registers need to be able to capture the input data on a rising edge of the **clk** input and drive output **q** with the captured data. The value of input **a** is assigned to output **q** when a rising edge occurs on input **clk**. The assignment is delayed by 1 nanosecond to remove delta delay problems during simulation. A symbol for the **reg** entity is shown in Figure.



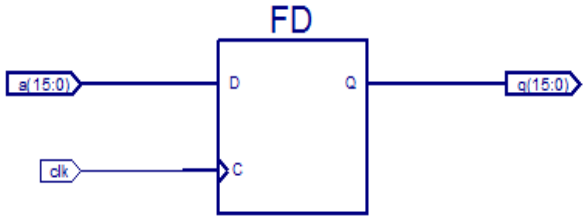


Fig 4. RTL schematic of register

Shift:

The **shift** entity is used to perform shifting and rotation operations within the CPU. The **shift** entity has a 8-bit input bus, a 8-bit output bus, and a **sel** input that determines which shift operation to perform.

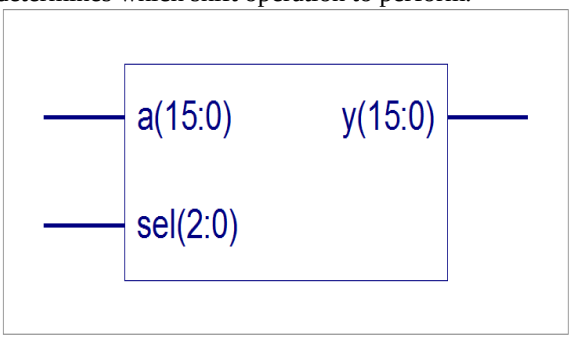
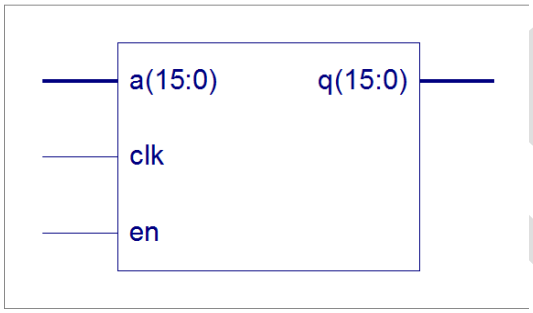


Fig 5. RTL schematic of shift register

Tri-register:

The last component of the CPU is the tristate register component, **trireg**. The tristate register is connected to the main data bus and can store information from the data bus as well as drive information to the data bus.



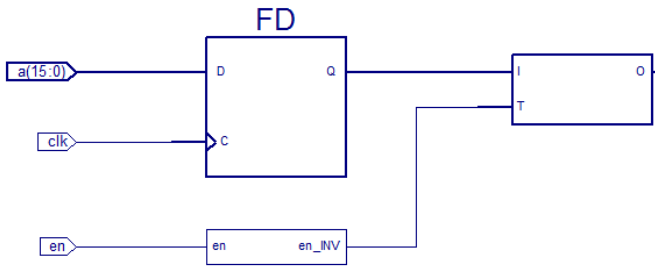


Fig 6. RTL schematic of tri-register

Arithmetic Logic Unit (ALU):

An ALU is a digital circuit that calculates arithmetic operations like: addition, subtraction, shifting and exclusive or. In this processor we have designed ALU to carry out logical operations, arithmetic operations like addition, subtraction, increment, decrement, etc.

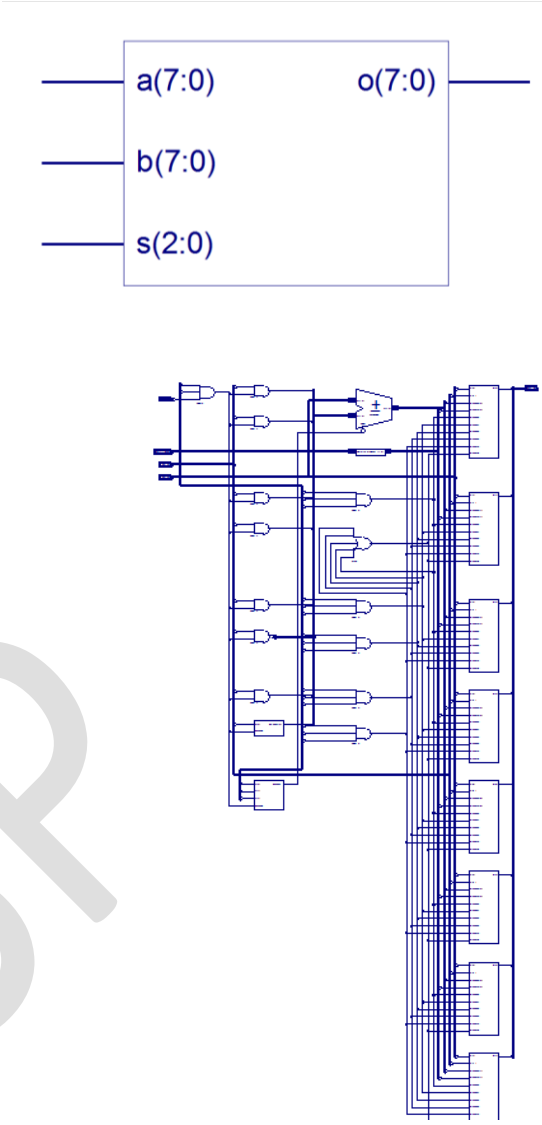


Fig 7. RTL schematic of ALU

IV. SAMPLE INSTRUCTION REPRESENTATION:

Instructions share common attributes, but come in a number of flavors. Sample instructions are shown in Figure below. All instructions contain the opcode in the five most significant bits of the instruction. Single-word instructions also contain two 3-bit register fields in the lowest 6 bits of the instruction. Some instructions, such as INC (Increment), only use one of the fields, but other instructions, such as MOV (Move), use both register fields to specify the From register and the To register. In double-word instructions, the first word contains the opcode and destination register address, and the second word contains the immediate instruction location or data value to be loaded. For instance, a LoadI (Load Immediate) instruction would look like this:

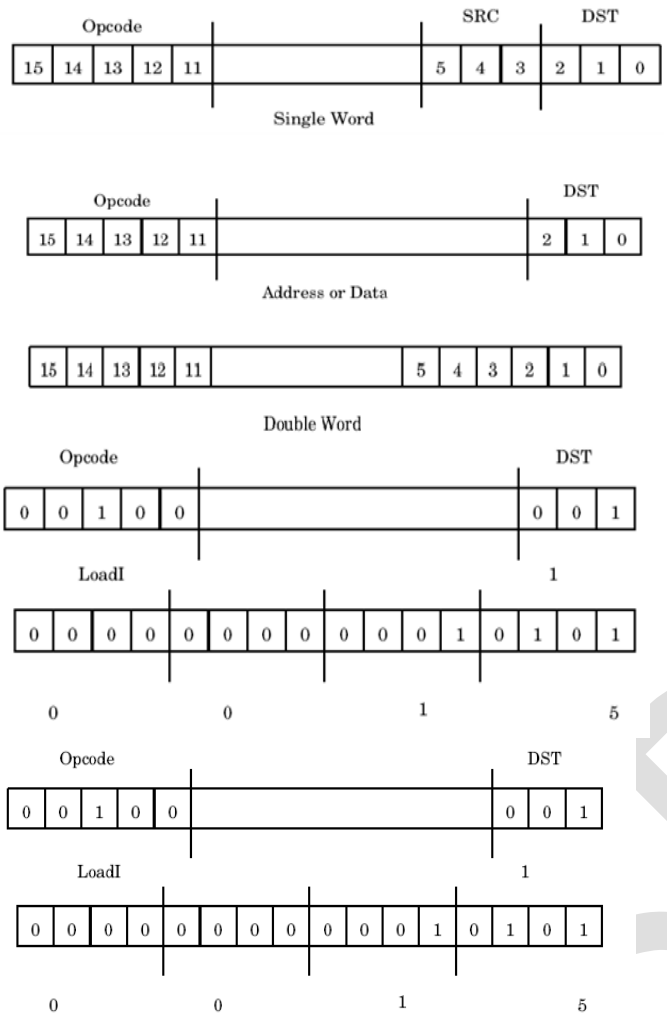


Fig 8. Structure of instruction in instruction register

This instruction loads the hex value 15 into register 1. The instruction words look like those shown in Figure 12-3. When the control unit decodes the opcode of the first word, it determines that the instruction is two words long and loads the second word to complete the instruction.

Typical commercial processors are much more complicated and have pipelined instruction streams for faster execution.To reduce complexity, this example is not pipelined.

V. IMPLEMENTATION AND RESULTS:

The results of simulation of processors and its sub-components generated on ModelSim simulator are shown below-

A) Control unit: The process that a control unit carries out whenever an instruction is fetched, can be observed in the output of simulator below:

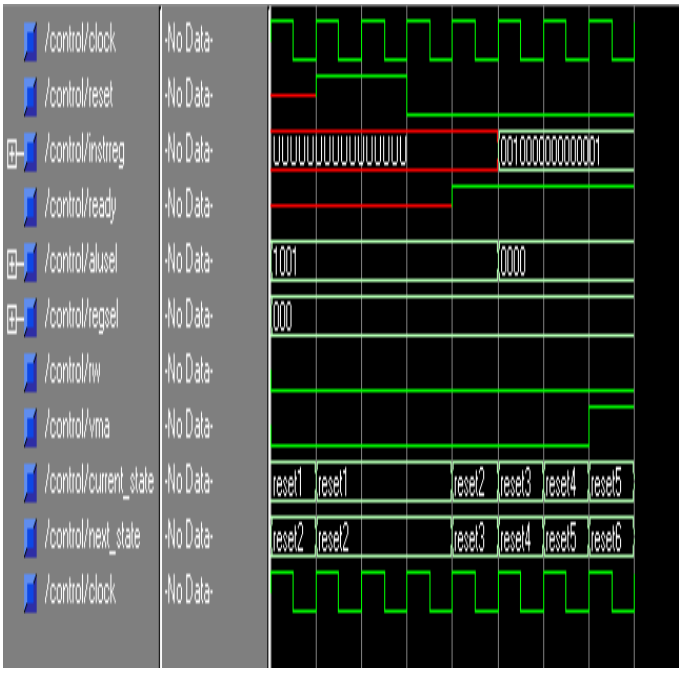


Fig 9. Control Unit simulation

B) ALU: The ALU has 'a' and 'b' as inputs along with 'sel' signal to select the operation to be performed.Output given by 'c'.

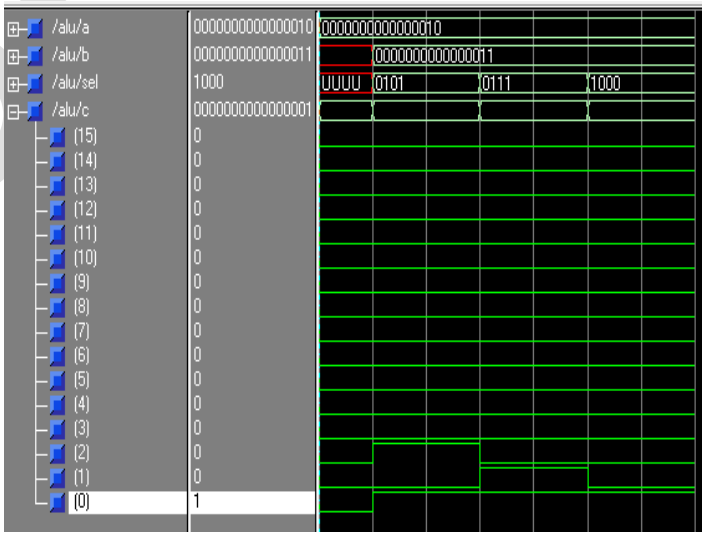


Fig 10. ALU Simulation

C) Central Processing Unit: CPU consist of all the registers and alu that together work for the functioning of CPU. The simulation results of CPU on Modelsim are as:

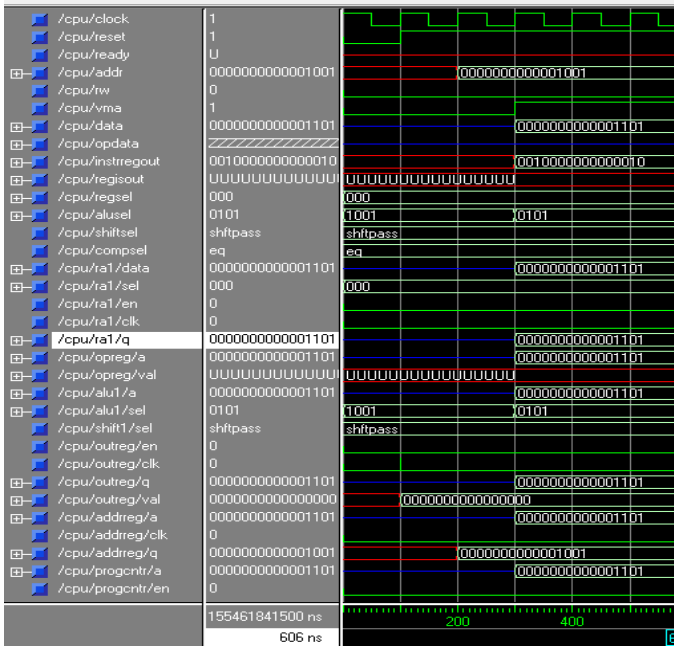


Fig 11. CPU Simulation

D) Once the sub-elements were designed and tested, they were put together to constitute the FPGA processor. The simulation results of whole processor in the below diagram shows the execution of instruction of instructions designed for the processor.

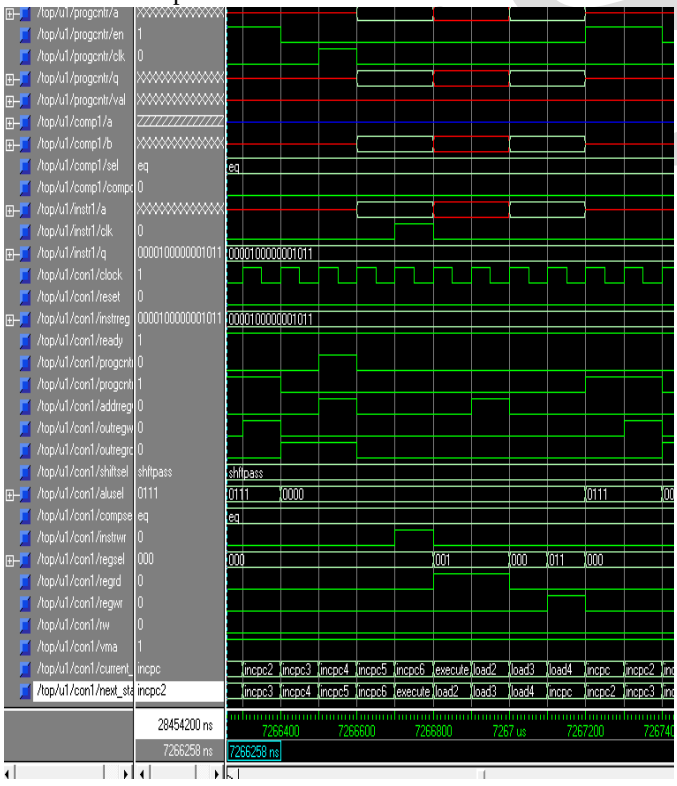


Fig 12. Processor simulation

VI. IMPLEMENTATION OF MEMORY BLOCK COPY USING FPGA PROCESSOR:

Using the FPGA processor we designed, we developed a sample program to copy a block of memory from one location to other. The results can be shown in ModelSim simulator as:

The below shown screenshot shows the numbers 1 to 15 in binary format that are stored in memory from location 16 to 32.

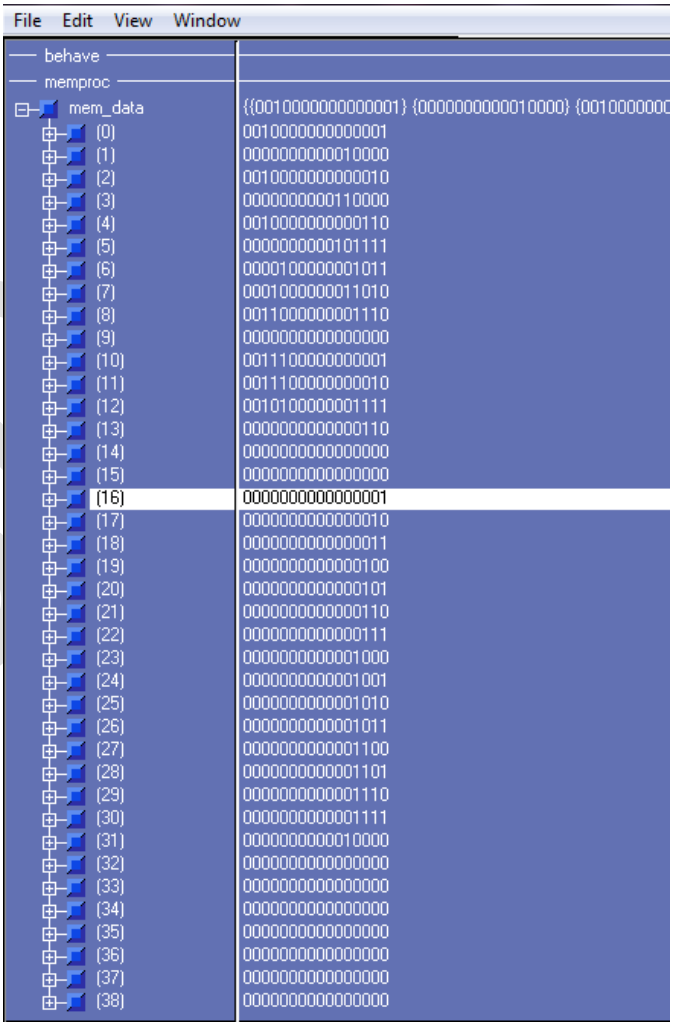


Fig 13(a). Memory block copy simulation

While the below screenshot shows whole block copied from location number 48 to 62, which was declared as destination

address.

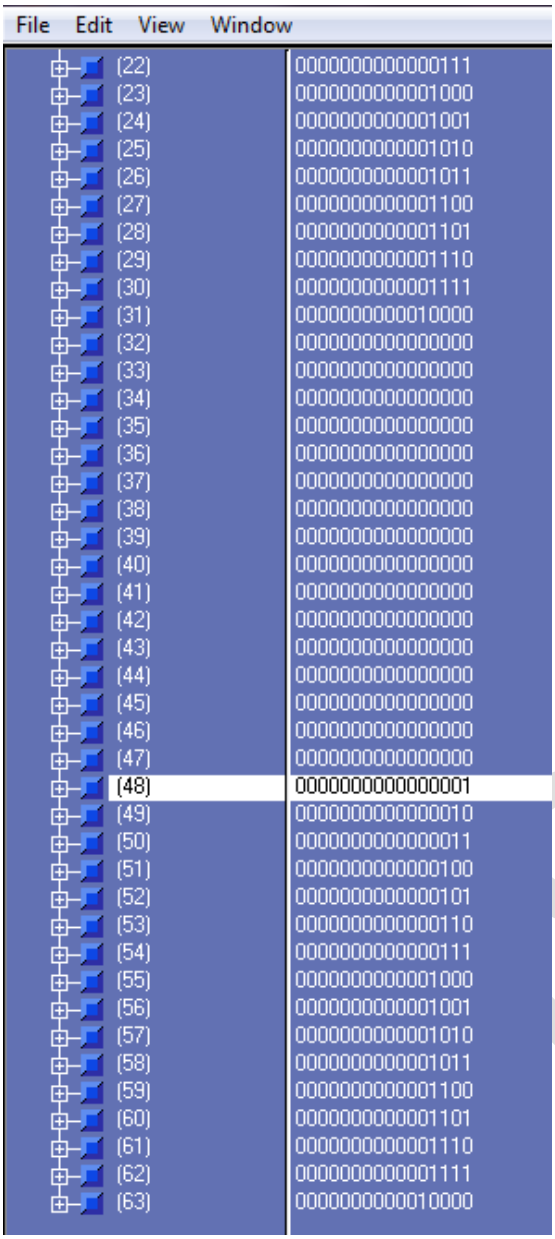


Fig13(b). Memory block copy simulation

CONCLUSION:

We have successfully implemented an FPGA-based 8-bit processor on Xilinx Spartan 3 board using the VHDL language. Our processor is made of the control unit, and an arithmetic logic unit (ALU) and a memory unit.

REFERENCES:

[1] FPGA Implementation of an 8-bit Simple Processor *E. Ayeh, K. Agbedanu, Y. Morita, O. Adamo, and P. Guturu* University of North Texas, Denton, TX. 76207 (IEEE PAPER 2008)

[2] Digital logic and microprocessor design using VHDL *By Hwang*

[3] Design and Implementation of a 8 bit RISC Processor on Xilinx FPGA *Wael M ElMedany, Khalid A AlKoohej* Department of Communications and Electrical Engineering, faculty of Engineering, Fayoum University, Egypt. Department, Computer,Engineering Information Technology College, University Of Bahrain, 32038 Bahrain

[4] V. S. Balakrishnan, H. Pottinger, F. Ercal, and M. Agarwal, Design and implementation of an FPGA based processor for compressed images (poster abstract). In *Proceedings of the 2000 ACM/SIGDA Eighth international Symposium on Field Programmable Gate Arrays* (Monterey, California, United States, February 10 - 11, 2000). FPGA '00. ACM, New York, NY, 218.

[5] R. Fryer, FPGA based CPU instrumentation for hard realtime embedded system testing. *SIGBED Rev.* 2, 2 (Apr. 2005), 39-42.

[6] P. Yiannacouras, Rose, J., and Steffan, J. G. 2005. The microarchitecture of FPGA-based soft processors. In Proceedings of *the 2005 international Conference on Compilers, Architectures and Synthesis For Embedded Systems* (San Francisco, California, USA, September 24 - 27, 2005). CASES '05. ACM, New York, NY, 202-212.

[7] G. Achery, C. Trinitis, R. Buchty, "CPU-independent assembler in an FPGA," *Field Programmable Logic and Applications, 2005. International Conference* , vol., no., pp. 519-522, 24-26 Aug. 2005.

[8] Y Nagaonkar and M. L. Manwaring. "An FPGA-based Experiment Platform for Hardware-Software Codesign and Hardware Emulation" In proceedings of *The 2006 World Congress in Computer Science, Compute Engineering, and Applied Computing* Pages: 169-174.

[9] D. Chiou, H. Sanjeliwala, D. Sunwoo, J. Z. Xu, and N.Patil, "FPGA-based Fast, Cycle-Accurate, Full-System Simulators," in *Proceedings of the second Workshop on Architecture Research using FPGA Platforms, held in conjunction with HPCA-12, Austin, TX*, Feb. 2006.