Design & Implementation of ALU, RAM, Control Unit for pic Microcontroller for Robotic Applications Using VHDL

Dharmana Bhagavan

Student (M.Tech), Department of ECE

Baba Institute of Technology (BITS), Visakhapatnam-41

B Siva Prasad

Associate Professor, Department of ECE

BITS, Visakhaptnam-41

Abstract- In this work, we have implemented 8 bit PIC using VHDL, which is compatible with Intel 8051. Project consists of the control block and a memory block communicating concluded a bi-directional data bus, an address bus, and a few control lines. The control unit raises instructions from the peripheral memory and implements these directions to run a program. These instructions are kept in the ROM and decoded by the decoder. The control unit produces the opposite signal to execute the instruction. With this PIC we can accomplish 8 bit data transfer, simple arithmetic & logical operation, branching process. Also this design support instantaneous, direct, register, indirect, register indirect addressing methods. In this design part, the codes for the PIC machineries like ALU (Arithmetic and Logic Unit), shift register, comparator, etc. have been written in VHDL (VHSIC Hardware Description Language) where VHSIC stands for Very High Speed Integrated Circuit. These discrete components together form PIC. In the simulation part, the codes have been tested and verified using Xilinx ISE 9.1i synthesis tool. The waveforms of some of the machineries are also been proved. In the implementation part, the codes have been developed using Xilinx. The components of microcontroller like ALU, RAM and ROM have also been industrialized using Xilinx ISE 12.4i.

Today the best mode to generate ASIC and FPGA devices is VHDL. Synthesis is a procedure of automatic adaptation of a higher level of construct to lower level of abstraction. There are numerous tools existing currently. In our project, we have used the commercially available Xilinx ISE 12.4i synthesis tool. It alters Register Transfer Level (RTL) explanation into gate level net lists. It can also be authenticate by FPGA kit. The design of this type of architecture gives the versatility of handling the memory capacity because the enthusiastic blocks are easily ascendable and adjustable from the same VHDL description.

1. INTRODUCTION

We Microcontroller designers have been inspiring for a long time for fascinating fastidious customers and every couple of days a new chip with a established operating frequency, more memory and progressive A/D converters seemed on the market. However, most of them had the equal or at least very similar architecture known in the world of microcontrollers as “8051 attuned”. What is all this near? The whole story has its early stages in the far 80s when Intel hurled the first series of

microcontrollers called the MCS 051. Even nonetheless these microcontrollers had quite unsure features in assessment to the new ones, they subject the world very soon and turn out to be a average for what at the present time is called the microcontroller. [1] The main cause for their great accomplishment and acceptance is a adeptly chosen conformation which gratifies different needs of a large number of workers allowing at the same time constant developments (refers to the new types of microcontrollers). Besides, the software has been established in great range in the interim, and it purely was not moneymaking to change anything in the microcontroller's rudimentary core. This is the reason for having a great number of various microcontrollers which basically are exclusively promoted versions of the 8051 family. [2]

Reconfigurable structures offer a explanation to solve multipart problems by conjoining the speed of hardware with the elasticity of software to improve show and system performance. Past three eras have seen the overview of the technology that has drastically changed the technique one studies and joysticks the world about them. A consequence of Microprocessor development by Intel is the 8051 Microcontroller which novelties its use in almost all walks of life. The Microcontrollers are not as well known to the general public, or even the technical community, as are the more trendy Microprocessors. The public is still very well aware that "approximately" is answerable for all the Smart VCRs, clock radios, washers and dryers, video games, TVs etc. That "something" is nothing but the Microcontroller. Any Microcontroller entails of three main components viz., Arithmetic and Logic Unit (ALU), a Random Access Memory (RAM) and a Read Only Memory (ROM). An ALU is responsible for Arithmetic and Logic operations like addition, subtraction, multiplication, AND, OR, NOT etc. operations. RAM is meant for storage and saving the data whenever necessary and ROM attends the purpose of storing the definitions affecting to the accomplishments agreeing to the op codes.

This paper presents the design, development and implementation of an Arithmetic and Logic Unit (ALU) and Random Access Memory (RAM) for PIC Microcontroller on Field Programmable Gate Arrays (FPGA) using VHDL i.e., VHSIC Hardware Description Language where VHSIC stands for Very High Speed Integrated Circuits. The ruse utilizations for ALU, RAM have been analyzed and tabulated to explore the design space.

2. LITERATURE REVIEW

Past few eras have seen gigantic research in the design and implementation of Microcontrollers. Arithmetic and Logic Unit (ALU) existence the computational heart of any controller needs to be implemented in an efficient method. In this regard, work by Shrivastava et. al. [3] presents VHDL situation for floating point arithmetic and logic unit using pipelining. Pipelining is used to perform multiple instructions simultaneously. In top-down design method, four arithmetic modules, addition, subtraction, multiplication and division are collective to form a floating point ALU unit. Each module is allocated into sub- modules. Two selection bits are joint to select a in the ALU design are comprehended using VHDL, design functionalities are legalized through VHDL simulation. Red-top by Khurana et. al [4] discusses the erection of arithmetic Logic Unit (ALU) using Hardware Description Language (HDL) using Xilinx ISE 9.2i and implementing it on Field Programmable Gate Arrays (FPGAs) to analyze the design parameters. Extraordinary developments in the wired and wireless communications area, the summons for secure data transmission increases [5]. In order to treasure trove solutions for this increasing directive new algorithms and security ethics such as Hash functions are developed. Paper [5] discusses about the implementation of the Hash processor using VHDL and FPGA. Work by Ashwini Deshmukh [6] put emphasis on on "novel leading one anticipation algorithm" allowing the author to suggestively reduce the eagerness failure rate with respect to the

state-of the art. The author claims to insert the technique into a broad FPU. The work is also compared for its presentation against dominant solutions, showing both area savings and total latency decrease. In tabloid [7] writers contemporaneous the design of an Arithmetic Logic Unit (ALU) based on Redundant Binary signed Digit (RBSD) Number System. Work by Khurana et. al. [8] throws light on implementation of a 32-bit Arithmetic Logic Unit (ALU) using VHDL. In this work the behavioral VHDL model of ALU is designed to achieve 16 operations which include both logical and arithmetic operations. Data preservation and leakage current are midst the most important area of apprehension in today's CMOS technology. In the paper [9] 6T SRAM cell has been analyzed on the basis of read noise margin (RNM), write noise margin (WNM), read delay, write delay, and data retention voltage (DRV). Application and simulations are approved out using VHDL. Paper [10] converses the FPGA implementation about DDR3 SDRAM controller for extraordinary presentation. Work by Vikas Gupta et. al. [11] describes a way for designing and implementing multiplierless digital PID controller based on Field Programmable Gate Array (FPGA) device. Paper [12] illuminates design of full architecture of an embedded processor for grasping arithmetic, logical, shifting and branching operations.

3. THE 8051 MICROCONTROLLER

Identical all good gears, this influential piece i.e., Microcontroller is mostly very simple. It is completed by act as a team tested and high- quality "requirements" (components) as per following recipe:

1. The unassuming computer processor is secondhand as the "brain" of the prospect system.

2. Depending on the perception of the manufacturer, a bit of recollection, a few A/D converters, timers, input/output lines etc. are additional.

3. All that is positioned in some of the normal packages.

4. Guileless software able to regulator it all and which everyone can simply learn about has been settled.

On the foundation of these instructions, plentiful categories of microcontrollers were designed and they rapidly became man's indistinguishable companion. Their incredible effortlessness and elasticity subjugated manhood a long time ago and if one attempts to invent something about them, one should know that they are possibly late; someone before you, has either done it or at least has taxed to do it [2].

3.1 Operation of Microcontroller

Even still there are a large number of altered types of microcontrollers and even more programs shaped for their use only, all of them have many belongings in communal. Thus, if you absorb to knob one of them you will be capable to handle them all. A distinctive situation on the basis of which it all roles is as follows:

1. Power supply is twisted on and everything flinches to ensue at high speed! The control logic unit saves everything beneath control. It spikes all other routes excluding quartz crystal to operate. While the provisions are in progress, the first milliseconds drive by.

2. Power supply voltage spreads its maximum and oscillator frequency develops stabilized. SFRs are being filled with bits sparkly the ceremonial of all circuits within the microcontroller. All pins are organized as inputs. The overall electronics starts operation in rhythm with pulse classification. After now on the time is unhurried in micro and nanoseconds.

3. Program Counter is fixed to zero. Instruction after that address is sent to instruction decoder which distinguishes it, after which it is executed through instantaneous effect.

4. The worth of the Program Counter is increment by 1 and the complete course is frequent several million times per second.

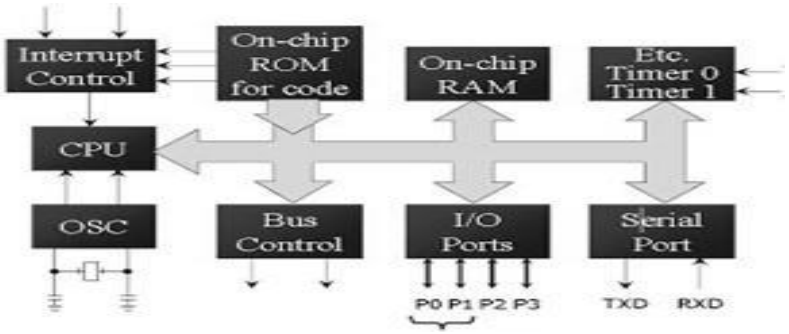


Fig.1. Basic interpretation of Microcontroller Architecture

For instance one can perceive, all the operations contained by the microcontroller are executed at high speed and reasonably simply, but the microcontroller itself would not be so expedient if there are not special circuits which mark it complete.

The main constituents which relief in making any Microcontroller dominant are as follows.

- Read Only Memory (ROM)

Read Only Memory (ROM) is a type of memory used to forever save the program which is being performed.

- Random Access Memory (RAM)

Random Access Memory (RAM) is a type of memory used for impermanent loading data and midway results twisted and used during the process of the microcontrollers.

- Electrically Erasable Programmable ROM (EEPROM)

The EEPROM is a singular type of memory not controlled in all microcontrollers. Its insides may be changed during program implementation (similar to RAM), but remainders lastingly saved even after the damage of power (comparable to ROM).

- Special Function Registers (SFR)

Special function registers are portion of RAM memory. Their determination is predefined by the constructor and cannot be reformed. Since their bits are physically associated to particular circuits within the microcontroller, such as A/D converter, serial communication module etc., any modification of their state straight affects the operation of the microcontroller or some of the circuits. For example, writing zero or one to the SFR regulatory an input/output port causes the proper port pin to be constructed as input or output. In other words, each bit of this register gear stick the function of one single pin.

- Program Counter

Program Counter is an device seriatim the program and points to the memory address covering the next order to perform. After each instruction performance, the value of the counter is increment by 1. For this motive, the program executes only single instruction at a time just as it is transcribed.

- Central Processor Unit (CPU)

As its term advises, this is a unit which displays and panels all processes within the microcontroller and the operator cannot affect its work. It contains of numerous smaller subunits, of which the most important are:

1. Instruction decoder
2. Arithmetical Logical Unit (ALU)
3. Accumulator

- Input/output ports (I/O Ports)

In instruction to make the microcontroller valuable, it is necessary to attach it to marginal devices. Each microcontroller has one or more records (called a port) associated to the microcontroller pins.

- Oscillator

Even pulses produced by the oscillator permit harmonic and synchronous process of all circuits within the microcontroller. It is typically organized as to use quartz-crystal or ceramics resonator for frequency stabilization.

The AT89S51 contains low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of in-system programmable Flash memory. The device is contrived using Atmel's high-density nonvolatile memory expertise and is attuned with the industry-standard 80C51 instruction set and pin out. The on-chip Flash agrees the program memory to be reprogrammed in-system or by a conservative nonvolatile memory programmer [1,2]. By joining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S51 is a influential microcontroller which delivers a highly-flexible and cost-effective solution to many embedded control tenders. Comprehensive architecture of Atmel's PIC Microcontroller can be seen in figure 2 as labeled in Atmel's data sheet.

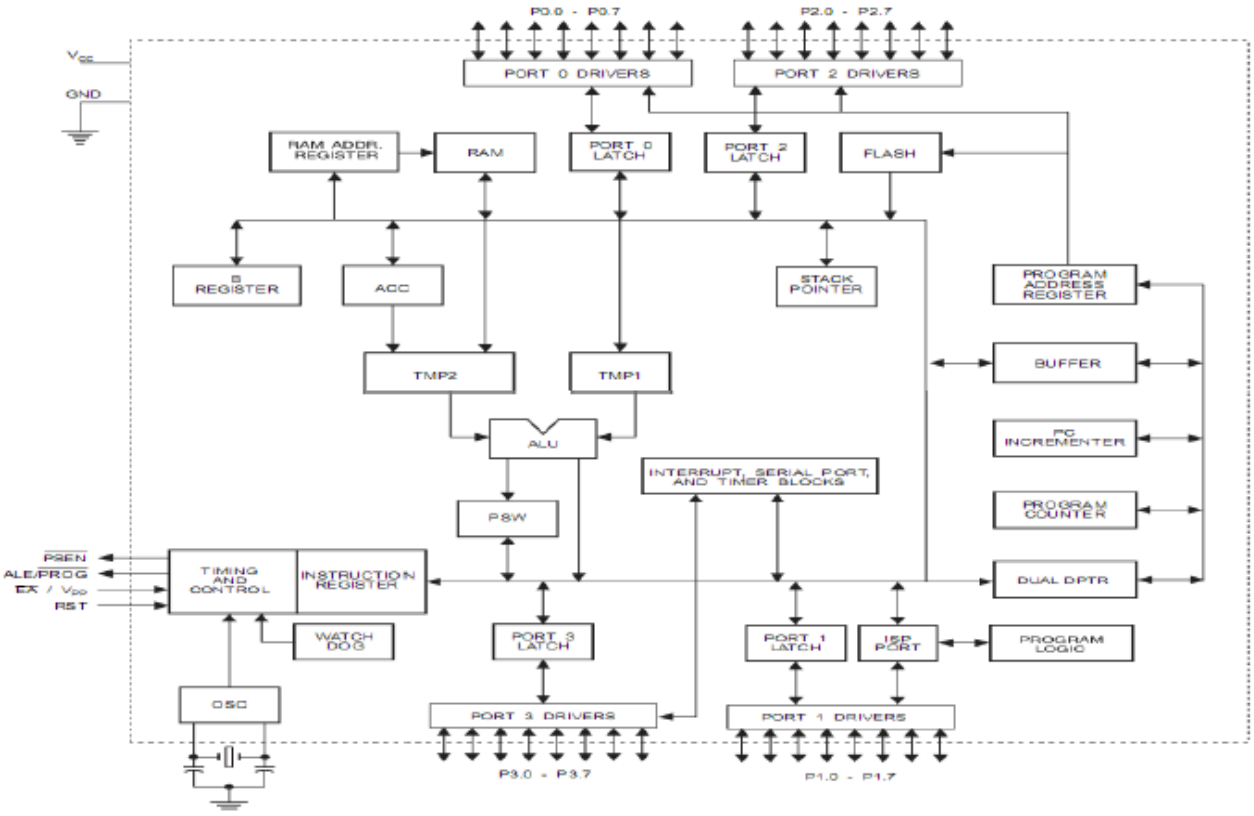


Fig.2- Architecture of PIC Microcontroller

4. DESIGN AND IMPLEMENTATION

The programming milieu for ALU, RAM and ROM is based on VHDL. The behavioral portrayal of the functionality is primarily written and is then examined, simulated and synthesized onto Xilinx FPGAs. An presentation for the ALU, RAM and ROM modules employment is developed by writing behavioral explanation in VHDL and the explanation is iteratively polished and debugged with the simulator presented i.e., ModelSim version 10.4 a. After the description code is verified to be functionally precise by simulation, it is interpreted onto a Xilinx net list form. The net list is then charted onto FPGA architecture by automatic partition, placement and routing tool to form a loadable FPGA article module. A inert timing analysis tool is then practical to the object module to govern maximum effective speed.

4.1 Entity Declaration

Entity declaration entails proclaiming input and output ports for the requisite design. The ALU collects op_code, a 4-bit input and two inputs of 8 bits each i.e., src_1 and src_2. The ALU is on condition that with reset signal rst. It is also on condition that with an ancillary carry (src_ac) and carry (src_in) as two more inputs to assistance arithmetic operations. It has two 8-bit outputs i.e., des_1 and des_2 and three single bit outputs auxiliary carry out (des_au), carry out (des_cy) and overflow bit(des_ov) . The entity statement for an 8-bit ALU can be seen in Figure 3.

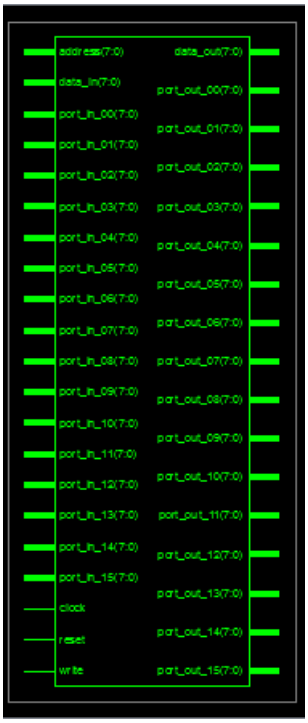


Fig.3. Entity Declaration of Memory

Subsequent the matching approach entity for RAM can be affirmed. It has an active high reset pin (rst), once asserted, the registers are set to evasion values, a clk (rising edge) : clock signal - all ram i/o is synchronous, an 8-bit address bus (addr), in_data pin which requires the data being written hooked on the ram/reg, an 8-bit out_data pin to read data from the ram/reg, a in_bit_data pin to write into the ram/reg, an out_bit_data pin for interpretation bit data from the ram/reg, an active little read signal (rd) which is proclaimed to signal a ram/reg read, an active low write signal (wr) declared to signal a ram/reg write, an lively high is_bit_addr pin for proclaiming if entreating a ram/reg bit-data and four 8-bit input and output ports. Entity statement for ROM can be seen in Figure 3 and entity declaration for ALU can be seen in Figure 4. The ROM has a 16-bit address bus (addr), a clock pin (clk), a read pin (rd), a reset pin (rst) and an 8-bit data bus. This read only memory attends the resolve of permanently storing the essential functionality.

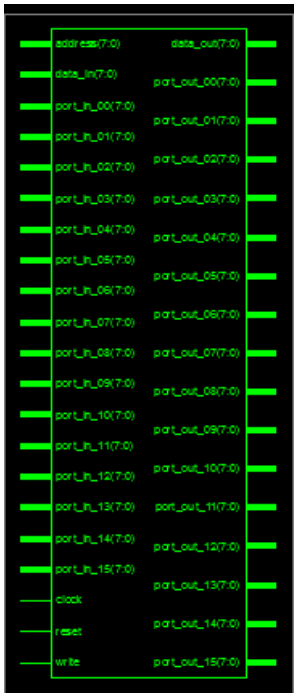


Fig.4. Entity Declaration of CPU

Subsequent the same approach entity for RAM can be affirmed. It has an lively high reset pin (rst), when emphasized, the registers are set to default values, a clk (rising edge) : clock signal - all ram i/o is synchronous, an 8-bit address bus (addr), in_data pin which specifies the data being written into the ram/reg, an 8-bit out_data pin to read data from the ram/reg, a in_bit_data pin to write into the ram/reg, an out_bit_data pin for reading bit data from the ram/reg, an active low read signal (rd) which is proclaimed to signal a ram/reg read, an lively low write signal (wr) proclaimed to signal a ram/reg write, an active high is_bit_addr pin for proclaiming if demanding a ram/reg bit-data and four 8-bit input and output ports. Entity declaration for ROM can be gotten in Figure 3 and entity declaration for CPU can be seen in Figure 4. The ROM has a 16-bit address bus (addr), a clock pin (clk), a read pin (rd), a reset pin (rst) and an 8-bit data bus. This read only memory attends the resolution of lastingly storing the mandatory functionality.

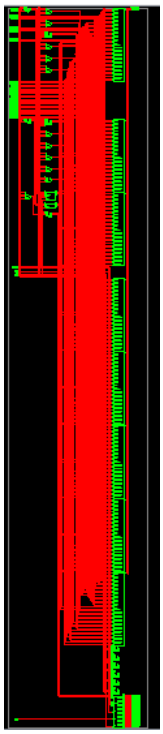
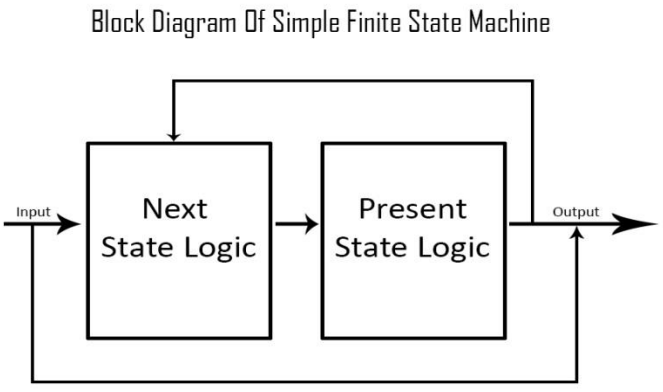


Fig.5 Basic RTL view of PIC Microcontroller

The controller segment planned as a FSM. The controller segment design as four control states as exposed in below block diagram of simple FSM. The roles of the controller and consistent CPU_STATES are: Reset Present State, Next State is earmarked for future development of interrupt handling, instruction fetch & decode , and instruction execute.



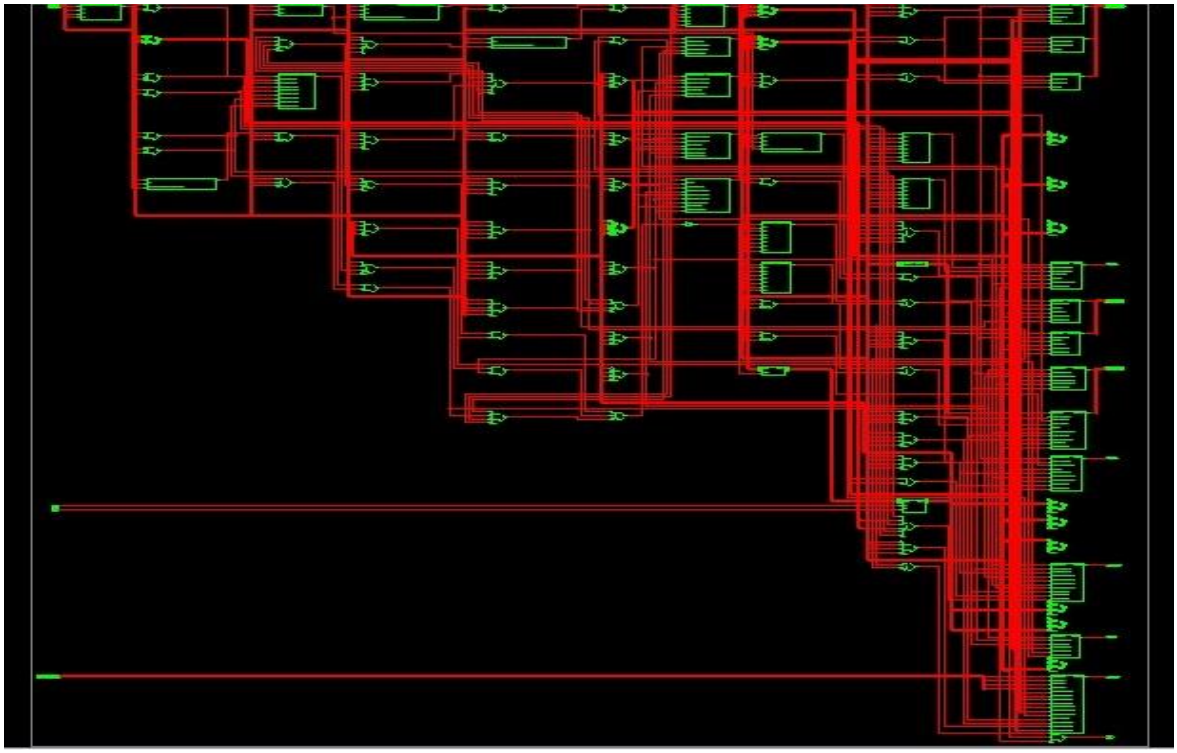


Fig.6- inside Architecture of PIC Microcontroller (Top level RTL)

5. RESULT

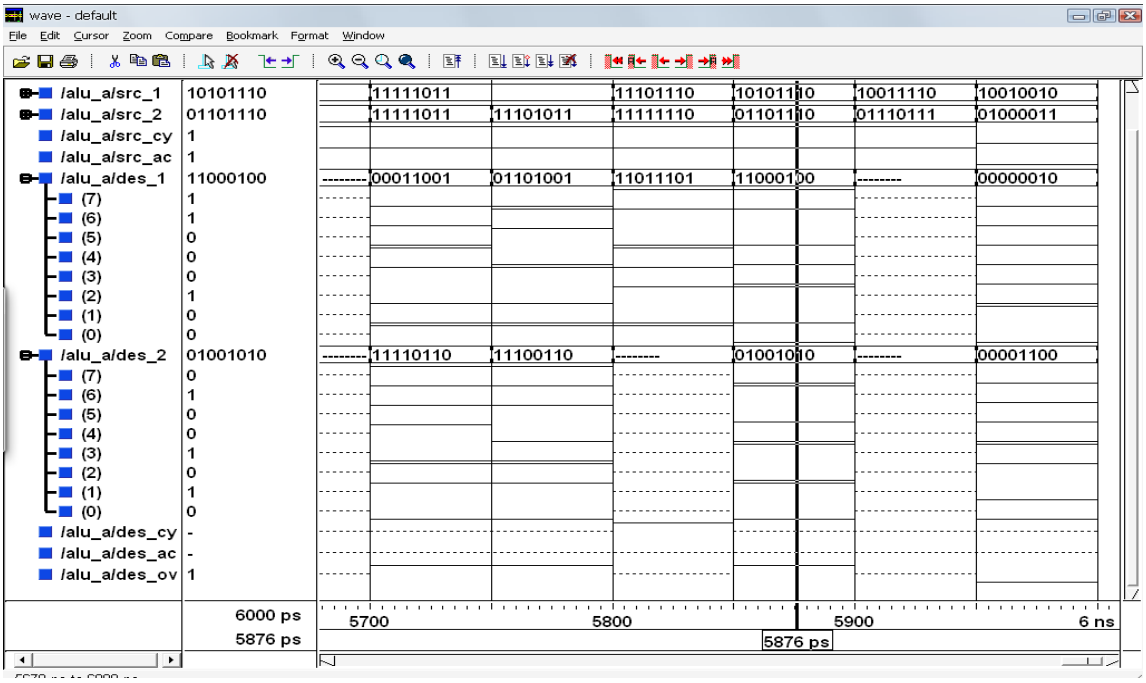


Fig.7- Simulation waveform of ALU

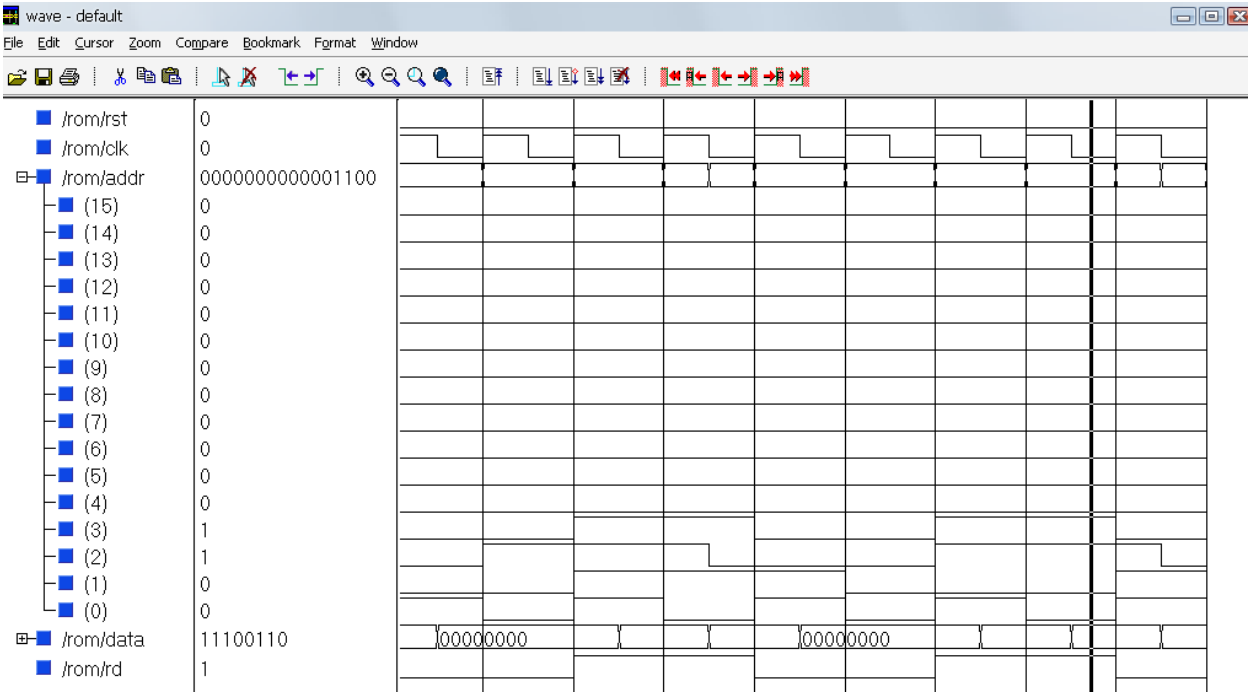


Fig.8- Simulation waveform of Memory Unit

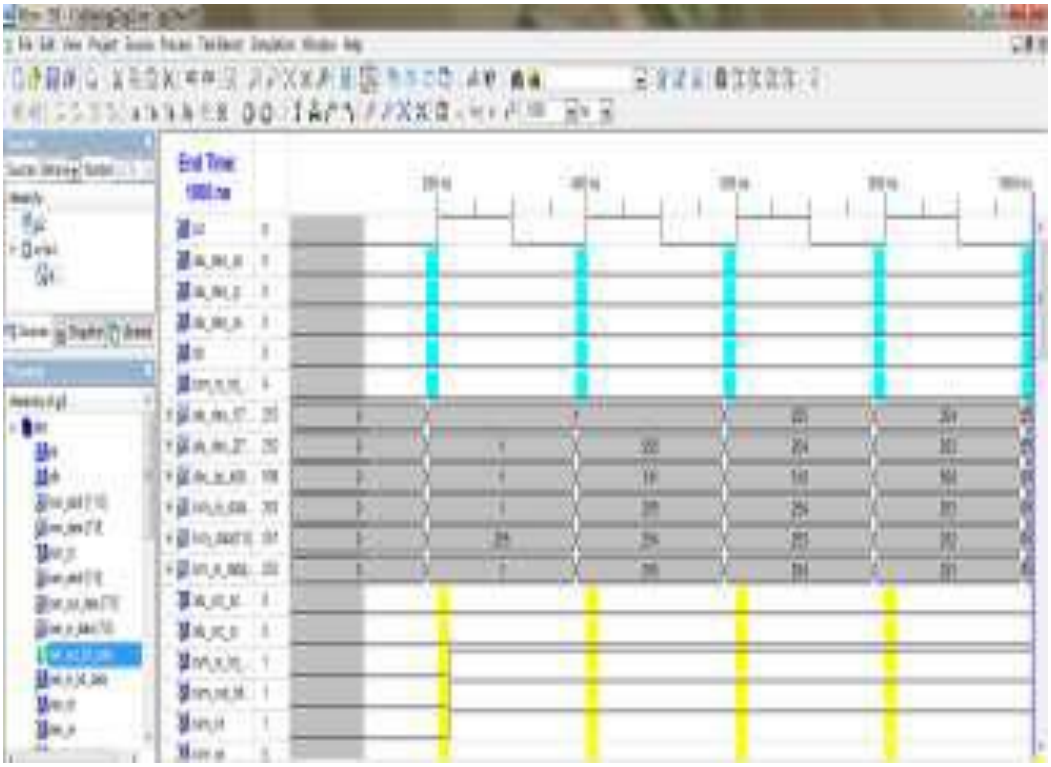


Fig.9- Simulation waveform of PIC

6. CONCLUSION-

In this paper an effort has been made to contrivance Arithmetic and Logic Unit (ALU), a Random Access Memory (RAM) specific for PIC Microcontroller on Field Programmable Gate Array (FPGA) using VHDL as the hardware description language. A collection to keep in path of the opcodes and constants which are PIC Microcontroller detailed has been fashioned. The design has been tested for its accuracy using Test Bench and simulations. The system designed is traveled for the design space. The design of this type of construction gives the versatility of managing the memory ability because the enthusiastic blocks are easily ascendable and adjustable from the same VHDL explanation. The methodology of this microcontroller guarantees a equivalent working with another type of microcontrollers or hardware elements of the same type, in order to be able to use in multitasking manner

ACKNOWLEDGEMENT

This work was supported in part by the BITS, Visakhapatanam.

REFERENCES

[1]. Angelov V, Lindenstruth V. The Educational Processor Sweet-16. In: 2009 International Conference on Field Programmable Logic and Applications. IEEE, 2009. p. 555–9.

[2] Jansen D, Dusch B. Every Student Makes his Own Microprocessor. In: 10th European Workshop on Microelectronics Education (EWME), IEEE, 2014, p. 97–101.

[3] Guzman F. Natalius 8 bit RISC: Overview. Date accessed: 08/06/2012

[4] Pepelyashev D. 8-bit Microcontroller with Extended Peripheral Set Cores. Date 10/08/2008.

[5] Riedel U. Tiny 8: Overview. Open Cores. Date accessed: 11/02/2007.

[6] Hays K, Jshamlet, Open 8 uRISC: Overview, Open Cores. Date accessed: 20/06/2016.

[7] Julio Enrique Rodríguez Prieto, Edwar Jacinto Gómez and Fernando Martínez Santa, "Implementation of an 8-Bit Softcore Microcontroller on Xilinx Spartan FPGA Family", Indian Journal of Science and Technology, *June 2017 Vol 10(22), DOI: 10.17485/ijst/2017/v10i22/114827.*

[8] Siddalingesh S Navalgund and Prakash R Tonse, Article: Design, Development and Implementation of ALU, RAM and ROM for 8051 Microcontroller on FPGA using VHDL, International Journal of Computer Applications 80(1):8-14,October 2013, Published by Foundation of Computer Science, New York USA.

[9] Kenneth J Ayala, West Carolina University, "The 8051 Microcontroller Architecture, Programming and Applications", West Publishing Company, St Paul. ISBN 0-314-77278-2.

[10] Shrivastav Purnima, Tiwari Mukesh, Singh Jaikaran and Rathore Sanjay, "VHDL Environment for Floating point Arithmetic Logic Unit- ALU Design and Simulation", Research Journal of Engineering Sciences, 1-6 August (2012), ISSN 2278- 9472, pp: 1-6,Vol.1(2)

[11] R. Uma, "Design and Performance Analysis of 8-bit RISC Processor using Xilinx Tool": International Journal of Engineering Research and Applications (IJERA) , 2012 Vol. 2, Issue 2, pp.053-058 53.

[12] AnnuVerma, "Design an 8 bit Microprocessor with all the Operations of 8051 Microcontroller with the help of VHDL", International Journal of Engineering, Applied and Management Sciences (IJEAMS), April 2013,Vol. 02, Issue 01,

[13] Raj Kamal, "Embedded Systems - Architecture, Programming and Design", Tata McGraw-Hill Education, 2008.

[14] Krishna Kant,"Microprocessors and Microcontrollers : Architecture, Programming and System Design 8085, 8086, 8051", PHI Learning

[15] Kui YI and Yue-Hua DING, " 32 bit multiplication and Division ALU Design Based on RISC Structure",WuHan Polytechnic University proc. 2009, International Joint Conference on ArtificalIntelligence,2009, pp. 761-764

[16]Rohit Sharma, Vivek Kumar Sehgal, Nitin,PranavBhasker and lshitaverma," Design and Implementation of a 64-bit RISC Processor using VHDL, Jaypee University , proc. Uksim, 11th International Conference on Computer Modelling and Simulation,2009,pp 568-573.

[17] Xia Li LongweiJi , Bo Shen, wenhong Li and Qianling Zhang,, " VLSI Implementation of a high – performance 32- bit RISC microprocessor", Fudan University,2002,pp.1458- 1461.

[18] Mamun Bin IbneKeaz, Md. Shabiul Islam, Mohd. S. Sulaiman, "A single Clock cycle MIPS RISC processor Design using VHDL",multimedia university, 2002 pp 199- 203.