Global Journal of researches in engineering
Electcical and electronical engineering
Volume 11 Issue 7 Version 1.0 December 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

16-Bit Embedded Microprocessor for Information System Using FPGA

By Abu Md. Numan-Al-Mobin, Sheikh Md. Rabiul Islam,
Md. Rezwanul Haque, Md. Hasanuzzaman
University of Engineering and Technology, Khulna, Bangladesh

Abstract - This paper introduces a novel 16-bit information processor in resister level to be used in an information processing system for wireless communication hardware. Different types of instructions like as add, sub, jump, load have to be executed by this processor. For these instruction we have to convert add, sub, jump, load into a binary bits. For the different instructions that's binary bits help to differentiate the overall operations. The main work is to execute any instruction to create control signal. We have made different type of control signal for different instruction. In the begging (turn on the processor) there is no data in the RAM so we have to write some instruction set in binary bit that would be executed and also have a input pin for this. When the pin is 1 the instructions would be written and set 1, then processor will execute all the instructions one by one. We have used FPGA that helps to show the timing response of the instruction that executed form. We have analysis timing response of all block and final TOP blocks FPGA.

Keywords : 16 bit information processor, Program counter, MAR, RAM, ALU.

GJRE Classification : FOR Code: 050299

Strictly as per the compliance and regulations of :

16-Bit Embedded Microprocessor for Information System Using FPGA

Abu Md. Numan-Al-Mobin[α], Sheikh Md. Rabiul Islam[Ω], Md. Rezwanul Haque[β], Md. Hasanuzzaman[ψ]

Abstract - This paper introduces a novel 16-bit information processor in resister level to be used in an information processing system for wireless communication hardware. Different types of instructions like as add, sub, jump, load have to be executed by this processor. For these instruction we have to convert add, sub, jump, load into a binary bits. For the different instructions that's binary bits help to differentiate the overall operations. The main work is to execute any instruction to create control signal. We have made different type of control signal for different instruction. In the begging (turn on the processor) there is no data in the RAM so we have to write some instruction set in binary bit that would be executed and also have a input pin for this. When the pin is 1 the instructions would be written and set 1, then processor will execute all the instructions one by one. We have used FPGA that helps to show the timing response of the instruction that executed form. We have analysis timing response of all block and final TOP blocks FPGA.

Keywords : 16 bit information processor, Program counter, MAR, RAM, ALU.

I. Introduction

An information processor or information processing system, as its name suggests, is a system (be it electrical, mechanical or biological) which takes information (a sequence of enumerated states) in one form and processes (transforms) it into another form, e.g. to statistics, by an algorithmic process. An information processing system is made up of four basic parts, or sub-systems: input, microprocessor, storage and output.

A microprocessor incorporates the functions of a computer's central processing unit (CPU) on a single integrated circuit [1, 3]. A microprocessor - also known as a CPU or central processing unit - is a complete computation engine that is fabricated on a single chip. It is a multipurpose, programmable, and clock-driven, register based electronic device that accepts binary data as input, processes it according to instructions stored in its memory, and provides results as output. The Microprocessor communicates and operates in the binary numbers 0 and 1. That Called bits. Each Microprocessor has a fixed set of instructions in the form of Binary patterns called a Machine Language [9]. It is difficult for humans to communicate in the language of 0s and 1s. Therefore, the binary instructions are given abbreviated names, mnemonics, which form the assembly language for a given microprocessor. The main purpose of our microprocessor is to introduce all the crucial ideas behind computer operation without burying you in unnecessary detail. Microprocessor indicates the width of the registers. A 16-bit microprocessor can process data and memory addresses that are represented by 16 bits. A microprocessor or processor is the heart of the computer and it performs all the computational tasks, calculations and data processing etc. inside the computer. Microprocessor is the brain of the computer. In the computers, the most popular type of the processor is the Intel Pentium chip and the Pentium 1V is the latest chip by Intel Corporation. A silicon chip that contains a CPU [2]. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles. Say we want to load a program like:

LDA 00100100
DEL 00011110

For first instruction, we need some place to write and read. The RAM is that place. However, there is a lot of place. Therefore, we should write a specific location of RAM. MAR use to specify the RAM location. There are two instructions LDA and DEL. We should execute these one after another which done by the program counter. A microprocessor does not understand without binary instruction so LDA should be converting into binary bits. It is done by the instruction register. Controller creates the different types of signals for various instructions.

Different types of instructions have to be executed. Different types of instruction may be add, sub, jump, load etc. For the instruction we have to convert add, sub, jump, load etc. into a binary bits and that's binary bits can be found by instruction register. For the different operation, the binary bits are not same. For the different instructions that's binary bits help to differentiate the overall operations.

We used FPGA that helps to show the timing response of the instruction that executed form [20, 21]. Xilinx software is used for the block diagram and circuit

Author α : is with Khulna University of Engineering and Technology, Khulna, Bangladesh.
(Phone : +8801716637972; E-mail : nmobin27@yahoo.com).
Author Ω : is with Khulna University of Engineering and Technology, Khulna, Bangladesh.
(Phone : +8801714087378; E-mail : robi@ece.kuet.ac.bd)
Author β :is with Khulna University of Engineering and Technology, Khulna, Bangladesh. (E-mail : rizuece.kuet@yahoo.com)
Author ψ : is with Khulna University of Engineering and Technology, Khulna, Bangladesh. (E-mail : hasanasah@yahoo.com)

level design [11, 12]. By this thesis we can learnt how the instructions can be executed.

The paper consists a general overview of the Microprocessor and it's architecture and how it works. This chapter gives the idea of timing response of different instruction that executed & conclusion and future works.

II. Proposed Architecture of 16-Bit Microprocessor for Information Systems

a) *Typical model of Information system*

An object may be considered an information processor if it receives information from another object and in some manner changes the information before transmitting it. There are four basic parts, or sub-systems of an information processing system that are input, microprocessor, storage and output. Figure 1 shows the block diagram of an information processing system and its workflow.

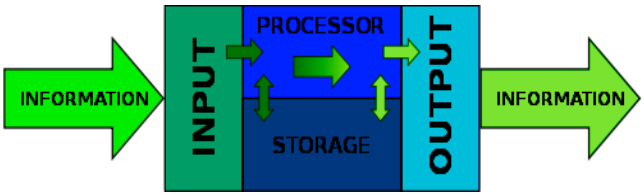

Fig.1 : Typical block diagram of information processing system.

b) *Microprocessor Model*

In proposed 16-bit microprocessor model all register outputs to the W bus are three-states, this allow orderly transfer of data. All other register outputs are two-states; these outputs continuously drive the boxes they are connecting to. Control unit that helps how the data works as input. Microprocessors are the devices in a computer that make things happen. Microprocessors are capable of performing basic arithmetic operations, moving data from place to place, and making basic decisions based on the quantity of certain values.

The block diagram shown in Figure 2 has Program Counter, MAR, RAM Instruction Register Controller, Accumulator ALU, B Register, Temporally and Counter all the parts connected to the W Bus. Data passed through the W Bus. Controller helped to data which way the binary data connected as the input of block diagram.

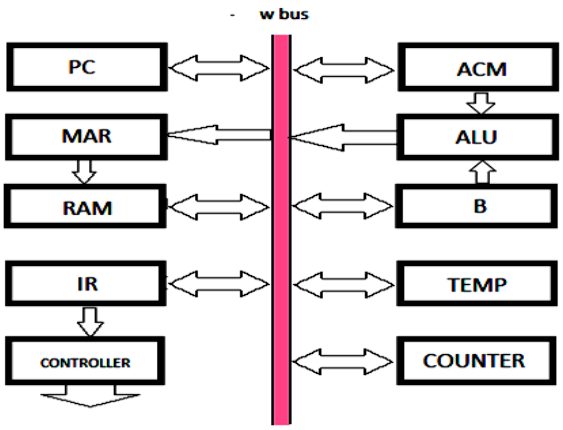

Fig.2 : Block diagram of 16-bits microprocessor architecture for information systems.

The components of this simple microprocessor:

- Registers A, B and C are simply latches made out of flip-flops. (See the section on "edge-triggered latches" in How Boolean Logic Works for details.)
- The address latch is just like registers A, B and C.
- The program counter is a latch with the extra ability to increment by 1 when told to do so, and also to reset to zero when told to do so.
- The ALU could be as simple as an 8-bit adder (see the section on adders in How Boolean Logic Works for details), or it might be able to add, subtract, multiply and divide 8-bit values. Let's assume the latter here.
- The test register is a special latch that can hold values from comparisons performed in the ALU. An ALU can normally compare two numbers and determine if they are equal, if one is greater than the other, etc. The test register can also normally hold a carry bit from the last stage of the adder. It stores these values in flip-flops and then the instruction decoder can use the values to make decisions.
- There are six boxes marked "3-State" in the diagram. These are tri-state buffers. A tri-state buffer can pass a 1, a 0 or it can essentially disconnect its output (imagine a switch that totally disconnects the output line from the wire that the output is heading toward). A tri-state buffer allows multiple outputs to connect to a wire, but only one of them to actually drive a 1 or a 0 onto the line.
- The instruction register and instruction decoder are responsible for controlling all of the other components.

c) *Microprocessor Logic*

To understand how a microprocessor works, it is helpful to look inside and learn about the logic used to create one. In the process you can also learn about assembly language - the native language of a microprocessor - and many of the things that engineers can do to boost the speed of a processor. A microprocessor executes a collection of machine instructions that tell the processor what to do. Based on the instructions, a microprocessor does three basic things [14]:

- Using its ALU (Arithmetic/Logic Unit), a microprocessor can perform mathematical operations like addition, subtraction, multiplication and division. Modern microprocessors contain complete floating-point processors that can perform extremely sophisticated operations on large floating-point numbers.
- A microprocessor can move data from one memory location to another.
- A microprocessor can make decisions and jump to a new set of instructions based on those decisions.

This is an edge-triggered device. The rising or falling edges of the primary clock drives everything else, with a slight delay, which may or may not be shown in these timing diagrams. ALE drops on the rising edge of the clock. The address lines are guaranteed to be set and stable when ALE drops and that is when the 8085 latches used to hold an address are triggered. It doesn't matter when AD0-AD7 are dropped, as long as they are dropped long enough after ALE is dropped to not corrupt the address latches. AD0-AD7 are probably dropped on the next primary clock edge.

d) Program Counter

The program counter, PC, is a special-purpose register that is used by the processor to hold the address of the next instruction to be executed. The Program Counter (PC) is a register structure that contains the address pointer value of the current instruction. Each cycle, the value at the pointer is read into the instruction decoder and the program counter is updated to point to the next instruction.

The program is stored at the beginning of the memory with the first instruction at binary address 00000000. The second instruction at address 00000001. The third at address 00000010. And so on. The program counter, which is part of the control unit, counts from 00000000 to 11111111. Their job is to send to the memory the address of the next instruction to be fetched and executed.

The program counter is reset to 00000000 before each computer run. When the computer run begins, the program counter sends address 00000000 to the memory. The program counter is then incremented to get 00000001. After the first instruction is fetched and executed, the program counter address 00000001 to memory. Again, the program counter is incremented. After the second instruction is fetched and executed. The program counter sends address 00000010 to the memory. In the way, the program counter is keeping track of the next instruction to be fetched and executed.

e) Input and MAR

Below the program counter is the input and MAR block. It includes the address and data switch register, which contains the address field that may execute. This means that which memory address will be used for any operation. These switch register send 8 address bits and 16 data bits to the RAM.

The memory address register (MAR) is the part of microprocessor. During a computer run, the address in the program counter is latched into the MAR. The Memory Address Register (MAR) is a CPU register that either stores the memory address from which data will be fetched to the CPU or the address to which data will be sent and stored. MAR holds the memory location of data that needs to be accessed. When reading from memory, data addressed by MAR is fed into the MDR (memory data register) and then used by the CPU. When writing to memory, the CPU writes data from MDR to the memory location whose address is stored in MAR. One of the registers located in the computer's processor is the memory address register or MAR. The MAR stores the physical memory address where the next piece of data will be written or the next instruction.

f) The RAM

The RAM is a 16 X 8 static TTL RAM. It allows to store a program and data in the memory before a computer run[4,5]. We can program the RAM–by means of the address and data switches. During a computer run, the RAM receives 8-bits addresses from the MAR and a read or write operation is performed. In this way, the instruction or data word stored in the RAM is placed on the W bus for use in some other part of the computer. Random-access memory (RAM) is a form of computer data storage. Today, it takes the form of integrated circuits that allow stored data to be accessed in any order with a worst-case performance of constant time. Random access memory (RAM) is the best-known form of computer memory. RAM is considered "random access" because you can access any memory cell directly if you know the row and column that intersect at that cell. The opposite of RAM is serial access memory (SAM). SAM stores data as a series of memory cells that can only be accessed sequentially (like a cassette tape). If the data is not in the current location, each memory cell is checked until the needed data is found. SAM works very well for memory buffers, where the data is normally stored in the order in which it will be used (a good example is the texture buffer memory on a video card). RAM data, on the other hand, can be accessed in any order. Similar to a microprocessor, a memory chip is an integrated circuit (IC) made of millions of transistors and capacitors. In the most common form of computer memory, dynamic random access memory (DRAM), a transistor and a capacitor are paired to create a memory cell, which represents a single bit of data. The capacitor holds the bit of information -- a 0 or a 1 (see How Bits and Bytes Work for information on bits). The transistor acts as a switch that lets the control circuitry on the memory chip read the capacitor or change its state. A capacitor is like a small bucket¬ that is able to store electrons. To store a 1 in the memory cell, the bucket is filled with electrons. To store a 0, it is emptied. The problem with the capacitor's bucket is that it has a leak. In a matter of a few milliseconds a full bucket becomes empty. Therefore, for dynamic memory to work, either the CPU or the memory controller has to come along and recharge all of the capacitors holding a 1 before they discharge. To do this, the memory controller reads the memory and then writes it right back. This refresh operation happens automatically thousands of times per second.

g) Instruction Register

The instruction register is a part of control unit [5]. To fetch an instruction from the memory the computer does a memory read operation. This places the contents of the addressed memory location on the W bus. At the same time, the instruction register is set up for loading on the next positive clock edge.

The contents of the instruction register are split into two nibbles. The upper nibble is two-state output that goes directly to the blocked labeled "control sequence." The lower nibble is a three-state output that is read onto the W bus when needed. In computing, an instruction register is the part of a CPU's control unit that stores the instruction currently being executed or decoded. In simple processors each instruction to be executed is loaded into the instruction register which holds it while it is decoded, prepared and ultimately executed, which can take several steps. Processors that are more complicated use a pipeline of instruction registers where each stage of the pipeline does part of the decoding, preparation or execution and then passes it to the next stage for its step. Modern processors can even do some of the steps of out of order as decoding on several instructions is done in parallel. A hardware element that receives and holds an instruction as it is extracted from memory; the register contains or is connected to circuits that interpret the instruction (or discover it's meaning), also known as current-instruction register.

h) Controller Sequencer

Before each computer, a CLR signal is sent to the program counter and a CLR signal to the instruction register. This resets the program counter to 00000000 and wipes out the last instruction in the instruction register.The 12 bits that come out of the controller-sequence from a word controlling the rest of the computer (like a supervisor telling others what to do) The 12 wires carrying the control word are called the control bus. The control word the has the format of CON=CpEpLmCE L1E1LaEaSvEvLbLo

This word determines how the registers will react to the next positive CLK edge. For instance a high Ep and a low Lm mean that the contents of the program counter are latched into the MAR on the next positive clock edge. As another example a low CE and a low La mean than the address RAM word will be transferred to the accumulator on the next positive clock edge. Later, we will examine the timing diagram to see exactly when and how these data transfer take place.

i) Accumulator

The accumulator (A) is a buffer register that stores inter-mediate answers during a computer run. The accumulator has two state output. The two state output goes directly to the adder-subtractor. The three state output goes to the W bus. In a computer's central processing unit (CPU), an accumulator is a register in which intermediate arithmetic and logic results are stored. In a computer's central processing unit (CPU), an accumulator is a register in which intermediate arithmetic and logic results are stored. Without a register like an accumulator, it would be necessary to write the result of each calculation (addition, multiplication, shift, etc.) to main memory, perhaps only to be read right back again for use in the next operation. Access to main memory is slower than access to a register like the accumulator because the technology used for the large main memory is slower (but cheaper) than that used for a register. The canonical example for accumulator use is summing a list of numbers. The accumulator is initially set to zero, and then each number in turn is added to the value in the accumulator. Only when all numbers have been added is the result held in the accumulator written to main memory or to another, non-accumulator, CPU register. An accumulator machine, also called a 1-operand machine, or a CPU with accumulator-based architecture, is a kind of CPU in which—although it may have several registers—the CPU always stores the results of most calculations in one special register—typically called "the" accumulator of that CPU. Historically almost all early computers were accumulator machines; and many microcontrollers still popular as of 2010 (such as the 68HC12, the PIC micro, the 8051 and several others) are basically accumulator machines.

j) B Register

The B register is another buffer register. It is used in arithmetic operation. A low Lb and positive clock edge load the word on the W bus into the B register. The two-state output of the B register drives the adder-subtractor, supplying the number to be added or subtracted from the contents of accumulator. In computer architecture, a processor register (or general purpose register) is a small amount of storage available on the CPU whose contents can be accessed more quickly than storage available elsewhere. Typically, this specialized storage is not considered part of the normal memory range for the machine. Most, but not all, modern computers adopt the so-called load-store architecture. Under this paradigm, data is loaded from some larger memory-be it cache or RAM-into registers, manipulated or tested in some way (using machine instructions for arithmetic/logic/comparison) and then stored back into memory, possibly at some different location. A common property of computer programs is locality of reference: the same values are often accessed repeatedly; and holding these frequently used values in registers improves program execution performance. Processor registers are at the top of the memory hierarchy, and provide the fastest way for a CPU to access data. The term is often used to refer only to the group of registers that are directly encoded as part of an instruction, as defined by the instruction set. More properly, these are called the "architectural registers". For instance, the IA-32 instruction set defines a set of 32-bit registers, but a CPU that implements the x86 instruction set will often contain many more registers than just these registers.

k) Instruction Set

A computer is a useless pile of hardware until someone programs it[1]. This means loading step by step instructions into the memory before the start of computer run. An instruction set, or instruction set architecture (ISA), is the part of the computer

architecture related to programming, including the native data types, instructions, registers, addressing modes, memory architecture, interrupt and exception handling, and external I/O. An ISA includes a specification of the set of opcodes (machine language), and the native commands implemented by a particular processor. Instruction set architecture is distinguished from the micro architecture, which is the set of processor design techniques used to implement the instruction set. Computers with different micro architectures can share a common instruction set. For example, the Intel Pentium and the AMD Athlon implement nearly identical versions of the x86 instruction set, but have radically different internal designs.
Example: add, sub, and jump.

l) Mnemonics

LDA, ADD, SUB, OUT, and HLT are the instruction set[7]. Abbreviated instruction like these is called mnemonics (memory aids). Mnemonics are popular in computer work because they remind the operation that will take place when the instruction is executed. The word mnemonic is derived from the Ancient Greek wormνημονι κόgmnemonikos ("of memory") and is related to Mnemosyne ("remembrance"), the name of the goddess of memory in Greek mythology. Both of these words refer back to μνημα mnema("remembrance").[2] Mnemonics in antiquity were most often considered in the context of what is today known as the Art of Memory.

m) Fetch Cycle

The control unit is the key to a computer's automatic operation. The control unit generates the control words that fetched and execute, the computer passes through different timing states (T states), periods during which register contents change. Let's find out more about this T states. An instruction cycle (sometimes it is called fetch-and-execute cycle, fetch-decode-execute cycle, or FDX) is the basic operation cycle of a computer. It is the process by which a computer retrieves a program instruction from its memory, determines what actions the instruction requires, and carries out those actions. This cycle is repeated continuously by the central processing unit (CPU), from boot up to when the computer is shut down. The next instruction is fetched from the memory address that is currently stored in the Program Counter (PC), and stored in the Instruction register (IR). At the end of the fetch operation, the PC points to the next instruction that will be read at the next cycle. Clock Pulse: T0-T1

A "fetch" means the read cycle used to read the code bytes that make up an instruction. There isn't a "fetch" instruction. A "read" instruction is one that reads data from memory (like LDA), and a "write" instruction is one that writes data (like STA).

n) Ring Counter

A ring counter is a type of counter composed of a circular shift register [7]. The output of the last shift register is fed to the input of the first register.
There are two types of ring counters:

- A straight ring counter or Over beck counter connects the output of the last shift register to the first shift register input and circulates a single one (or zero) bit around the ring. For example, in a 4-register one-hot counter, with initial register values of 1000, the repeating pattern is: 1000, 0100, 0010, 0001, 1000... . Note that one of the registers must be pre-loaded with a 1 (or 0) in order to operate properly.
- A twisted ring counter (also called Johnson counter or Moebius counter) connects the complement of the output of the last shift register to its input and circulates a stream of ones followed by zeros around the ring. For example, in a 4-register counter, with initial register values of 0000, the repeating pattern is: 0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000... .

The ring counter, which has output of T=T6T5T4T3T2T1

o) Clock Speed

Clock speed is measured in the MHz and it determines that how many instructions a processor can processed. The speed of the microprocessor is measured in the MHz or GHz. The processor is also known as the CPU (Central Processing Unit). It contains the control unit and the arithmetic unit and both works together to process the commands. CPU is used in every computer whether it is a workstation, server or a laptop. CPU is a complete computational engine that is designed as a chip. It starts the work when you turn on your computer. CPU is designed to perform the arithmetic and logical operations inside the computer. Common operations inside the computer include adding, subtracting, multiplying, comparing the values and fetching the different numbers to process them. The higher the CPU clocks' speed the more efficient will be the performance of the computer. Computer's performance is also influenced by the system bus architecture, memory used, type of the processor and software program being running. Pentium 4 is the fastest type of the Intel's processor that contains 125,000,000 transistors and operates at the speed of 3.6 GHz.

p) Clock Rates

Microprocessors are typically discussed in terms of their clock speed. The clock speed is measured in hertz (or megahertz, or gigahertz). A hertz is a "cycle per second". Each cycle, a microprocessor will perform certain tasks, although the amount of work performed in a single cycle will be different for different types of processors. The amount of work that a processor can complete in a single cycle is measured in "cycles per instruction". For some systems, such as MIPS, there is 1 cycle per instruction. For other systems, such as modern x86 chips, there are typically very many cycles per instruction.

The clock rate is equated as such [1, 2]:

$$\text{Clock Time} = \frac{1}{Clock\ Rate}$$

This means that the amount of time for a cycle is inversely proportional to the clock rate. A computer with a 1MHz clock rate will have a clock time of 1 microsecond. A modern desktop computer with a 3.2 GHz processor will have a clock time of approximately 3× 10-10seconds, or 300 picoseconds. 300 picoseconds is an incredibly small amount of time, and there is a lot that needs to happen inside the processor in each clock cycle.

q) Address State

The T1 state is called the address state because the address in the program counter (PC) is transferred to the memory address register (MAR) during this state. During the address state Ep and Lm are active; all other control bits are inactive. This means that the controller-sequencer is sending out a control word of

CON=CpEpLmCE L1E1LaEaSvEvLbLo
0 1 0 1 1 1 1 0 0 0 1 1

r) Increment State

In this state program counter is incremented so it is called incremented state(T2). This state is called increment state because the program counter is incremented. During the incremented state the controller-sequencer is sending out a control word of

CON=CpEpLmCE L1E1LaEaSvEvLbLo
1 0 1 1 1 1 1 0 0 0 1 1

s) Memory State

The addressed RAM instruction is transferred from the memory to the instruction register (T3).The only active control bits during the state CE and L1. This means that the controller-sequencer is sending out a control word of

CON=CpEpLmCE L1E1LaEaSvEvLbLo
0 0 1 0 0 1 1 0 0 0 1 1

t) Execution Cycle

The next three states (T4, T5, and T6) are the execution cycle. The register transfer during the execution cycle depends on the particular instruction being executed. For instance, LDA 9H requires different register transfer than ADD BH. The CU passes the decoded information as a sequence of control signals to the relevant function units of the CPU to perform the actions required by the instruction such as reading values from registers, passing them to the ALU to perform mathematical or logic functions on them, and writing the result back to a register. If the ALU is involved, it sends a condition signal back to the CU.

Clock Pulse: T3-T6 (Up to T6). The result generated by the operation is stored in the main memory, or sent to an output device. Based on the condition of any feedback from the ALU, Program Counter may be updated to a different address from which the next instruction will be fetched. The cycle is then repeated.

u) For LDA Routine T1,T2,T3

During the T1 state, Ep and Lm are active; the positive clock edge midway through this state will transfer the address in the program counter to the MAR[2,8]. During the T2 state, Cp is active and the program counter is incremented on the positive clock edge. During the T3 state, CE and L1 are active; when the positive clock edge occurs, the addressed RAM word is transferred to the instruction register. The LDA execution starts with the T4 state, where Lm and E1 are active; on the positive clock edge the address field in the instruction register is transferred to the MAR. During the T5 state, CE and La are active; this means that the addressed RAM data word is transferred to the accumulator on the positive clock edge. The T6 state of the LDA routine is a nop.

Various Executions Cycle:

Add:
T4. M(W)+IR(R)+A
T5.M(R)+R+B(W)
T6.A(R) B ADD +A(W)
T7.NOP

The next three states (T4, T5, and T6) are the execution cycle. The register transfer during the execution cycle depends on the particular instruction being executed. For instance, ADD 9H requires different register transfer than ADD BH.

Sub:
T4. M(W)+IR(R)+A
T5.M(R)+R-B(W)
T6.A(R) B SUB+A(W)
T7.NOP

The next three states (T4, T5, T6 and T7) are the execution cycle. The register transfer during the execution cycle depends on the particular instruction being executed. For instance, SUB 9H requires different registers transfer than Sub B

Inc:
T4. IR(L,R), M(W), CO(on), temp
T5.M(R), R(on), A(W), CO(1)
T6.A(R)+ ADD+CO(R)+A
T7.temp+ MAR(w)
T8.MAR(R)/RAM(on Add), RAM(W) +A(R)
T9.NOP

The next three states (T4, T5, T6, T7, T8 and T9) are the execution cycle. The register transfer during the execution cycle depends on the particular instruction being executed. For instance, Inc 9H requires different register transfer than Inc BH.

Load:
T4. IR(L/W), MAR (W)
T5. MAR(R), RAM(active), A(W)
T6. NOP

The next three states (T4, T5, and T6) are the execution cycle. The register transfer during the execution cycle depends on the particular instruction being executed. For instance, Load 9H requires different registers transfer than Load BH.

Del:
T4. IR (L/R), MAR (W), Count reset
T5. MAR(R), RAM (W), Counter (R)
T6. NOP

The next three states (T4, T5, and T6) are the execution cycle. The register transfer during the execution cycle depends on the particular instruction being executed. For instance, Del 9H requires different register transfer than Del BH.

Jump:
T4. IR(L/R), A(W), Counter reset
T5. Counter+1
T6. A(R), Sub, Counter(R)
T7. A(R), PC(W)
T8. NOP

The next three states (T4, T5, T6, T7, and T8) are the execution cycle. The register transfer during the execution cycle depends on the particular instruction being executed. For instance, Jump 9H requires different registers transfer than Jump BH.

III. Simulation and Synthesis

By using the FPGA synthesized the modules such as the Register Transfer Level (RTL) schematic and the timing response of different program. By using Xiling we obtained the blockdiagram and RTL design.

a) Program Counter

By using Xiling software we have shown *sthe* the blockdiagram of different types of instruction. The synthesized top-level entity of program counters as shown in Fig. 3.

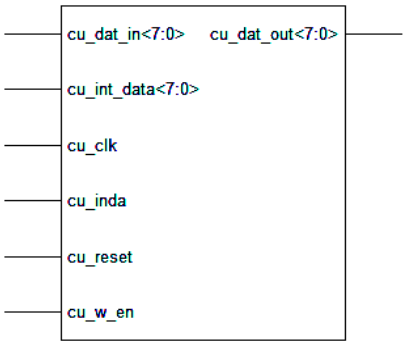

Fig.3 : Top level entity of program counter.

Fig. 3 the program counter has different input part and output depends on the input of counter, data of counter, clock, reset and enable of counter. After changing those properties, the output of program counter can be changed. Fig. 4. shows RTL Schematic of program counter which is Internal circuit level which have registers, adder-subtractors. After adding or subtracting the data stores in the registers and access from the registers.

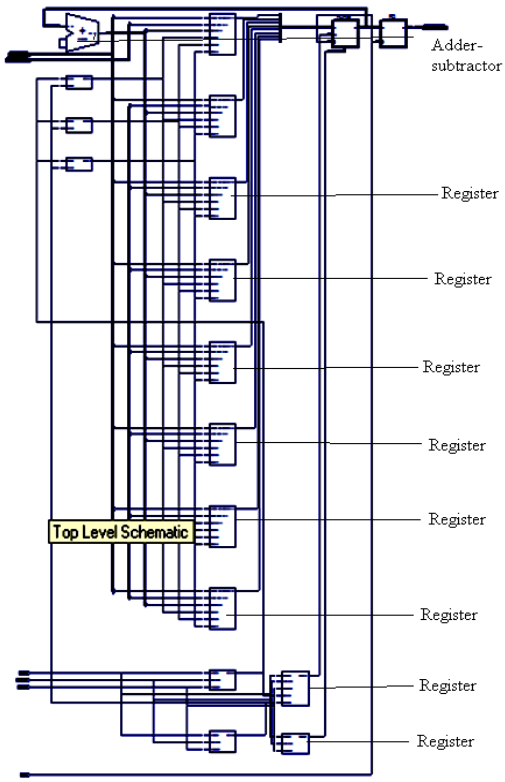

Fig.4 : RTL Schemetic of program counter(Internal part).

b) Output of Internal Counter

This block diagram Fig. 5 shows internal counter .We have designed the different parts of internal counter. We have observed the Output of internal counter will be changed due to the change the *clock* and *reset* point.

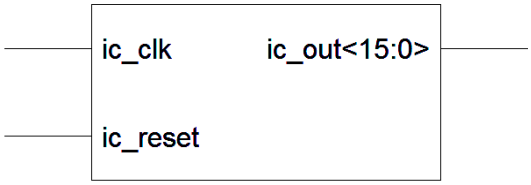

Fig.5 : Top-level entity of Internal Counter.

This is a 8 bit counter and it can count 2^8 of data we have considered. This counter is connected with MAR. We have given the following manner :
Input : 6 and Output :1[8 bit]

The simulation result or timing diagram shown in Fig. 6 we observed that for the positive edge and the negative edge the output of the internal counter can be found in *ic_out*.

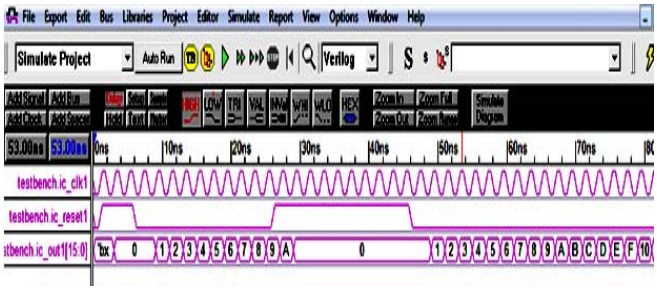

Fig.6 : Simulation results of Internal Counter.

c) MAR

8 bit input point goes to 8 bits output point. When clock is applied as an input and it goes to output. It contains the address of RAM.

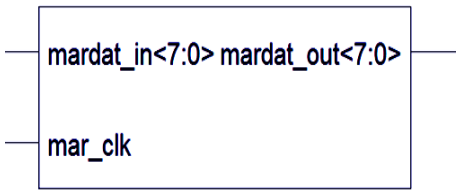

Fig.7 : Blockdiagram of MAR.

Fig. 7 shows the block diagram of MAR.The circuit level design of MAR two D-flipflpos that connect sequently as shown in Fig. 8. There is a MAR input point connected in 1st block and MAR clock that connects in both filpflops of the circuit. Due to change in MAR input and Clock the output of the MAR could be changed.

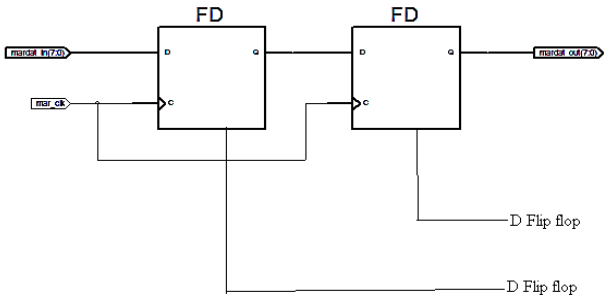

Fig.8 : RTL Schematic of MAR block.

d) Output of Register

Fig. 9 shown the block diagram of Register using Xiling software. Register that helps perform to arithmataic operation properly & its output there are register data input and clock. Output can be changed with respect to change of the property of register data input and clock that is shown in Fig. 10.

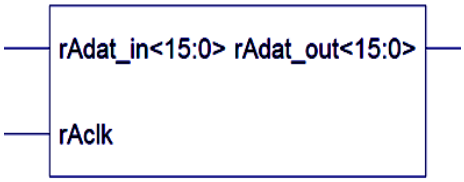

Fig.9 : Block diagram of Register.

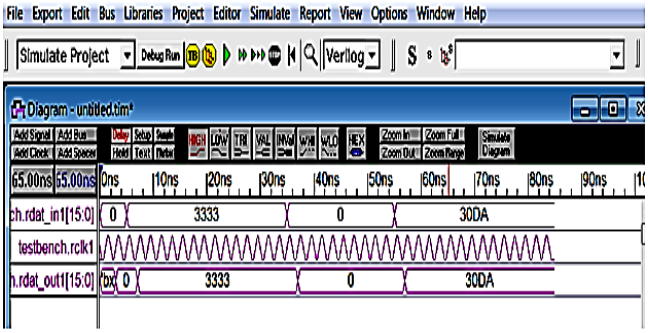

Fig.10 : Simulation results of register.

There are two register : A and B they are 16 bits. 16 bits goes to the output only when there clock is used. It is connected with W bus; input and output both goes to W bus.To transfer the data to ALU and process the data send to register from RAM. From the Verlog testbranch software we can see the change of output due to register input and register clock. For the posittive edge of clock ,the output obtained by that edge.

e) Output of Ring Counter

It controls the other blocks. It has 15 bits output and there is a ring counter in control that helps to speedup the processor. Ring counter counts from 0(H) to 13(H). When it gets the clock after 13(H) then it start from 0(H) once again. It works as output of I control, clock of other module or rest or increment of input.

In this stage if the pin is 1 then the data of RAM is loaded. If the pin is 0, the processor works with the bit of RAM.

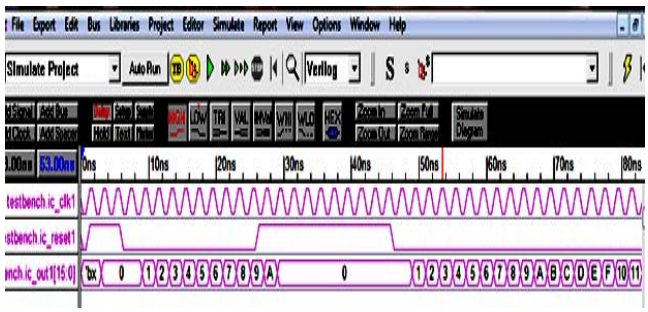

Fig.11 : Simulation result of Ring counter.

Fig. 11 shown simulation result of ring counter. Ring counter that can help to execute the instruction. By changin the clock and reset the ring counter can be changed the controller sequence. Due to change in control word the execution of instructions can be changed with the effect of ring counter. What will be the execution cycle that can be obtained by ring counter.

f) W Bus

Fig. 12 shows the RTL level of W bus by using Xiling software. From this block diagram we can see the different parts of W bus. Due to change the property of input side the output side can be changed. Here there are 16 bits inputs and outputs. While sending clock, then input goes to output. By the W bus the data can be send to one module to another module. Bus stands for the transfering the data from one module to another module. By this W bus the data can be send to another module or from another module the data can send to W bus and by this W bus the data send to another register.

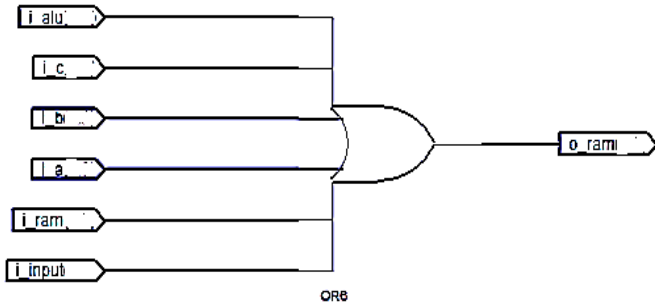

Fig.12 : RTL Schematic of W bus.

g) ALU

Fig. 13 shows the circuit level design of ALU. All the data use or load at ALU. ALU can be used as a store place of data to arithmatic operations. The circuit level design of ALU have perform with registers, buffer gates, adder-subtractors circuit. Output contains at the register get from the output of registers.

i. Instruction Register

Fig. 14 shows the block diagram of instruction register by using Xiling software. Instruction register that help the control words.There are two D-flip flpos that connect sequently.

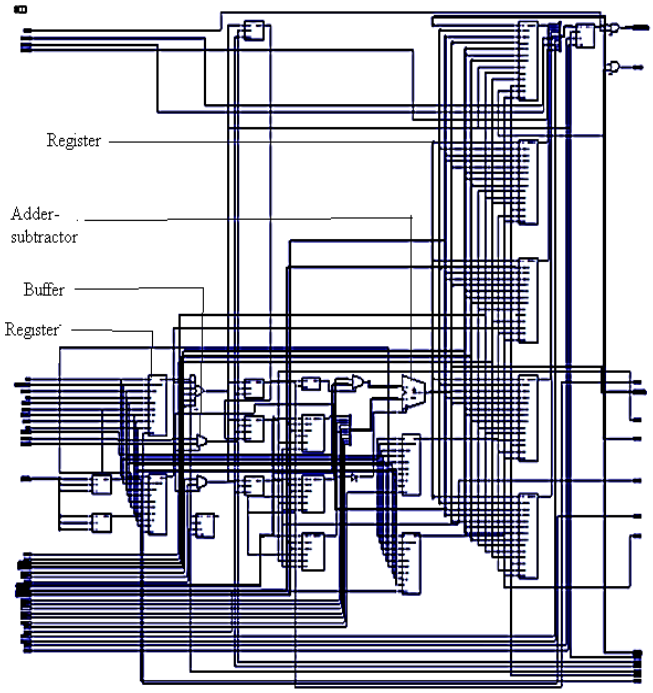

Fig.13 : Circuit level design of ALU.

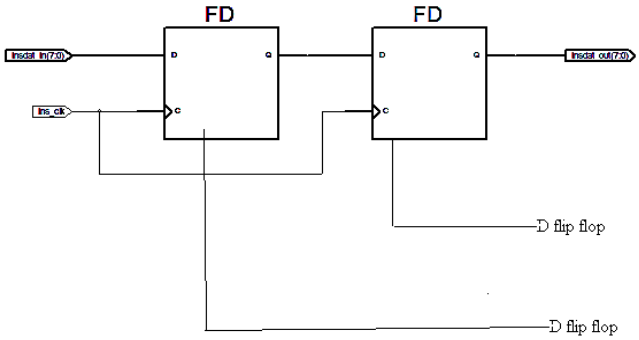

Fig.14 : Circuit level design of instruction register.

There is a Instruction Register input point that connect in *ist* block and Instruction Register clock that connects in both filpflops of the circuit. Due to change in Instruction Register input and Clock the output of the Instruction Register can be changed.

ii. TOP

A module calls Top that contains all other modules. It represents the processor. By this it shows all other module how connect the input.

iii. Output

To load the data in specific place of RAM the stage pin put in 1 and to see the value of address of RAM the stage keeps in 0. The stage pin is kept in 0 from outside of the addresss of RAM. In the RAM address the data showed as READ option as shown in Fig. 15&16.

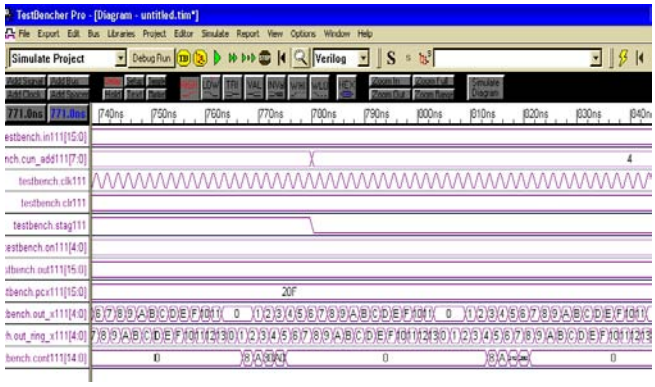

Fig.15 : Simulation results of control output of load and read.

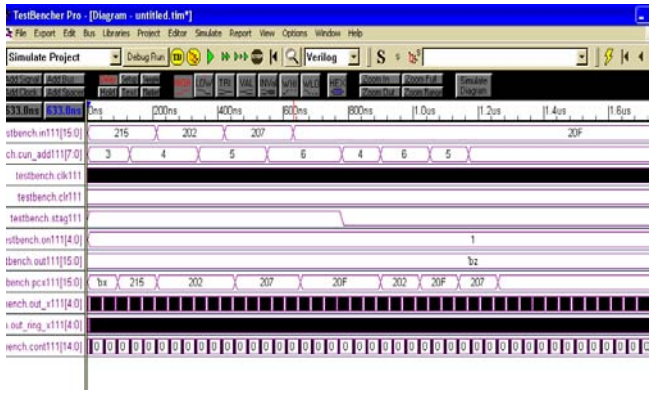

Fig.16 : Simulation results of output of load and read.

IV. Conclusion

In this paper we have used Xilinx FPGA Hardware description language that helped to show the timing response or simulation results of the different blocks. Finally we interconnected all blocks by using high label design. In the design of 16-bit microprocessor, increased RAM memory is done by using cashed memory and finally we analyzed the timing diagram for all the blocks individually. This increased memory system can be effectively used in processing large amount of information that is the special feature of this processor. There are some limitations with this microprocessor. The limitations are: First address of the RAM cannot be accessed. First bit of any module can't be defined. Due to the limitation of the software, size of the RAM is kept relatively small. We have followed that limitation and designed properly considering the limitation. In the early 1970's there were few support chips and microprocessors were programmed to perform functions that are now done by a wide variety of hardware chips. For this reason assembling a complete microprocessor-based system required both hardware and software expertise. With microprocessors, we have now entered the era of *distributed systems*. In

distributed systems, intercommunication between the number of processors is reduced to a minimum because they do not interact in real-time but exchange data words or block. This process can execute many types of instruction set. By changing control output many instruction can be accomplished. Control words describe what will be the execution function of the instructions and they work with the process of control words. In the last, to be able to interface various ancillary equipment, this microprocessor can be used as "wearing of communication".

References References Referencias

1. MALVINO: Digital Computer Electronics- An Introduction to Microcomputers- 4th Edition 2008.
2. RAMES GAONKAR: Microprocessor Architecture, Programming, and Applications with the 8085-5th Edition 2008.
3. CHINTAN PATEL: Systems Design and Programming - Second Edition 2000.
4. PROFESSOR KENNETH R. LAKER: Digital Integrated Circuits And VLSI Fundamentals-4th Edition 2008.
5. DOUGLAS HALL: Microprocessors and Interfacing, Programming and Hardware", Second Edition, Tata McGraw-hill 2008.
6. PETER ABLE: IBM PC, Assembly Language Programming- Fifth Edition 2004.
7. A.K.RAY, K.M.BHURCHANDI: Advanced Microprocessors and Peripherals, Tata McGraw Hill, 2000.
8. B.B.BREY : The Intel Microprocessors- Sixth Edition 2008.
9. Yu-Cheng Liu, Glenn A. Gibson :The 8086/8088 Family Architecture, Programming and Design- Second Edition, PHI
10. JOHN UFFENBACK : 8086/8088 Interfacing, Programming and Design", 1987, PHI
11. DEEPAK KUMAR TALA: Verilog Tutorial
12. SAMIR PALNITKAR :Verilog HDL A guide to Digital Design and Synthesis
13. J. BHASKAR:Verilog HDL synthesis a practical primer by.
14. Vivekananda Jayaram, Subbarao Wunnava, "Functional Microcontroller Design and Implementation" Florida International University, Miami, Florida, U.S.A., vjaya002@fiu.edu., subbarao@fiu.edu.
15. A Verilog HDL Primer, Star Galaxy press, Allentown, PA, 1997. ISBN-0-9656277-4-8.
16. A Verilog HDL Primer, 2nd Edition, Star Galaxy publishing, Allentown, PA, 1998. ISBN-0-9650391-9-6.
17. A VHDL Synthesis Primer, Star Galaxy press, Allentown, PA, 1996. ISBN-0-9650391-0-2.
18. A VHDL Primer: Revised edition, Prentice Hall, Englewood Cliffs Nj 1995. ISBN-0-13-181447-8.
19. A VHDL Primer, Prentice Hall, Englewood Cliffs Nj 1992. ISBN-0-13-952987-x.
20. A Guide to VHDL syntax, Prentice Hall, Englewood Cliffs Nj 1995.ISBN-0-13-324351-6. VHDL Feature and Application: Study Guide, IEEE, 1995, Order No. HL5712 .