Design and Analysis of 16-Bit Micro Processor Using Xilinx Tool

TESSY NINAN
Assistant Professor
Department of Electronics & Instrumentation Engineering
College of Engineering, Vadakara
Email:tessyninan@gmail.com

Abstract
In this project we have described the design of a 16-bit micro-processor for applications in real time embedded systems. The processor executes most of the instructions in a single machine cycle which provides high speed. The processor has been designed using VHDL and it can be implemented on an FPGA. The major components of micro-processor like ALU, comparator, shift register, control unit and registers are designed and implemented.

Keywords- FPGA, Xilinx,ASIC, VHDL, IEEE Standards.

I. INTRODUCTION
Microprocessor is basically an electronic device that consists of arithmetic and logic unit (ALU) and control circuitry which is required to function as computer’s CPU. Microprocessor is integrated circuit that interprets and executes the program instructions and behaves intelligently. The processor operates at a speed of the internal clock and the speed of the clock depends upon the no. of pulses per second. With each clock pulse, the processor performs the function that corresponds to the instruction.

16 bit microprocessor contains a number of basic modules which together completes the processor [1]. The processor uses 16 bit data bus to communicate through different sections like General purpose registers, Arithmetic logic unit, CU (control unit), memory, comparator, program counter, address register, instruction register and shift register[3]. With the advancement in integrated circuit technology the power of the processor has increased tremendously.

Microprocessors are widely used in the embedded sector based on general purpose application and special purpose application. Microprocessors are used in instruments to make it intelligent using behavioral coding. All the modules in the design are coded in VHDL, as it is very useful tool with its concept of concurrency to cope with the parallelism of digital hardware[4].

The objective of this project is like that writing a VHDL behavioral model. Developing test bench and simulating the behavior. Implementing of same on FPGA platform to validate the functionality. Synthesizing it using Xilinx ISE. In this project we used Spartan-3 family of FPGA. The Spartan®-3E family of Field-Programmable Gate Arrays (FPGAs) is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications.

II. BLOCK DIAGRAM OF PROCESSOR
A microprocessor is programmable device that accepts digital data inputs, processes it according to the instruction stored in its memory and provides results as output. It can be viewed as a programmable logic device that can be used to control processes or to turn on/off devices. The Microprocessor can be viewed as a data processing unit or a computing unit of a computer. It has computing and decision-making capability similar to that of the central processing unit of a computer. Nowadays, the microprocessor is being used in a wide range of products called microprocessor-based products or systems. The goal of project was to design a 16 bit microprocessor using VHDL language. This is a very interesting project. VHDL language is a general purpose language and this language has various features. The main aim of these project is to tie all of these features together to design a small CPU, verify its functionality that it can be synthesized.

The fig.1. shows the block diagram of a simple 16 bit microprocessor. The processor hold a number of basic pieces. There is a register array of eight 16 bit registers, there is an arithmetic and logic unit, a shifter, a program counter, an instruction register, a comparator, an address register, and a control unit. All of these blocks connected through a common 16 bit tristate data bus [5]. It has 16 bit address bus.

![Fig.1.Block diagram of processor](image-url)
A. ALU
The first VHDL component is the Arithmetic and logic unit or ALU. The ALU is the fundamental building block of the central processing unit of a computer. Depending on how the ALU is designed it can make the CPU more powerful. It performs a number of arithmetic and logical operations such as add and subtract and some logical operations such as AND, OR, and XOR.

B. Comparator
The comparator compares two values and returns a ‘1’ or ‘0’ depending on the type of comparison. The comparison type is selected by the value of input port sel. If input a and b are equal apply the value eq to port sel. If ports a and b have the same value, the port output returns ‘1’ and if the values are not equal, then output returns ‘0’.

C. Control
The control block supplies the necessary signal to make the data flow properly through the CPU and perform the expected functions. In the program the 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 has a few inputs and a lot of outputs. The control block provides all of the control signals to regulate data traffic for the CPU. The control block is a very large state machine that contains a number of states for each instruction. Executing all of the states for an instruction performs the necessary steps to complete the instruction.

D. Register
The register block is used for the address register and the instruction register. These registers are used to capture the input data on a rising edge of the clock input and drive output with the captured data. It contains three ports. Port a is the input port and q is the output port. Port clk (clock) controls when the data is stored in the register entity. When a rising edge occurs on input clock, the value of input a is assigned to output q. It also takes 1 nanosecond delay to remove delta delay problems during simulation.

E. Register Array
The regarray segment is used to form the set of registers within the CPU that are used to store intermediate values during instruction. To write a location in the regarray set input sel (select) to the location to be written, input data with the data to be written, and put a rising edge on the input clk (clock). To read a location from regarray, set input sel to the location to be read and set input en to a ‘1’; the data is output on port q.

F. Shift
The next device to be described is the shift. The shift block is used to perform shifting and rotation operations within the CPU. It has a 16 bit input bus, and a 16 bit output bus and a select input that determines which shift operation to perform. The shift entity can perform a shift left, shift right, and rotate left, and rotate right operation.

G. Triregister
The triregister is connected to the data bus and can store information from the data bus and at the same time drive information to the databus.

III. DESIGN AND VERIFICATION
The ALU inputs A and B are the two input busses upon which the ALU operations are performed. Output bus C returns the result of the ALU operation. Input select (sel) determines which of the arithmetic or logical operation is performed.

<table>
<thead>
<tr>
<th>Sel input</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>C=A</td>
</tr>
<tr>
<td>0001</td>
<td>C= A AND B</td>
</tr>
<tr>
<td>0010</td>
<td>C= A OR B</td>
</tr>
<tr>
<td>0011</td>
<td>C=NOT A</td>
</tr>
<tr>
<td>0100</td>
<td>C=A XOR B</td>
</tr>
<tr>
<td>0101</td>
<td>C=A + B</td>
</tr>
<tr>
<td>0110</td>
<td>C=A - B</td>
</tr>
<tr>
<td>0111</td>
<td>C=A + 1</td>
</tr>
<tr>
<td>1000</td>
<td>C=A-1</td>
</tr>
<tr>
<td>1001</td>
<td>C=0</td>
</tr>
</tbody>
</table>

Fig. 2. ALU Function Table

The table displays the comparison types and values. All operations work on two input values and return a single bit output. This bit is used to control the flow of operation within the processor while executing instructions. The comparator consists of a large case statement where each branch of the case statement contains an IF. If condition tested is true, a ‘1’ value is assigned; otherwise ‘0’ is assigned. Each statement occurs after 1 nanosecond to remove delta delay problems.

<table>
<thead>
<tr>
<th>Sel Input value</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>1 when a = b</td>
</tr>
<tr>
<td>NEQ</td>
<td>1 when a /= b</td>
</tr>
<tr>
<td>GT</td>
<td>1 when a &gt; b</td>
</tr>
<tr>
<td>GTE</td>
<td>1 when a &gt;= b</td>
</tr>
<tr>
<td>LT</td>
<td>1 when a &lt; b</td>
</tr>
<tr>
<td>LTE</td>
<td>1 when a &lt;= b</td>
</tr>
</tbody>
</table>

Fig. 3. Comparator operation table
IV. SOFTWARE AND TOOLS USED

Synthesis is done using Xilinx and implementation is planned on Spartan 3E FPGA kit.

A. FPGA

FPGAs are programmable semiconductor devices that are based around a matrix of Configurable Logic Blocks (CLBs) connected through programmable interconnects. As opposed to Application Specific Integrated Circuits (ASICs), where the device is custom built for the particular design, FPGAs can be programmed to the desired application or functionality requirements. Although One-Time Programmable (OTP) FPGAs are available, the dominant type is SRAM-based which can be reprogrammed as the design evolves. FPGAs allow designers to change their designs very late in the design cycle even after the end product has been manufactured and deployed in the field. In addition, Xilinx FPGAs [6] allow for field upgrades to be completed remotely, eliminating the costs associated with re-designing or manually updating electronics systems. The common blocks in an FPGA are configurable logic blocks, interconnects, selection and memory.

B. VHDL

VHDL [7] is a hardware description language. It describes the behavior of an electronic circuit or system, from which the physical circuit or system can be implemented. It is intended for circuit synthesis as well as circuit simulation. It is a standard, technology/vendor independent language, and is therefore portable and reusable. The two main immediate applications of VHDL are in the field of Programmable Logic Devices CPLDs (Complex Programmable Logic Devices) and FPGAs (Field Programmable gate Arrays), and in the field of ASICs (Application Specific Integrated Circuit). VHDL stands for VHSIC Hardware Description Language. VHSIC is itself an abbreviation for Very High Speed Integrated Circuits, an initiative funded by the United States Department of Defense in the 1980s that led to the creation of VHDL. Its first version was VHDL 87, later upgraded to the so-called VHDL 93. VHDL was the original and first hardware description language to be standardized by the Institute of Electrical and Electronics Engineers, through the IEEE 1076 standard. An additional standard, the IEEE 1164, was later added to introduce a multi-valued logic system.

V. SIMULATION AND RESULTS

The proper functioning of the processor is validated. The simulation result shows that the processor is capable of implementing the given modules in the design. After the programming have done in VHDL, a VHDL simulator is used to verify the functionality of the CPU. The VHDL RTL description of the CPU is simulated with a standard VHDL simulator. Here we have used ModelSim SE 6.5 simulator and xilinx ISE 14.2. A simulator needs two inputs, the description of the design and the stimulus to drive the design. Sometimes designs are self-stimulating and do not need any external stimulus, but in most cases VHDL designers use a VHDL test bench of one kind or another to drive the design being tested.

REFERENCES