Abstract

This paper presents the design and simple implementation of 8051 microcontroller, one of the most used microcontrollers in FPGA board from Spartan-3 family using VHDL language on Xilinx software. Instruction set such as arithmetic, logic, input and output were written and implemented and the performance test of the version have been carried out. If the microcontroller produced commercially by this design then it will be cheap and will cost low if produced in huge quantity by manufacturer.

1. Introduction

In the past years we have witnessed the area of expanding of embedded systems, which are mostly based on microcontrollers. Microcontrollers are systems on single chip. Microcontrollers have various applications in data acquisition system (DAS), process industries, industrial automation and in big projects where complexity is more. The advancement in the field of programmable...
devices such as Field programmable gate array (FPGA) has allowed manufacturers to develop high performance digital system which are highly flexible. These systems can be reprogrammed easily without reconfiguring the hardware.

The microprocessors and microcontrollers embedded within a Field Programmable Gate Arrays (FPGAs) provide the best features of both digital devices. Using microcontrollers programming control sequences or state machines in assembly or high level languages is often easier than creating similar structures in FPGAs. Customization is another advantage of embedded microcontrollers; example specific peripherals can be directly connected to microcontroller busses [1]. The design we are going to implement is an 8bit general purpose RISC processor with Harvard architecture, execute all its instructions in one clock cycle [3]

2. Background to the Study
As the microprocessor world evolved to higher end processor such as 32 bit and 64 bit architecture the demand of 8 bit processor is still maintained albeit its existence since 30 year ago. Although some observers were contemplating the death of 8 bit CPUs. The lost cost and easy operational features have made the processors still relevant throughout the years. Obviously the demands are focusing on low-end and non-critical applications. [2]

3. FPGA
FPGA was developed in mid 1970s as the first non-memory programmable logic device. It is used as AND array as well as programmable OR array. FPGA offer a number of configurable logic blocks (CLBs) which contains programmable combinational logic and register for sequential circuits there is also a set of input/output blocks which can be configured as fixed input, fixed output or bidirectional. Outputs have tri state nature and registers can be used for latching incoming or outgoing data.

FPGAs use a grid of logic gates, similar to that of an ordinary gate array, but programming is done by customer, not by manufacturer. In FPGA, all the configurable logic blocks and input/output blocks can be programmably interconnected to implement virtually any logic circuit. In larger FPGAs the configuration is volatile and must be reloaded into the device whenever power is applied or different functionality is required.
The building blocks of FPGA are:
- Logic cells (LCs) grouped into configurable logic blocks (CLBs).
- Input/output blocks
- Programmable interconnects.

3.1 Logic Cells

Each logic block in FPGA typically has a small number of inputs and outputs. A variety of FPGA products are in the market, featuring different types of logic blocks. The most commonly used logic block is a lookup table (LUT) which contains storage cells that are used to implement a small logic function. Each cell is capable of holding a single logic value either 0 or 1.

![Figure 3.2: Logic cells](image)

3.2 Input Output Blocks

Each bank can be configured individually to support a particular input/output standard. FPGA can actually be used to interface between different I/O standard. High pin count package cannot accommodate external termination registers. Thus a digitally control impedance (DCI) is employed. DCI eliminates the need of external registers and improves signal integrity.

![Figure 3.3: Input output blocks](image)
3.3 Programmable Interconnect
A programmable switch matrix form the heart of interconnects in a FPGA. The actual switching matrix employed a structure of six pass transistors per cross point. Thus connectivity can be easily established by controlling the transistors and various types of connections.

4. Microcontroller 8051
The microcontroller 8051 is one of the most popular microcontrollers in the market. It is a simple 8 bit microcontroller, which can be found as key element in various equipment and system. The microcontroller 8051 is a CISC (complex instruction set computer) it has about 100 instructions and uses a typical 12 MHz clock [5].

5. Design Description

5.1 Block Diagram Description
This system can be separated into several states as shown in Figure. Each state describes the current operation or process being performed by the CPU and is described in a VHDL module. Each module is connected via a single high-level module. The high level module is displayed in Figure 1. This system is the hardware within a computer system which carries out the instructions of a computer program by performing the basic arithmetical, logical, and input/output operations of the system.

![Block Diagram](image-url)
- **Register Set (RS):** In this information is encoded, stored, and retrieved. The RS of this system contains the following registers:
  - **IR** - holds the current instruction.
  - **PC** - holds the address of the next instruction.
  - **Load** - holds data loaded from memory.
  - **Store** - holds data being stored to memory.
- **SR** - when an operation involves two operands, the status signals are updated. The SR can also be used as an operand in arithmetic and logical operations.
- **GPR[x]** - up to 64 GPRs can be used in this architecture.
- **All GPRs and the SR can be used in any operation except for the load and store instructions. Only GPR can be used for loading and storing.**
- **Instruction Fetch Machine:** This machine fetches an instruction from Pseudo Rom, and upon completion of the instruction fetch cycle this machine signals the decoder to decode the instruction. This machine utilizes a 3-bit up counter with an active low reset. The CPU changes states and begins to decode the instruction.
- **Decoder:** Upon completion of the instruction fetch cycle, the instruction is decoded. The decoder reads bit 3 down to 0 of the IR, decides which of the sixteen operations the CPU needs to perform, and signals one of the next states to begin its operation.
- **Arithmetic logic unit:** The ALU performs arithmetic and logical operations on data. The data is taken from two GPRs and is moved to the ALU. The result is stored in a GPR. For operations that involve one operand, a GPR can be specified to store the result. The ALU supports two’s complement data.
- **Accumulator:** The accumulator will store the results of the operations performed in ALU.
- **Input output port:** The input on the input port through keys and the output is shown on LEDs.

### 6. Flowchart

![Flow Chart](image-url)

Figure 6.1: Flow Chart
7. Instruction Set

The processor with 4 bit op-code to allow instruction to perform various operations such as OR, AND, NAND, NOR, NOT, ADD, MOV, increment.

Table 1: Instruction set

<table>
<thead>
<tr>
<th>Op code</th>
<th>Function</th>
<th>Operation performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>OR</td>
<td>OR Operation of two registers</td>
</tr>
<tr>
<td>0001</td>
<td>AND</td>
<td>AND Operation of two registers</td>
</tr>
<tr>
<td>0010</td>
<td>NAND</td>
<td>NAND Operation of two registers</td>
</tr>
<tr>
<td>0011</td>
<td>NOR</td>
<td>NOR Operation of two registers</td>
</tr>
<tr>
<td>0100</td>
<td>NOT</td>
<td>NOT Operation</td>
</tr>
<tr>
<td>0101</td>
<td>ADD</td>
<td>ADD Operation of two registers.</td>
</tr>
<tr>
<td>0110</td>
<td>MOV</td>
<td>MOV Operation</td>
</tr>
<tr>
<td>0111</td>
<td>INC</td>
<td>INCREMENT Operation</td>
</tr>
</tbody>
</table>

7.1 Specification Of FPGA Kit

Spartan-3 IM Board provides an easy to use development platform for realizing various designs around SPARTAN-3 FPGA.

Features

Figure shows the SPARTAN-3, which includes the following components and features:


- **Traffic Light Control Interface Module**-16 green LEDs, 8 Red LEDs, 4 Yellow LEDS (OPTIONAL)
  - Traffic Light Interface module will be connected using 60 pin Connector (J3).
- **Analog Interface Module**- 8 bit ADC0808 and 12 bit AD7541 DAC.(OPTIONAL)
  - Analog Input – Eight channels using ADC using ADC0808, (10KSPS, 8 bit).
  - Analog Output- Two channels using Two DACs-AD7541. (12-bit, 100 ns conversion time)
- **Real Time clock Module (Optional )**
- Six-character multiplexed Seven-Segment Display
- **DIP Switches**: 16 DIP switches.
- **LEDs**: 21 onboard LEDs
  - 16 output LEDs (OL0 – OL15).
  - One Done LED.
  - 4 Power ON LEDs (LED5V, LED3V3, LED2V5, LED1V2).
- **Push Button Switches**: 16 momentary-contact push button switches in 4x4 matrix.
- LCD interface: 16 Character 2 row LCD
- **Serial Interface**: One RS-232 channel.
  - RS-232 transceiver/level translator using MAX3223 in SSOP package.
  - Uses straight-through serial cable to connect to computer or workstation serial port.
- **User selectable configuration modes** - Boundary scan, Master serial (Optional).
- **Free IOs**: 60 pin FRC Connector (J5) and 34 pin FRC Connector (J2) provided for free I/Os.
- **Clock Oscillator**: 4 MHz crystal clock oscillator.
  - **Provision** for an auxiliary crystal oscillator clock source.
- **JTAG port**: JTAG download cable (Parallel III) interface.
- **Power Supplies**: 9V regulated power supply provided along with the board.
  - On board 3.3V, 2.5V, 1.2V regulators.
  - FPGA supplies viz. Vcc int (1.2V) & Vcco (3.3V) are generated on board.

**Figure 7.1.1**: Block Diagram of Spartan 3
8. Advantages & Disadvantages

8.1 Advantages
1. Easy handling of hexadecimal number because of 8-Bit design.
2. Interfacable with large number of IC which have 8-Bit operations.
3. Can be used in variety of applications.
4. Vary high speed operation.
5. Low cost when produced in huge quantity.
7. High flexibility.

8.2 Disadvantages
1. Can’t be used where 16 and 32-bit operation is required.
2. Initial cost of production is high if not produced in large quantity.

9. Conclusion
The design have been implemented and easily seen in the Xilinx ISE design suite 8.2 as waveform. Processor has been designed and implemented in hardware on Xilinx Spartan 3 FPGA, the design has been achieved using VHDL and simulated with ModelSim. Diligent Spartan 3 development board has been used for the hardware part. This paper has presented a small and easy to understand processor developed using VHDL. Hence the implemented design is very efficient. Also the design is very flexible and cost effective if produced in huge quantity.

References
BIOGRAPHIES

Assistant professor S.D Anap (ME ETC) Department of electronics engineering, Pravara Rural Engineering College, Loni Ahmednagar (Maharashtra) India

Amir Abbas Rizvi is pursuing his Bachelor of Engineering, Department of Electronics Engineering Pravara Rural Engineering College, Loni Ahmednagar (Maharashtra) India

Javed Akhtar is pursuing his Bachelor of Engineering, Department of Electronics Engineering Pravara Rural Engineering College, Loni Ahmednagar (Maharashtra) India

Roshan Gupta is pursuing his Bachelor of Engineering, Department of Electronics Engineering Pravara Rural Engineering College, Loni Ahmednagar (Maharashtra) India