AV241 - Digital Electronics and VLSI Lab

Project Abstract

8-bit Microcontroller Design

Submitted by

Vatsalya Gupta (SC19B098) | **Rane Prathamesh Santosh (SC19B143)**

B.Tech ECE 4th Semester

Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram, Kerala, India
April 2021

1 Introduction

The proposed microcontroller design takes into consideration a very simple instruction set. It is non-pipelined (i.e., processes like decoding, fetching, execution and writing memory are merged into a single unit or a single step), and based on Harvard architecture type memory (i.e., separate memories for program and data instructions). RISC (Reduced Instruction Set Computer) is a design concept aimed at reducing the complexity of the instruction set, which in turn reduces the amount of space, cycle time, cost and other parameters considered during the design implementation. The development of FPGA has allowed the implementation of the complex logical systems on FPGA.

2 Objectives

- The aim of this project is to design the microcontroller based on 8-bit RISC architecture using Verilog.
- Design of Arithmetic Logic Unit, Control Unit, Registers, Program memory, MUX, Data memory, and Program Counter adder which are to be included in the microcontroller.
- Implementation of different instructions, and verification by simulation.

3 Design

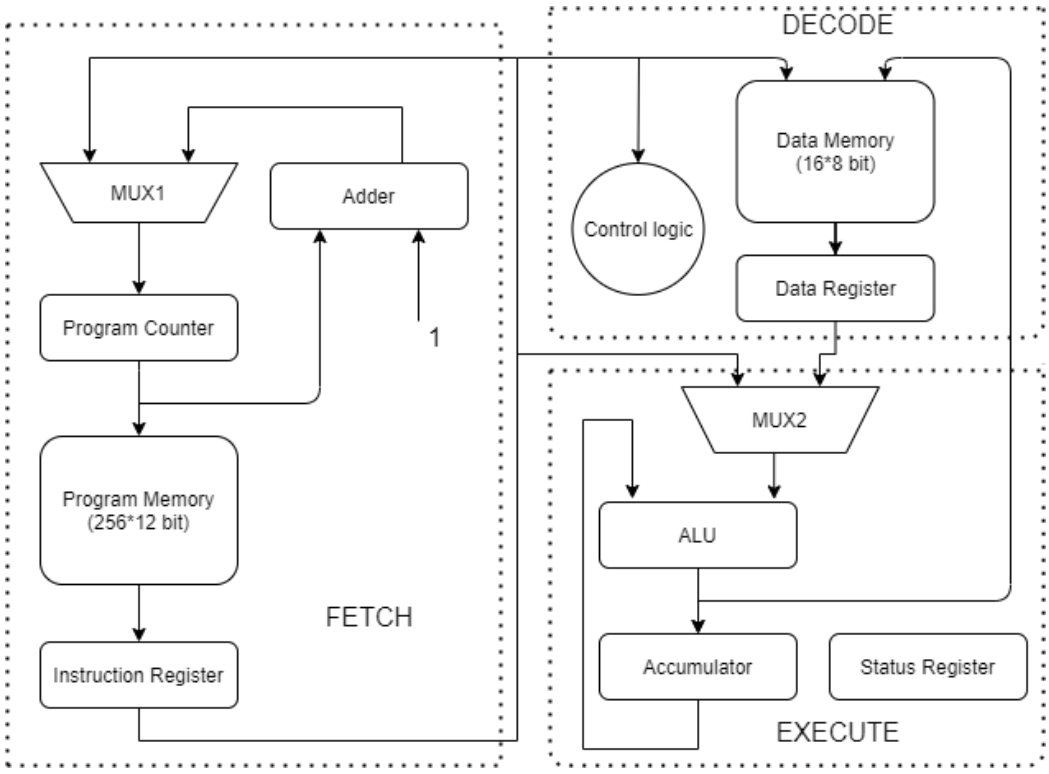

Figure 1: Architecture of the Microcontroller

The following two types of components hold programming context.

- Program counter, program memory, data memory, accumulator, status register. They are programmer visible registers and memories.
- Instruction register and data register. They are programmer invisible registers.

The following two types of components are Boolean logics that do the actual computation work. They are stateless.

- ALU, MUX1, MUX2, Adder, used as a functional unit.
- Control Logic, used to denote all control signals.

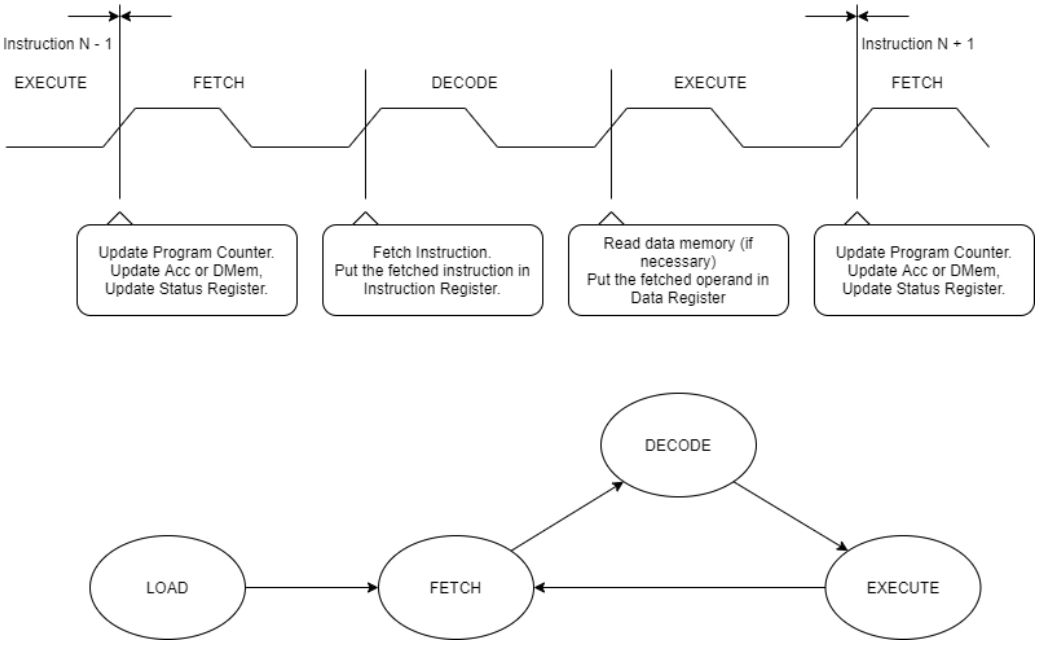

Figure 2: Timing and State Transition

There are 30 instructions in total. Each instruction is 12 bits, and needs 3 clock cycles to finish, i.e. FETCH stage, DECODE stage, and EXECUTE stage. Note that it is not pipelined. Together with the initial LOAD state, it can be considered as an FSM of 3 states (technically 4 states). States:

1. LOAD (initial state): program is loaded to program memory (one cycle per instruction).
2. FETCH (first cycle): current instruction is fetched from the program memory.
3. DECODE (second cycle): data memory is read for operand, and the instruction is decoded to generate control logic.
4. EXECUTE (third cycle): execute the following instructions
   (a) Non-branch instruction: PC = PC + 1

(b) Branch instruction: if branch is taken, PC = IR[7:0], otherwise PC = PC + 1

(c) ALU instruction: if destination is accumulator, Acc = ALU.Out, else if the result destination is data memory, DMem[IR[3:0]] = ALU.Out

(d) ALU instruction, SR = ALU.Status

SR: Status Register, PC: Program Counter, IR: Instruction Register, Acc: Accumulator.

4 Motivation

Electronic devices are being used on a larger basis in day-to-day life and are reaching almost all houses, the most common being mobile phones, laptops, ovens, security systems, and many other devices. All these devices need to be controlled in some way or other. Microcontrollers play a major role in all these control operations. They are very small and flexible. Due to their higher integration, the cost and size of the system are reduced. They are easy to use, and troubleshooting and system maintenance is straightforward. Hence, microcontroller units form a major part of today's society, and their advent has led to many technological advancements.

■ ■ ■