

ELE 3510 MidTerm Project: Single Cycle Processor

2 dedicated lab periods, 2 lab periods to complete

Objectives

- Create a full system design
- Combinational Logic Design
- Sequential Logic Design
- State Machine Design
- Memory Integration

Prelab

- None

student
check off

Assignment

This lab is to be done individually - no collaboration.

This assignment involves the development of a single cycle processor similar to the one discussed in class. You will need to design each of the processor components AND a state machine to drive the front end (FETCH) portion of the processor. You design your processor as a collection of components that are integrated at the top level (structural design).

Top Level Inputs: rstb, clk, start, start_address, instruction_count, output_select

Outputs: done, register_A,B,C,D, set

Processor Components:

256 Word, 16bit/word Instruction Memory (Mux based – see VHDL Memories, slide 4)

4 register (8 bit), Register File with 2 read ports and 1 write port

8 bit ALU with support for ADD, SUB, AND, OR, NAND, NOR, SLT, NOP

256 Word, 8 bit/word Data Memory (RAM) (FF based – see VHDL memories, slide 8)

Control Block

Support for LD, ST, and LDI memory instructions

Program Control State Machine (Sequence Selector)

Additional Components:

Clock Divider for demonstration purposes – 1Hz configuration

Output Selector

Sequence Selector:

The Sequence Selector replaces all the normal program flow control with a state machine. The selector accepts a 3 bit starting address along with a 5 bit program length value. The Sequence selector will provide addresses for the Instruction Memory starting at the modified start address,

incrementing by 1 for “program_length ” clock cycles. The selector multiplies the starting address by 32 to allow for 8 – 32 instruction programs to be accessed from the Program Memory. The program length value determines the actual program length to be executed. Program execution starts on the rising edge of clock after the Start signal is asserted and indicates completion by raising the Done signal. Each program will end with a NOP instruction. The sequencer is to stay at this instruction until reset.

Output Selector:

The Output Selector is used to present either the values for Register A and Register B to 4 7-segment displays or the values for Register C and Register D to 4 7-segment displays. This will be toggled by the user after a program has completed to see all 4 register values.

Program Memory:

You must preload your Program Memory with instructions for each of the test cases. You are responsible for encoding the instructions in our simplified instruction format. (Embed the programs into your ROM HDL to allow simulation)

Signal Mapping:

- RSTB – switch 3
- CLK – 50MHz for simulation, 1Hz for demonstration
- Start_Address – switches 2-0
- Inst_Count – switches 9-5
- Start – pushbutton 0
- Done – LED 9-0
- Output_select – switch 4
- Register A – 7 seg 0-1 (hex)
- Register B – 7 seg 2-3 (hex)
- Register C – 7 seg 0-1 (hex)
- Register D – 7 seg 2-3 (hex)
- Set – 7 seg 5 (0 or 1 displayed) (hex)

Wk 1: Your goal for week 1 should be to get each block working independently in simulation.

Wk 2: Your goal for week 2 should be to integrate all the pieces and verify full operation

Test Cases:

On the website

Processor Block Diagram:

