LAB 8: Elementary Central Processing Unit (CPU) Design

OBJECTIVES
The objective of this lab is to design and implement an elementary central processing unit (CPU) based on the ALU designed in Lab 6. A 2-bit instruction field will be used to control a simple state machine that in turn will be used to set the mux lines in the ALU according to what type of instruction is designated for execution.

MATERIALS
Proto-Board, Wires, Switches, LEDs & CPLD Board

REVIEW - LAB #6 ALU MUX Signals
The ALU designed in Lab 6 consisted of four 4:1 muxes on the inputs of REGA (A) and four 4:1 muxes on the inputs of REGB (B). The select lines for these muxes were designated MSA1:0 and MSB1:0, respectively. For a quick review, the muxes selected a bus in the following manner:

<table>
<thead>
<tr>
<th>MSA1:0 / MSB1:0</th>
<th>Bus Selected as Input to A/B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>INPUT Bus</td>
</tr>
<tr>
<td>0 1</td>
<td>A Bus</td>
</tr>
<tr>
<td>1 0</td>
<td>B Bus</td>
</tr>
<tr>
<td>1 1</td>
<td>OUTPUT Bus</td>
</tr>
</tbody>
</table>

The outputs of A and B were then passed to a combinatorial logic block and the results of this were then passed to four 8:1 muxes. The select lines for these four muxes were designated as MSC2:0. For review purposes, these (3) lines selected the following:

MSC2:0
000 => complement of A to OUTPUT Bus
001 => bit wise A AND B to OUTPUT Bus
010 => bit wise A OR B to OUTPUT Bus
011 => A Bus to OUTPUT Bus
100 => B Bus to OUTPUT Bus
101 => shift A Bus right one bit to OUTPUT Bus (0 is shifted into OUTPUT Bus[3])
110 => shift A Bus left one bit to OUTPUT Bus (0 is shifted into OUTPUT Bus[0])
111 => A Plus B Bus Plus Cin to OUTPUT Bus

INTRODUCTION – ALU CONTROLLER
A state machine controller and Instruction Register (IR) are now added to the ALU to facilitate the execution of simple instructions. See Figure 1 for the total system components. The IR register contains 2 bits which represent the following four instructions:

- 00: Move A contents => B
- 01: Load A with data on the INPUT bus
- 10: Rotate A Left 1 bit => A
- 11: Sum A Plus B => A

A flowchart for the Controller is shown in Figure 2. You are to convert the flowchart into an ASM diagram.

Instruction Register Design
The IR is a 2-bit storage register with a synchronous IR.LD input. When IR.LD = T, data is loaded into the register at the next active clock transition. When IR.LD = F, the register contents are held. This register can be simply realized with a 2:1 mux on the input of each flip-flops of the IR. When the 2:1 mux select line is low, select the IR outputs back to the IR inputs and when the select line is high, run the INPUT bus signals to the inputs of the IR.

PRE-LAB REQUIREMENTS
1. Convert the flowchart in Figure 2 into an ASM diagram; i.e., put in the actual mux select signals in the ASM diagram and make sure the timings are correct. Use the minimum number of states. Hint: use conditional outputs.
2. Create a Next State table, K-Maps and logic equations for the ALU Controller.
3. Using the graphic editor design in Quartus II, add the IR. The Controller circuitry can be implemented graphically or VHDL.
4. Simulate and test all instructions created in the Controller circuitry.

IN-LAB REQUIREMENTS
1. Download your design to the breakout board.
2. Connect LEDs and switches to the breakout board; verify it functions as specified in the Pre-Lab Requirements.
3. You need a de-bounced switch for the CLK input.

HELPFUL HINTS
Debug as you design for a much better chance of success. When something goes wrong, i.e., when a design does not work as expected, what should you do? Think of some experiments that you can do to break the problem down into pieces in order to isolate the error. A useful tool for debugging a design is to add outputs for some of the internal signals, i.e., signals that are neither outputs nor inputs of your design. This will allow you to “peer inside” a design both in simulation and with the actual hardware.

POINT BREAKDOWN
Pre-lab Materials (ASM chart, Controller design) 15%
Pre-lab Simulation Results 15%
Demonstrating a function design in lab 30%
In-lab Quiz 40%
LAB 8: Elementary Central Processing Unit (CPU) Design

Figure 1. System Components.

**Note 1:** When not specified, the default actions for each state is to “hold” REGA and REGB and OUT = REGA.

**Note 2:** Reset is used to set all registers/flip-flops to a known (zero) state.

Figure 2. Flowchart for the Controller.