
Page 1) 18 points  
Page 3) 14 points  
Page 5) 18 points  

TOTAL_ Sum → Casey out of 100

Grade Review Information: 1. Deadline of request for grade review is the day the exam is returned. 2. Do not make any changes to problems in the test as this will be considered cheating. 3. Write only in this blocked area for a re-grade request. 4. Simply write the problem number that you would like re-graded. 3 Maximum.

See the ASM Flow Chart in Appendix A to answer question 1-5.

1. Assuming that we will implement the ASM Flow Chart in Appendix A with a 128K x 8 EEPROM and JK Flip Flops, draw the complete functional block diagram for the system below. Label all signals and assume that all unused address lines will be tied to 3.3V. Note: Place the system state variables (Qn:Q0) and JK inputs in the least significant bits of the address or data bus where they are used. (10 pt.)

2. How many memory locations will need to be programmed in this design and what is the address range of these locations? Also, identify the designated part number of this device. (8 pt.)

2. Number of locations programmed in memory ___________________________ (Decimal number)

4. Range of memory to be programmed ___________________________ (Hex)

2. What is the part number of this device? ___________________________
3. Draw the complete Next State Table (inputs, state variables, outputs, etc.) below for only the rows that correspond to when the present state is State 0. Recall: JK Flip-Flops & EEPROM implementation. (12 pt.)

<table>
<thead>
<tr>
<th>W</th>
<th>Q1</th>
<th>Q0</th>
<th>Q1'</th>
<th>Q0'</th>
<th>A</th>
<th>B</th>
<th>J</th>
<th>K1</th>
<th>J</th>
<th>K0</th>
<th>JK</th>
<th>Q1</th>
<th>Q0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>1 0</td>
<td>0 X</td>
<td>1 X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
<td>1 0</td>
<td>0 X</td>
<td>1 X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
<td>0 1</td>
<td>1 1</td>
<td>0 X</td>
<td>1 X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>0 X</td>
<td>0 X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

-1 if Q1 and Q2 out of order (least)

4. Show the EEPROM memory contents (Address and Data in Hex) that needs to be programmed for the locations corresponding to State 0. Program Don’t Cares (X) in Data as Zeros. Note: Signal definitions are W.H, Z.H, A.L, B.L. (6 pt.)

**Address (Hex):**
- FFF0
- FFF4
- FFF8
- FFFC

**Data (Hex):**
- 0
- 0
- 12
- 02
- 20
- 08
- 08
- 08

5. Assuming again that we are referencing the ASM Flow Chart in Appendix A. Fill in the Voltage Timing Diagram below. Note: Signal definitions are W.H, Z.H, A.L, B.L, and C.L. Assume all devices have a delay equal to =>1 | <= and the flip-flops are falling edge triggered. Q1:0 = Present State (12 pt.)
6. In problem #5, if the logic and flip-flops are all implemented in your CPLD what will be the delays in nanoseconds for the state variables Q1:0, Unconditional Outputs and Conditional Outputs? (3 pt.)

- State Variables (Q1:0) Delay \[ \boxed{10} \text{ nsec} \]
- Unconditional Output Delay \[ \boxed{20} \text{ nsec} \]
- Conditional Output Delay \[ 10 - 20 \text{ nsec} \]

7. You are given a microprocessor with a 17 bit address bus and a 16 bit data bus. The control bus consists of a RW.H (+Read/-Write) signal and a low true data strobe (DS.L). You are given any number of 16K x 16 ROMs and 16K x 8 SRAMs. Place 20K of RAM starting at address 18000 Hex in the system memory map. Place 16K of ROM in the system memory map with the understanding that the first instruction will be fetched from address 00000 Hex.

Show the required Rom & Ram memory devices below. Label all signals and use bus nomenclature where appropriate. **Do not show the decode equations below.** (11 pt.)
8. Show the required decode logic equation for the ROM device. Note: The decode signal name should match the one that you specified/drew for your ROM memory above. (4 pt.)

\[ \text{Rom} = \overline{A_{16}} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{RW} \cdot DS \]

\[ \text{Rom also okay} \]

9. Show the required decode logic equations for the RAM devices. Your decode signal names should match those that you drew for the SRAM memory in problem #7. (6 pt.)

\[ \text{- SR0 = } \overline{A_{16}} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot DS \]

\[ \text{1/10xx/xxxx/xxxx/xxxx/xxxx} \]

\[ \text{16K} \]

\[ \text{- SR1 = } \overline{A_{16}} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot DS \]

\[ \text{1/11000/xxxx/xxxx/xxxx} \]

\[ \text{4K} \]

10. What is the address range for ROM in the memory map? \(0 - 3 \text{FFF\ }\) Hex (3 pt.)

11. What is the address range for the RAM in the memory map? \(18000 - 1BFF\) Hex (3 pt.)

12. When we remove and program the ROM what is the address should be programmed with the first instruction and where should the last program instruction be placed in the ROM? (2 pt.)

1st instruction address \(0\) Hex

Last instruction address \(3 \text{FFF\ }\) Hex

14. A student is designing a controller for a clothes dryer that has (4) low true drying sensors DS3:0. When a sensor senses moisture it outputs a H and when no moisture is detected a L is output. First, design a circuit to detect when all sensors indicate a 'dry condition'. This new signal is called DRY.L.

Show the circuit to create DRY.L (2 pt.):

Next, use DRY and another low true input Door Open (OPEN) to control the high true outputs: AIR, HEAT and SPIN. AIR turns on an air pump to inject fresh air, HEAT turns on a heater to heat the incoming air and SPIN turns on a motor that spins the main dryer drum. Inputs are: DRY, OPEN Outputs are: AIR, HEAT and SPIN

Assuming the clock has a period = 5 minutes, the controller should have the following specifications:

1. Inject and heat air for 5 minutes while spinning the clothes.
2. Next, continue drying/spinning the clothes until all (4) dry sensors show the clothes are dry.
3. Once the clothes are dry, cool for 5 minutes where the clothes spun and cooled only with fresh air.
4. At any point during the drying cycle, if the door is opened (OPEN = T), turn off the heat and stop spinning the dryer drum immediately for safety purposes. Also, extend the dry time for as long as the door is open.
13. Show the required ASM Logic Flow Diagram below. (8 pt.)

14. Given the ALU in Appendix B, Fill in the signals below to compute \((6 \times 2) + 7\) with the final answer in Reg. B. Note: Use X for unknown Register contents and X for don't care Inputs. Zero propagation delays, all answers in Hex and the solution with the least number of cycles will be awarded the most points. (10 pt.)

<table>
<thead>
<tr>
<th>clock</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Bus</td>
<td>7</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>MSA1:0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>MSB1:0</td>
<td>X</td>
<td>3</td>
<td>X</td>
<td>3</td>
</tr>
<tr>
<td>MSC2:0</td>
<td>6</td>
<td>X</td>
<td>7</td>
<td>X</td>
</tr>
<tr>
<td>REG A</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>REG B</td>
<td>X</td>
<td>X</td>
<td>C</td>
<td>X</td>
</tr>
<tr>
<td>REG C</td>
<td>X</td>
<td>C</td>
<td>X</td>
<td>3</td>
</tr>
</tbody>
</table>

\[6 \times 2 = C \quad C + 7 = 3\]
14. Given the ALU in Appendix B, fill in the signals below for \((6\times 2) + 7\) with the final answer in Register B. Note: Use \(X\) for unknown Register contents and \(X\) for don't care inputs. Zero propagation delays, all answers in Hex and the solution with the least number of cycles will be awarded the most points. (10 pt.)

<table>
<thead>
<tr>
<th></th>
<th>clock</th>
<th>Input Bus</th>
<th>MSA1:0</th>
<th>MSB1:0</th>
<th>MSC2:0</th>
<th>REG A</th>
<th>REG B</th>
<th>REG C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(X) 7 (X) (X) (X) (X)</td>
<td>(X) 0 (X) (X) (X) (X)</td>
<td>(X) 3 (X) (X) 3 (X)</td>
<td>(X) 7 (X) (X) (X)</td>
<td>6 Hex</td>
<td>0 Hex (\text{OK})</td>
<td>0 Hex (\text{OK})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[1100 + 011 = 0011\]
Appendix B. ALU with Registered Output Bus

<table>
<thead>
<tr>
<th>MSA1/MSB1</th>
<th>MSA0/MSB0</th>
<th>Bus Selected as Input to REGA/B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>INPUT Bus (Input3:0)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>REGA Bus (REGA3:0)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>REGB Bus (REGB3:0)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>REGC Bus (REGC3:0)</td>
</tr>
</tbody>
</table>

MSC2:0 (Most Significant Bit is on the left)

- 000 => complement of REGA, result in REGC
- 001 => REGA AND REGB, result in REGC
- 010 => REGA OR REGB, result in REGC
- 011 => REGA to REGC
- 100 => REGB to REGC
- 101 => shift REGA right one bit, result in REGC
- 110 => shift REGA left one bit, result in REGC
- 111 => REGA ADDED To REGB, result in REGC