LAB OBJECTIVES

1. Design a more complex state machine
2. Design a larger combination logic solution on a PLD
3. Integrate two designs into a higher level design
4. Experience using a 7-segment display device
5. Basic understanding of "tri-state" device

LAB PROCEDURE

PART 1: Design a state machine that counts both forward or backward (obviously not at the same time) through ten states. The states will represent your birthday in the following format: yyyy-mm-dd. Each state has four outputs. The beginning state, for everyone, will be the binary code for the number one or 00012. The output of state S1 will be the binary code for the number nine or 10012. The output of state S2 will be the binary code for the tens digit of the year. The output of state S3 will be the binary code for the ones digit of the year. For S4, use any number other than 0-9. This will represent the dashes in your birthday.

This idea continues. The last state, S9, will be the binary code for the ones digit of the day of your birthday.

If the counter is counting up, and if you had LEDs connected to the outputs, then you will see digit by digit the binary representation of your birthday (from left to right). If the counter is counting down, your birthday would scroll from right to left; digit-wise that is. Bring to lab a Verilog implementation of this state machine. Using the Verilog file that you created outside of lab time, compile, assign pins to the LEDs, download, test and demo your solution to the lab instructor. Note, using LEDs to display binary codes for your birthday is an important step for the next two parts.

Include your STATE DIAGRAM and all design support in your lab report.
PART 2 - Study the 7-segment display on the circuit board in lab. Are the signals to activate (turn on) individual segments on the circuit board active high or active low? If segments ‘a’ through ‘f’ are ON and ‘g’ is OFF, the number "0" will appear on the 7-segment display because the ‘g’ segment will NOT be turned ON.

Design a decoder that accepts four bits as inputs and outputs the correct segments to be turned on.

For example, 0000 should turn on all segments except segment g.
Another example, 0001, should turn on segments b and c.
Another example, 0011, should turn on segments a, b, c, d, and g.

Include in your decoder design some lower and upper case HEX displays. For example, 1010, which is a hex “A”, should turn on all segments except segment d. This will be an upper case “A”. Another, 1011, which is a hex “B”, has a problem. Turning on all segments would also display an “8”. So to comprise, display a lower case “B” as “b”. Even this is problematic because all segments on except a and g might look to be a six to some people.
Solution: make your six have a 'hat' on its top!

Do the Verilog code, compile and assign pins for the seven segment display on the XILINX_64 system. Download and test your design.

PART 3 - Connect your birthday generator state machine PLD to the inputs of your decoder PLD. The connection is accomplished by merging your two Verilog designs into a single module. Download and test your design.
PART 4

This part has nothing to do with state machines but it has everything to do with understanding computer system bus structures - the last subject in CpE 64. The topic is "tri-state", or as it is commonly called "High Impedance" or "High-Z". "Z" is a symbol used in electrical engineering for impedance.

Parts to be used to build the circuits: OPTIONAL (Instructor can modify)
- 74LS125 integrated circuit (IC)
- red and black switch (single pole single throw)
- 2 resistors (any value between 470 ohms and 1K ohms)
- 2 Light Emitting Diodes (LEDs) (any color)
- protoboard with 840 tie points

Off-Line - To understand tri-state, it is important to recall what we know about voltage levels for logic 0 and logic 1. With +5 volt logic (commonly called TTL), a 74LSxxx device must output the following:

- Output Levels Logic 1 from 2.7 volts to 5 volts (2.7 volts is the minimum)
- Output Levels Logic 0 from 0 to 0.5 volts (0.5 volts is the maximum)

- Input Levels Logic 0 will accept any input from 0 to 0.8 volts as a logic 0
- Input Levels Logic 1 will accept any input above 2.0 to 5.0 volts as a logic 1

CMOS levels will operate at lower power supply voltages.

The course lab kit contains a 74LS125, a device that contains four tri-state buffers (non-inverters). On the protoboard connect the following: pin 14 to +5 volts and pin 7 to ground. Check out a Digital Volt Meter (DVM) from your instructor using your one-card (or use the DVM from your tools kit) and prepare to make voltage measurements on the output of the buffer. Make a table to record your measurements in your report. The table should contain the voltage measurements on the output of the buffer under the following conditions: **Create a “complete” schematic of what you built for your lab report.**

<table>
<thead>
<tr>
<th>Out Enable</th>
<th>Input Logic Out</th>
<th>LEDs Connected</th>
<th>LEDs Not Connected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>V out</td>
<td>V out</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Schematic diagram](image-url)
PART 5
The outputs of tri-state devices can be tied together under the condition that only one output is active (is enabled) at any point in time. Build the circuit below. Test it by enabling each buffer, one at a time, and verifying that the output is equal to the input of the buffer that is enabled. Record your observations for your lab report. Also be sure to include your complete schematic in your report.

<table>
<thead>
<tr>
<th>Output</th>
<th>Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables</td>
<td>Inputs</td>
</tr>
<tr>
<td>E[3:0]</td>
<td>D[3:0]</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>1 0 1 1</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>0 0 1 1</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>1 1 1 0</td>
</tr>
</tbody>
</table>

PART 6
[ Has this lab become fun yet? ]
Change your design to display three characters at the same time of your birthday using all available 7-segment displays. That is, design a three character-moving window. As you scroll left or right through your birthday digits, you will see three digits at the same time.

PART 7
[ Even more fun. ]
Review the diagram for the 5x7 dot matrix display and change your design to display characters using this display instead of the 7-segment display. That is, design a character-map for displaying letters and numbers using this matrix display. Try to make your characters as “REAL” as possible.
Design a small "sprinkler" control state machine!

Design a system based on multiple state machines to make a simple control system for a single sprinkler system. Yes - like that used to irrigate lawns.

1. Use the 32 M Hz oscillator on the Spartan3E board to create a one second clock tick (by dividing down the frequency to 1 Hz.).

2. Use the pair of seven segment displays on to show time measured in seconds. You can display from 0 to 99 seconds. In a real sprinkler system you are dealing with days, hours and minutes, but not seconds.

3. Design a feature to "set" the time when you want the sprinkler output to turn on. Design a feature to then turn the sprinkler off. For example, during the demo, one might want to set the sprinkler to go on at 14 seconds and then go off at 27 seconds.

4. The challenging part for the demonstration is that the instructor must be able to "program" the On-Time and "program" the Off-Time. And, of course, your design must then work properly with those set points. This is excellent example of the practical aspects of engineering!
<table>
<thead>
<tr>
<th>Part Number</th>
<th>Requirements</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pick one part to complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Any work necessary to complete</td>
<td></td>
</tr>
</tbody>
</table>