**Finite State Machine Module: Lab Procedures**

**Goal:** The goal of this experiment is to reinforce state machine concepts by having students design and implement a state machine using simple chips and a protoboard. This experiment also introduces students to basic physical components.

**Contents:**

<table>
<thead>
<tr>
<th>Background</th>
<th>Page 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prelab</td>
<td>Page 6</td>
</tr>
<tr>
<td>Experiment</td>
<td>Page 8</td>
</tr>
<tr>
<td>Appendix</td>
<td>Page 11</td>
</tr>
</tbody>
</table>

**Background:**

The general steps to be followed for performing state machine design include:

1) Convert a description of the problem into a state transition diagram
2) Transfer the information from the state transition diagram to a state transition logic table that has inputs consisting of system inputs and current values of the state, $S_i$, and outputs consisting of system outputs and next state values, $NS_i$.
3) Design a combination circuit to implement the logic in the table.
4) Select the chips to implement the combinational circuit and to implement the memory portion of the state machine, for example, D flip flops can be used to implement registers.
5) Draw a pin diagram to illustrate how to wire the chips together to implement the state machine logic.
6) Insert the chips into a protoboard and wire the ground, high voltage, enables, and clock pins.
7) Complete the circuit by making the connections indicated from Step 5).
8) Test the circuit.

**Notation and Definitions:**

**States:** $A = 00$, $B = 01$, $C = 10$, $D = 11$ where the states are defined by the values stored in the registers; for example, State B corresponds to $S_1S_0 = 01$ where $S_1$ is the value of Register 1 and $S_0$ is the value of Register 0. The next state for values of the registers is defined by $NS_i$ for Register $i$. For example, if the current state is B and the next state is C, then $S_1S_0 = 01$ and $NS_1 = 1$ and $NS_0 = 0$.

**External input:** The external input in this circuit is denoted as “X.”

**Debounce Circuit:** A circuit to filter a chip input so that it smoothes unintended bounces in the signal.

**3 to 8 Decoder:** Signals $A_0 - A_2$ represent the inputs and $Y_0 - Y_7$ represent the outputs. The convention is that $A_2$ represents the most significant bit of a binary number and $A_0$ represents the least significant bit; for example, and input of 011 is designated as $A_2A_1A_0 = 011$.

©Bonnie H. Ferri, School of Electrical and Computer Engineering, Georgia Tech, bonnie@ece.gatech.edu
**Protoboards:** Protoboards (also known as breadboards) make many connections under the board in order to reduce the number of wires that you have to connect. Typically, groups of 5 holes are connected. There are also lines of groups of 5 that are all connected. See the example in Figure 1.

![Protoboard diagram](image)

**Integrated Chip:** An integrated chip (IC) can have several gates (circuits) on it. For example, the 74HC32 IC has 4 2-input OR gates. It has 14 wire pins coming out the sides with the connections from the pins to the logic gates shown in Figure 2.

![Pin Diagram and Logic Gates](image)
**Pin Connection Diagram:** A pin connection diagram shows the physical layout of the connections between components. Each component is shown along with the corresponding pin numbers for that component. The pin diagram for the logic expression \( F = (A \text{ OR } B) \text{ OR } (C \text{ OR } D) \) is shown below. The pin numbers correspond to the pins as labeled above for the 74HC32 2-input OR chip.

![Pin Diagram for \( F = (A+B)+(C+D) \)](image)

To build this circuit, you would need to connect a wire from pin 3 of the IC to pin 9, another wire from pin 6 to pin 10. Pins 1, 2, 4, and 5 are inputs and connect to outputs from other components while pin 8 is connected to an LED or an input to another component.

**Example:**

Consider a state machine defined by the state transition diagram in Figure 4:

![State Transition Diagram](image)

Figure 4: State Transition Diagram.

© Bonnie H. Ferri, School of Electrical and Computer Engineering, Georgia Tech, bonnie@ece.gatech.edu
Finite State Machine Lab

This state machine has the following truth table:

<table>
<thead>
<tr>
<th>State</th>
<th>S_1</th>
<th>S_0</th>
<th>X</th>
<th>New State</th>
<th>NS_1</th>
<th>NS_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>B</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>C</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>C</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>--</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

The corresponding circuit diagram is given as:

Figure 5: State machine schematic.
Using the pin layouts defined in the Appendix, a pin diagram for the state machine with schematic shown in Figure 5 is

![Pin Diagram](image)

Figure 6: Pin diagram for state machine in Figures 1 and 2. For simplicity, some of the connections, such as ground, \( V_{cc} \), clock and reset are not shown.
**Prelab:**

Prior to class, do the following:
- View the Tutorial on “Fundamental Concepts”
- Read and thoroughly understand the Example on the previous pages.
- View the video introducing the experiment
- Complete the prelab procedure given below

1) For the state transition diagram shown, complete the truth table below.

<table>
<thead>
<tr>
<th>STATE</th>
<th>S₁</th>
<th>S₀</th>
<th>X</th>
<th>NEW STATE</th>
<th>NS₁</th>
<th>NS₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2) Draw the circuit diagram for the state machine. Your circuit should include a decoder, two registers, and 2-input OR gates.

3) Draw the pin connection diagram for the state machine. Start with the basics of the pin diagram shown below, and complete it using Figure 6 as a guideline. Use the IC descriptions in the appendix to determine the proper pins to use.
Experiment:

Part A: Build the Finite State Machine defined in the Example section of this lab with the pin connection diagram shown in Figure 6.

Two registers, two pushbuttons (RESET and CLOCK), DIP Switches, a decoder, registers, and OR gates are already placed on the breadboard. The IC diagrams and pin layouts are given in the appendix.

The circuit is partially wired as shown in Figure 7:

- The battery pack supplies high voltage (Vcc) from the red lead and the ground (GND) from the black lead. The power routed through the rightmost dip switch, which must be turned on to power the circuits.
- Vcc and GND are connected to all of the ICs.
- The ICs are all oriented so that the upper right pin is PIN 1 (when the board is oriented as shown above).
- The resistors (small cylindrical components with stripes) are used to limit the current in some of the lines. Otherwise, the components may burn out too quickly.
- A “debounce” circuit, consisting of an additional IC and a capacitor, is connected to the clock button. A bounce is when a switch does not produce a clean signal. There may be some ripple in the signal (or perhaps a small spike) that might be interpreted by the digital logic as a second input signal, in this case, an extra clock pulse. The debounce circuit filters the signal to remove the ripple or spike.
**Finite State Machine Lab**

- The decoder has the enable pins already wired.

To operate the state machine:
- The light emitting diodes (LEDs) are connected to S₀ and to S₁. The LEDs light when the signal is “1”.
- The reset button resets the state to State A. It is already connected to the registers.
- The clock button advances the clock. The register stores the values of its input and passes the value to its output after each time that the clock button is pushed. The clock is already connected to the registers.

**Steps:**
1) Make sure that the power is turned off of the board prior to constructing the circuit, that is, turn off the right-most dip switch.
2) Complete the circuit according to the PIN Connection Diagram shown in Figure 6. Use the loose lead wires that are supplied with the kit. The color of wire is insignificant. Refer to the protoboard description to see which holes are connected within the protoboard, and refer to the appendix to see the pin numbers of the ICs.
   - The input to the circuit is routed from the bottom terminal of the left-most dip switch.
3) Turn on the power to the circuit.
4) Now, test the State Machine to make sure it is working properly. First reset the state machine to put it in state A. Set the input dip switch to 0 and then press the clock button. Determine the state from the LEDs and mark it in the table below. Change the input to 1 and press the clock button to see the next state value. Move the DIP switch to either the on or off position according to the table, and then press the clock button to change the state. Complete the table below to show the states corresponding to the given input sequence. Verify that this sequence follows the transitions as dictated by the state transition table in Figure 4.

<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>State</th>
<th>Input</th>
<th>State</th>
<th>Input</th>
<th>State</th>
<th>Input</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Instructor or TA Initials: ____________

**Doesn't Work?**

**Here are the top reasons:**
- The input is not connected to the dip switch terminal.
- The different chips have different numbers of pins, so pin 14 from one chip may be at a different spot than for another chip. Count the pins starting in the upper right spot (as the board is oriented in the figure).
- The wires are not aligned properly. Carefully check that they are in the correct spots, it is easy to misalign them.
Finite State Machine Lab

Part B: Build the Finite State Machine designed for the Prelab.

1) Turn off the power to the circuit before making any changes.
2) Remove only the wires that you added to the protoboard for Part A (do not remove the wires that were already there).
3) Build the circuit using the pin connection diagram that was completed for the Prelab.
4) When the state machine is complete, turn on the power. Reset the state machine to put it in state A, and input the following sequence into the circuit. Write down the state after each step:

<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>State</th>
<th>Input</th>
<th>State</th>
<th>Input</th>
<th>State</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Instructor or TA Initials: __________

When finished, pull out only the wires that you added to the circuit, and turn off the power switch.
Appendix

Decoder IC (74HC238):

Register (D-FlipFlop) IC (74HC74):

Fig. 1  Pin configuration DIP14, SO14 and (T)SSOP14.

Fig. 5  Functional diagram.
Finite State Machine Lab

2 Input OR Gate IC (74HC32):

![2 Input OR Gate IC (74HC32)](image1)

Fig. 1 Pin configuration DIP14, SO14 and (T)SSOP14.

3 Input OR Gate IC (74HC4075B1R)

![3 Input OR Gate IC (74HC4075B1R)](image2)

Fig. 3 Logic symbol.

©Bonnie H. Ferri, School of Electrical and Computer Engineering, Georgia Tech, bonnie@ece.gatech.edu