Exp#2

Prototyping of Logic Circuits using EEPROMs

COE203
Digital Logic Laboratory

Dr. Ahmad Almulhem
COE - KFUPM
Spring 2009
Digital Memory

- Read-Only Memory (ROM), Random Access Memory (RAM)
- non-volatile vs volatile
- $n$ inputs: address inputs. $b$ outputs: data outputs
- A $2^n \times b$ ROM stores the truth table of an $n$-input,$b$-output logic function
- Programmable ROM (PROM)
- Electrically Erasable PROM (EEPROM)
EEPROM Programmer
Designing circuits using EEPROMs

- Store truth table in EEPROM
Design: ROM vs SSI/MSI

+ ROM-based circuit is usually faster
+ ROM contents can easily be structured to handle unusual or undefined cases
  - No need for optimization
+ ROM function is easily modified just by changing the stored pattern, without changing any external connections.
+ Smaller

- For small circuits, a ROM-based solution will consume more power
Today's Experiment

Design a sequence recognizer circuit for the sequence 0010

- State Diagram
- State Table
- Use EEPROM to store the table
Design

(a) Block diagram

(b) Timing diagram of clock pulses
## Design

<table>
<thead>
<tr>
<th>ROM Address</th>
<th>$A_2$</th>
<th>$A_1$</th>
<th>$A_0$</th>
<th>$I/O_2$</th>
<th>$I/O_1$</th>
<th>$I/O_0$</th>
<th>ROM Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>02</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>00</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>04</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>00</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>04</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>06</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>01</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>00</td>
</tr>
</tbody>
</table>
Switch Bouncing Problem

Figure D.1: Debouncing Circuit
DATA SHEETS
## Pin Configurations

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0 - A12</td>
<td>Addresses</td>
</tr>
<tr>
<td>CE</td>
<td>Chip Enable</td>
</tr>
<tr>
<td>OE</td>
<td>Output Enable</td>
</tr>
<tr>
<td>WE</td>
<td>Write Enable</td>
</tr>
<tr>
<td>I/O0 - I/O7</td>
<td>Data Inputs/Outputs</td>
</tr>
<tr>
<td>RDY/BUSY</td>
<td>Ready/Busy Output</td>
</tr>
<tr>
<td>NC</td>
<td>No Connect</td>
</tr>
<tr>
<td>DC</td>
<td>Don’t Connect</td>
</tr>
</tbody>
</table>

### PDIP, SOIC

**Top View**
- RDY/BUSY (or NC) 1
- A12 2
- A11 3
- A10 4
- A9 5
- A8 6
- A7 7
- A6 8
- A5 9
- A4 10
- A3 11
- A2 12
- A1 13
- A0 14
- I/O7 15
- I/O6 16
- I/O5 17
- I/O4 18
- I/O3 19
- VCC 20
- WE 21
- NC 22
- OE 23
- A12 24
- A11 25
- A10 26
- A9 27
- A8 28
- A7 28
Connection Diagram

7474 (D-Flip Flop)

Order Number 54LS74DMQB, 54LS74FMQB, 54LS74LMQB, DM54LS74AJ, DM54LS74AW, DM74LS74AM or DM74LS74AN
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR</td>
<td>CLR</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>

H = High Logic Level
X = Either Low or High Logic Level
L = Low Logic Level
↑ = Positive-going Transition
* = This configuration is nonstable; that is, it will not persist when either the preset and/or clear inputs return to their inactive (high) level.
Q₀ = The output logic level of Q before the indicated input conditions were established.