Exam 2 Results

- Average = 74; Median = 78; StDev = 20 (68% within 54 and 94)
  - Q1 avg = 23.4/45; Q2 avg = 12.3/20; Q3 avg = 38.4/45

Final Lab Project

- Magnetic stripe card reader to LCD display
- Given:
  - basic schematic
  - test fixtures
- Your job:
  - design the core of the system
Overview of Magnetic Stripe Cards

- Commonly used in credit, debit, transportation, and gift cards
- Magnetic material (iron-ion rich) is contained in a plastic-like film
  - Stripe is 5.66 mm from edge of card and is 9.52 mm wide
  - Contains three tracks, each 2.79 mm wide
    - Tracks one and three are typically recorded at 8.27 bits per mm
    - Track two typically has a recording density of 2.95 bits per mm
- Various ISO standards define format
  - 7810, 7811, 7812, 7813, and 4909
  - Defined by each industry

---

Data encoded as 7-bit characters
- 6 bits for value (least significant bit first)
- 1 bit for parity

Card Data Format - Track 1

<table>
<thead>
<tr>
<th>Track</th>
<th>Recording Density (Bits per inch)</th>
<th>Character Configuration (Including parity bit)</th>
<th>Information Content (Including control characters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0110°</td>
<td>1 ABA</td>
<td>7 bits per character</td>
<td>70 alphanumeric characters</td>
</tr>
<tr>
<td>0110°</td>
<td>2 ABA</td>
<td>5 bits per character</td>
<td>40 numeric characters</td>
</tr>
<tr>
<td>0110°</td>
<td>3 ABA</td>
<td>3.375 bits per character</td>
<td>107 numeric characters</td>
</tr>
</tbody>
</table>

See [http://en.wikipedia.org/wiki/Magnetic_stripe_card](http://en.wikipedia.org/wiki/Magnetic_stripe_card) for details
Reader serial data format

- 3 signals
  - RCP – “clock”
    - RCP only oscillates if card is moving
  - RDP – data
  - CLS – card “present” indicator
    - CLS is only active if a card is present
- Decoding
  - Use RCP falling transition to sample RDP only when CLS is asserted

Block diagram

- Major components
  - Reader outputs (simulation test fixture)
  - LCD controller (Lab 8)
  - Reader buffer, decoder, and serial-to-parallel converter (Lab 9)
  - LCD display (simulation test fixture)
LCD interface

- Eleven signal wires plus PWR/GND/V₀
  - 1 mode input
  - 1 read/write control
  - 1 enable
  - 8 data lines (bi-directional)

<table>
<thead>
<tr>
<th>Operation</th>
<th>RS</th>
<th>DB7...DB0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear Display</td>
<td>0</td>
<td>0000 0001</td>
</tr>
<tr>
<td>Function Set</td>
<td>0</td>
<td>0011 0011</td>
</tr>
<tr>
<td>Display On</td>
<td>0</td>
<td>0000 1100</td>
</tr>
<tr>
<td>Entry Mode Set</td>
<td>0</td>
<td>0000 0110</td>
</tr>
<tr>
<td>Write Character</td>
<td>1</td>
<td>DDDD DDDD</td>
</tr>
</tbody>
</table>

Basic LCD operations

- Requires sequence of 4 instructions/commands on initialization (RS = 0)
  - Command write (RW = 0) – data bus pins carry command code
- Many more instructions/commands (RS = 0)
  - E.g., backup cursor, blink, etc. (look up appropriate DB values)
- Printing a character to the display (RS = 1)
  - Data write (RW = 0) – data bus pins carry character to display
- Read busy signal (on DB7)
  - LCD uses it to force a wait
  - RW = 1
  - Need to make sure not driving data lines (DB = 8'bzzzzzzzz)
  - Use DB7 as input, check if 0 (not busy) or 1 (busy)
Timing details

![Timing Diagram]

<table>
<thead>
<tr>
<th>WRITE OPERATION</th>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Cycle Time</td>
<td>$t_{\text{en}}$</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Enable Pulse Width</td>
<td>PW E</td>
<td>460</td>
<td></td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Enable Rise/Fall Time</td>
<td>$t_{\text{rise}}$</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Address Set-up Time</td>
<td>$t_{\text{as}}$</td>
<td>149</td>
<td></td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Address Hold Time</td>
<td>$t_{\text{ah}}$</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Data Set-up Time</td>
<td>$t_{\text{ds}}$</td>
<td>195</td>
<td></td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Data Hold Time</td>
<td>$t_{\text{dh}}$</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

ASCII codes

![ASCII Codes Table]
### Purpose of the project

- Learn how to build a complete system that does something useful
- Read data sheets
- Use communicating state machines
- Use test fixtures and read some more complex Verilog code