Lecture 9:
ROMs, PLAs and PALs, Oh My

CSE 370, Autumn 2007
Benjamin Ylvisaker

Where We Are

- Last lecture: Combinational Verilog
- This lecture: ROMs, PLAs and PALs, oh my
- Next lecture: Quine-McCluskey Minimization
- Homework 3 due Wednesday
- Read lab 3
Discrete Gates are a Pain

• Building circuits with discrete chips, like what you’ve done in lab, is only tractable up to modestly sized circuits

• Alternatives:
  • ROMs
  • PLAs
  • PALs
  • FPGAs (later)

Read-Only Memories

• Direct implementation of a truth table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F0</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

ROM
8 words x 4 bits/word

A B C F0 F1 F2 F3
address
outputs

block diagram
ROM Implementation

- “words”
- “word-size”
- “address”

![Diagram showing ROM implementation with a 2^n-1 decoder and bit lines connected to word lines controlled switches.]

ROM Pros and Cons

- **Pros**
  - Simple to use (just build a truth table for your functions)
  - Guaranteed to be able to implement any functions of a given number of inputs

- **Cons**
  - Large: Size is proportional to $2^\#$ of inputs
  - Inefficient: can’t exploit don’t cares or “structure” of a function
PLAs and PALs

• “Programmable Logic Array” and “Programmable Array Logic”

Program with Wire Connections

\[
\begin{align*}
F_0 &= A + B'C' \\
F_1 &= AC' + AB \\
F_2 &= B'C' + AB \\
F_3 &= B'C + A
\end{align*}
\]
Drawing PLA “Programs”

Before Programming

After Programming

F0 = AB + A'B'
F1 = CD' + C'D

Bigger PLA Example

F1 = ABC
F2 = A + B + C
F3 = A' B' C'
F4 = A' + B' + C'
F5 = A xor B xor C
F6 = A xnor B xnor C

A B C | F1 F2 F3 F4 F5 F6
0 0 0 | 0 0 1 1 0 0
0 0 1 | 0 1 0 1 1 1
0 1 0 | 0 1 0 1 1 1
0 1 1 | 0 1 0 1 0 0
1 0 0 | 0 1 0 1 1 1
1 0 1 | 0 1 0 1 0 0
1 1 0 | 0 1 0 1 0 0
1 1 1 | 1 1 0 0 1 1
The Difference Between PLAs and PALs

- **PLAs**
  - Programmable AND and OR
- **PALs**
  - Only programmable ANDs
  - ORs hard-wired

BCD to SSD Converter

BCD to 7-segment control-signal decoder

\begin{tabular}{ccccccc}
\hline
\text{c5} & \text{c4} & \text{c3} & \text{c2} & \text{c1} & \text{c0} \\
\hline
\text{c5} & \text{c4} & \text{c3} & \text{c2} & \text{c1} & \text{c0} \\
\end{tabular}

\begin{tabular}{cccccccc}
\hline
\text{A} & \text{B} & \text{C} & \text{D} \\
\hline
\text{0} & \text{0} & \text{0} & \text{0} \\
\text{0} & \text{0} & \text{0} & \text{0} \\
\end{tabular}
BCD to SSD Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>f0</th>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
<th>f5</th>
<th>f6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

PLAs and PALs

- **Pros**
  - More efficient than ROMs for many “realistic” functions
  - Still quite flexible

- **Cons**
  - Cannot implement all functions of N inputs
  - Requires logic minimization for efficiency
Additional PLA Features

- Feedback terms
  - Used to implement multi-level logic
- Registered outputs
  - Used to implement memory/sequential logic
- Tri-state outputs
  - Used to allow multiple circuits to share a single output wire (very carefully)

Thank You for Your Attention

- Read lab 3
- Continue homework 3
- Continue reading the book