Multi-level circuits. Design Examples.  
(Class 5.1 – 2/12/2013)

CSE 2441 – Introduction to Digital Logic
Spring 2013
Instructor – Bill Carroll, Professor of CSE
Today’s Topics

• Multi-level circuits
  – Fan-in constraints
  – Factoring
• Design examples
Synthesis of Combinational Logic Circuits (11)

- Fan-in – the number of gate input lines (ports)
- Circuits with more than two levels are often needed due to fan-in, i.e., max number of input nodes, constraints of gates.

![Diagram](image)
Synthesis of Combinational Logic Circuits (12)

- **Factoring**
  - A technique to obtain higher-level forms of switching functions.
  - Higher-level forms:
    - May need less hardware
    - May be used when there are fan-in constraints
    - More difficult to design
    - Slower
- **Example 2.39:**
  \[ f(A, B, C, D) = AB + AD + AC = A(B + D + C) = A(BCD) \]

(a) Original form

(b) After factoring
Example 2.40: Implement $f(a,b,c,d) = \Sigma m(8,13)$ with only two-input AND and OR gates.

- Write the canonical SOP form:
  
  $$f(a,b,c,d) = \Sigma m(8,13) = ab'c'd' + abc'd$$  
  
  Two four-input AND gates and one two-input OR gate are needed.

- Apply factoring:

  $$f(a,b,c,d) = a\bar{b}\bar{c}\bar{d} + ab\bar{c}d = (a\bar{c})(bd + \bar{b}\bar{d})$$

![Logic Circuit Diagram]
Synthesis of Combinational Logic Circuits (14)

- Convert the previous circuit to an all NAND realization.
Test Your Understanding

Implement the following function with only NAND2 gates.

\[ f_{on} (A,B,C) = \Sigma m(2,3,4,5,7) \]

Assume all literals are available as inputs.
Test Your Understanding – Self-Check

\[ f_{on}(A,B,C) = \Sigma m(2,3,4,5,7) \]
\[ = A'B + AB' + AC \]
\[ = A'B + A(B' + C) \]
Design Example

The vault area of Unsecure Bank and Trust is secured by a lock that can be opened by bank officials according to the following protocol.

- **During business hours** – the bank president (P) or both of the vice presidents (VP1, VP2).
- **Off hours** – the president and either vice president.

Design a combinational logic circuit that will UNLOCK the lock when the appropriate combination of officials enter ID codes. Assume that logic 1 indicates that the proper code has been entered by the respective officer and that logic 0 means that the proper code was not entered. The variable OPEN=1 indicates that the bank is open for business.
Design Example – Develop a Truth Table

<table>
<thead>
<tr>
<th>OPEN</th>
<th>P</th>
<th>VP1</th>
<th>VP2</th>
<th>UNLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</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>
</tr>
<tr>
<td>0</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>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Design Example – Derive Logic Equations

UNLOCK = \( \Sigma m(5,6,7,11,12,13,14,15) \)  
\[ \text{[Minterm list]} \]

\[
= O' \cdot P \cdot VP1' \cdot VP2 + O' \cdot P \cdot VP1 \cdot VP2' + O' \cdot P \cdot VP1 \cdot VP2 \\
+ O \cdot P' \cdot VP1 \cdot VP2 + O \cdot P \cdot VP1' \cdot VP2' + O \cdot P \cdot VP1' \cdot VP2 \\
+ O \cdot P \cdot VP1 \cdot VP2' + O \cdot P \cdot VP1 \cdot VP2 \\
\text{[CSP]} \]

\[
= P \cdot VP1 + P \cdot VP2 + O \cdot P + O \cdot VP1 \cdot VP2 \\
\text{[MSP]} \]

\[
= P(VP1+VP2) + O(P+VP1 \cdot VP2) \\
\text{[Factored MSP]} \]
Design Example – Two-level Realizations

• **AND/OR circuit**

  ![AND/OR Circuit Diagram](image)

  • Chips needed – SN7408, SN7411, 3xSN7432.

• **NAND/NAND circuit**

  ![NAND/NAND Circuit Diagram](image)

  • Chips needed – SN7400, SN7420.

Maximum $t_{pd} = 2\Delta$ where $\Delta$ is the gate propagation delay.
Example – Multi-level Realizations

- AND, OR gates
  - Max $t_{pd} = 4\Delta$
  - Chips needed
    - SN7408
    - SN7432

- All NAND gates
  - Max $t_{pd} = 4\Delta$
  - Chips needed
    - 2xSN7400
Design a circuit to distinguish BCD digits $\geq 5$ from those $< 5$

(a) Logic circuit

<table>
<thead>
<tr>
<th>ABCD</th>
<th>Minterm</th>
<th>$f(A, B, C, D)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>1010</td>
<td>10</td>
<td>d</td>
</tr>
<tr>
<td>1011</td>
<td>11</td>
<td>d</td>
</tr>
<tr>
<td>1100</td>
<td>12</td>
<td>d</td>
</tr>
<tr>
<td>1101</td>
<td>13</td>
<td>d</td>
</tr>
<tr>
<td>1110</td>
<td>14</td>
<td>d</td>
</tr>
<tr>
<td>1111</td>
<td>15</td>
<td>d</td>
</tr>
</tbody>
</table>

(b)
Design a circuit to distinguish BCD digits $\geq 5$ from those $< 5$ (con’t)

$$f(A,B,C,D) = \sum m(5,6,7,8,9) + d(10,11,12,13,14,15) = A + BC + BD$$
Design a circuit to distinguish BCD digits $\geq 5$ from those $< 5$ (cont’d)

$f(A,B,C,D) = A + BC + BD$

$= A + B(C + D)$