Combinational Logic Circuit Synthesis
(Class 4.2 – 2/7/2013)

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

• Two- and three-level circuit design (synthesis)
  – AND/OR and NAND/NAND circuits
  – OR/AND and NOR/NOR circuits

• Timing diagrams and propagation delay
Synthesis of Combinational Logic Circuits (1)

- **Two- and three-level Circuits**

  Input signals pass through two levels of gates before reaching the output.
Synthesis of Combinational Logic Circuits (2)

- **AND-OR and NAND Networks**
  - Given a switching expression in SOP form.
  - Example
    \[ f_\delta(p, q, r, s) = p\overline{r} + qrs + \overline{ps} \]

  \[ f_\delta(p, q, r, s) = \overline{p\overline{r}} + qrs + \overline{ps} \]
  \[ = \overline{p \cdot qrs \cdot \overline{ps}} \]
  \[ = x_1 \cdot x_2 \cdot x_3 \]

  where \( x_1 = \overline{p\overline{r}}, \ x_2 = qrs, \ \text{and} \ x_3 = \overline{ps} \]
Synthesis of Combinational Logic Circuits (3)

Find a minimum **AND-OR** implementation of $f_\phi (X,Y,Z) = \Sigma m(0,3,4,5,7)$

1. $f_\phi (X,Y,Z) = \Sigma m(0,3,4,5,7)$
2. Find MSOP using a K-map or other methods

\[
\begin{array}{c|cccc}
 XY & 00 & 01 & 11 & 10 \\
 \hline
 Z & 0 & 0 & 0 & 1 \\
 & 1 & 0 & 1 & 1 \\
\end{array}
\]

$\therefore f_\phi (X,Y,Z) = Y'Z' + YZ + XZ$

3. Realize each product term with AND gates and the OR operations with an OR gate.
Synthesis of Combinational Logic Circuits (4)

- **Example 2.37**: NAND implementation of \( f_\phi(X, Y, Z) = \Sigma m(0, 3, 4, 5, 7) \)
  1. \( f_\phi(X, Y, Z) = \Sigma m(0, 3, 4, 5, 7) \)
  2. \( f_\phi(X, Y, Z) = m_0 + m_3 + m_4 + m_5 + m_7 \)
     \[ = \overline{X}YZ + \overline{X}YZ + XYZ + \overline{X}Y + XYZ \]
  3. \( f_\phi(X, Y, Z) = \overline{YZ} + YZ + XZ \) \hspace{1cm} [T6(a)]
  4a. \( f_\phi(X, Y, Z) = \overline{YZ} + \overline{Y}Z + \overline{X}Z \) \hspace{1cm} [T3]
      or
  4b. \( f_\phi(X, Y, Z) = \overline{YZ} + \overline{YZ} + XZ \) \hspace{1cm} [T3]
      \[ = \overline{YZ} \cdot \overline{YZ} \cdot XZ \] \hspace{1cm} [T8(a)]
Synthesis of Combinational Logic Circuits (5)

(a) NAND Implementation

\[ f(X, Y, Z) \]

Diagram:

- NAND gates for \( f(X, Y, Z) \)
- Additional NAND gates for \( f(\phi(X, Y, Z)) \)
Synthesis of Combinational Logic Circuits (6)

- **OR-AND and NOR Networks**
  - Switching expression must be in POS form.
  - Example: \( f_e(A, B, C, D) = (\overline{A} + B + C)(B + C + D)(\overline{A} + D) \)

\[
\begin{align*}
f_e(A, B, C, D) &= (A + B + C)(B + C + D)(A + D) \\
&= \overline{A} + B + C + B + C + D + \overline{A} + D \\
&= y_1 + y_2 + y_3
\end{align*}
\]

where \( y_1 = \overline{A} + B + C \), \( y_2 = B + C + D \), and \( y_3 = \overline{A} + D \)
Synthesis of Combinational Logic Circuits (7)

Find a minimum **OR-AND** implementation of $f_{\phi}(X, Y, Z) = \Sigma m(0,3,4,5,7)$

1. $f_{\phi}(X, Y, Z) = \Sigma m(0,3,4,5,7)$
2. Find MPOS using a K-map or other methods

$$
\begin{array}{cccc}
X & Y & Z \\
00 & 01 & 11 & 10 \\
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 1 \\
\end{array}
$$

$f_{\phi}(X, Y, Z) = (Y' + Z)(X + Y + Z')$

3. Realize each sum term with OR gates and the AND operations with an AND gate.
Synthesis of Combinational Logic Circuits (8)

Example 2.38: NOR implementation of \( f_\phi (X,Y,Z) = \Sigma m(0,3,4,5,7) \)

1. \( f_\phi (X,Y,Z) = \Sigma m(0,3,4,5,7) = \prod M(1,2,6) \)

2. \( f_\phi (X,Y,Z) = M_1 \cdot M_2 \cdot M_6 \)
   \[= (X+Y+Z')(X+Y'+Z)(X'+Y'+Z)\]

3. \[= (X+Y+Z')(Y'+Z)\] [T6(b)]

4a. \[= ((X+Y+Z')')(Y'+Z')'\] [T3]

or

4b. \[= [[[X+Y+Z'](Y'+Z)]']'\] [T3]
   \[= [(X+Y+Z')' + (Y'+Z)']'\] [T8(b)]
   \[= [X'Y'Z + YZ']'\] [T8(a)]
Synthesis of Combinational Logic Circuits (9)

NOR implementation of $f_\emptyset$

(b)
Synthesis of Combinational Logic Circuits (10)

• **AND-OR-invert Circuits**
  – A set of AND gates followed by a NOR gate.
  – Used to readily realize two-level SOP circuits.
  – 7454 circuit:
    \[ F = AB + CD + EF + GH \]

![Diagram of 7454 circuit](a) 7454 circuit package (top view)  
![Diagram of 7454 used as a 4-to-1 multiplexer](b) 7454 used as a 4-to-1 multiplex
Synthesis of Combinational Logic Circuits (11)

AND-OR-Invert implementation of $f_\phi(X,Y,Z) = \Sigma m(0,3,4,5,7)$

$= \prod M(1,2,6)$

$= (X+Y+Z')(X+Y'+Z)(X'+Y'+Z)$

$= (X+Y+Z')(Y'+Z)$

$= [((X+Y+Z')(Y'+Z))']'$

$= [(X+Y+Z')'(Y'+Z)']'$

$= [(X'Y'Z + YZ')']'$
Synthesis of Combinational Logic Circuits (12)

AND-OR-Invert Implementation of $f_\emptyset(X,Y,Z)$
Timing Diagrams (1)

• The *Timing diagram* is a graphical representation of input and output signal relationships over time.
• Timing diagrams may show intermediate signals and propagation delays.
### Example 2.35: Derivation of truth table from a timing diagram

#### (a) Timing Diagram

- Circuit schematic showing the logic gates.
- Outputs $Y = f_a(A, B, C)$ and $Z = f_b(A, B, C)$.

#### (b) Timing Diagram

- Input and output waveforms for $A$, $B$, and $C$.
- Outputs $Y = f_a(A, B, C)$ and $Z = f_b(A, B, C)$.

#### (c) Truth Table

<table>
<thead>
<tr>
<th>Time</th>
<th>Inputs $ABC$</th>
<th>Outputs $f_a(A, B, C)$</th>
<th>Outputs $f_b(A, B, C)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t0</td>
<td>0 0 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t1</td>
<td>0 0 1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t2</td>
<td>0 1 0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>t3</td>
<td>0 1 1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>t4</td>
<td>1 0 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>t5</td>
<td>1 0 1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>t6</td>
<td>1 1 0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t7</td>
<td>1 1 1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Timing Diagrams (3)

- **Propagation Delay**
  - Physical characteristics of a logic circuit to be considered:
    - Propagation delays
    - Gate fan-in and fan-out restrictions
    - Power consumption
    - Size and weight
  - *Propagation delay*: The delay between the time of an input change and the corresponding output change.
  - Typical two propagation delay parameters:
    - $t_{PLH}$ = propagation delay time, low-to-high-level output
    - $t_{PHL}$ = propagation delay time, high-to-low-level output
  - Approximation:
    $$ t_{PD} = \frac{t_{PLH} + t_{PHL}}{2} $$
Timing Diagrams (4)

- Propagation delay through a logic gate

(a) Two-input AND gate

(b) Ideal (zero) delay

(c) $t_{PD} = t_{PLH} = t_{PHL}$

(d) $t_{PLH} < t_{PHL}$
Timing Diagrams (5)

- Power dissipation and propagation delays for several logic families (Table 2.7)

<table>
<thead>
<tr>
<th>Logic Family</th>
<th>Propagation Delay $t_{PD}$ (ns)</th>
<th>Power Dissipation Per Gate (mW)</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>7400</td>
<td>10</td>
<td>10</td>
<td>Standard TTL</td>
</tr>
<tr>
<td>74H00</td>
<td>6</td>
<td>22</td>
<td>High-speed TTL</td>
</tr>
<tr>
<td>74L00</td>
<td>33</td>
<td>1</td>
<td>Low-power TTL</td>
</tr>
<tr>
<td>74LS00</td>
<td>9.5</td>
<td>2</td>
<td>Low-power Schottky TTL</td>
</tr>
<tr>
<td>74S00</td>
<td>3</td>
<td>19</td>
<td>Schottky TTL</td>
</tr>
<tr>
<td>74ALS00</td>
<td>3.5</td>
<td>1.3</td>
<td>Advanced low-power Schottky TTL</td>
</tr>
<tr>
<td>74AS00</td>
<td>3</td>
<td>8</td>
<td>Advanced Schottky TTL</td>
</tr>
<tr>
<td>74HC00</td>
<td>8</td>
<td>0.17</td>
<td>High-speed CMOS</td>
</tr>
</tbody>
</table>
Timing Diagrams (6)

- Propagation delays of primitive 74LS series gates (Table 2.8)

<table>
<thead>
<tr>
<th>Chip</th>
<th>Function</th>
<th>Typical $t_{PLH}$</th>
<th>Maximum $t_{PLH}$</th>
<th>Typical $t_{PHL}$</th>
<th>Maximum $t_{PHL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>74LS04</td>
<td>NOT</td>
<td>9</td>
<td>15</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>74LS00</td>
<td>NAND</td>
<td>9</td>
<td>15</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>74LS02</td>
<td>NOR</td>
<td>10</td>
<td>15</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>74LS08</td>
<td>AND</td>
<td>8</td>
<td>15</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>74LS32</td>
<td>OR</td>
<td>14</td>
<td>22</td>
<td>14</td>
<td>22</td>
</tr>
</tbody>
</table>