CIS 410/510
Combinational Circuit Design

Prof. Michel A. Kinsky
A combinational circuit consists of logic gates whose outputs, at any time, are determined by combining the values of the inputs.

- For $n$ input variables, there are $2^n$ possible binary input combinations.
- For each binary combination of the input variables, there is one possible output.
A combinational circuit can be described by:

- A truth table that lists the output values for each combination of the input variables, or m Boolean functions, one for each output variable.

Combinational circuits are memory-less:

- Output values depend ONLY on the current input values.
Analysis Procedure

- Boolean Expression Approach

\[ F_1 = AB'C' + A'BC' + A'B'C + ABC \]
\[ F_2 = AB + AC + BC \]
Analysis Procedure

- Truth Table Approach
Binary Adder

- **Half Adder**
  - Adds 1-bit plus 1-bit
  - Produces Sum and Carry

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$C$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ x + y \rightarrow C \quad S \]

\[ x \rightarrow S \quad C \]

\[ y \rightarrow C \quad S \]
Binary Adder

- **Full Adder**
  - Adds 1-bit plus 1-bit plus 1-bit
  - Produces **Sum and Carry**

<table>
<thead>
<tr>
<th>(x)</th>
<th>(y)</th>
<th>(z)</th>
<th>(C)</th>
<th>(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 0 1</td>
<td>0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 0 1</td>
<td>0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1 1</td>
<td>0 1 1</td>
<td>1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 1 1</td>
<td>0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 0 1</td>
<td>1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 1 0</td>
<td>1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1</td>
<td>1 1 1</td>
<td>1 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
S = xy'z' + x'yz' + x'y'z + xyz = x \oplus y \oplus z
\]

\[
C = xy + xz + yz
\]
Binary Adder

- Full Adder

\[ S = x'y'z' + x'yz' + x'y'z + xyz = x \oplus y \oplus z \]
\[ C = xy + xz + yz \]
Binary Adder

- Full Adder

$x \rightarrow HA \rightarrow S$
$y \rightarrow HA \rightarrow S$
$z \rightarrow HA \rightarrow S$

$x \rightarrow HA \rightarrow S$
$y \rightarrow HA \rightarrow S$
$z \rightarrow HA \rightarrow S$
Binary Adder

\[ x_3 x_2 x_1 x_0 + y_3 y_2 y_1 y_0 \]

\[ c_y \quad S_3 \quad S_2 \quad S_1 \quad S_0 \]

FA

\[ x_3 \quad y_3 \]

\[ C_4 \quad S_3 \]

FA

\[ x_2 \quad y_2 \]

\[ C_3 \quad S_2 \]

FA

\[ x_1 \quad y_1 \]

\[ C_2 \quad S_1 \]

FA

\[ x_0 \quad y_0 \]

\[ C_1 \quad S_0 \]

0
Binary Adder

- Carry Propagate Adder

\[
\begin{align*}
x_7 & x_6 & x_5 & x_4 \\
y_7 & y_6 & y_5 & y_4 \\
A_3 & A_2 & A_1 & A_0 & B_3 & B_2 & B_1 & B_0 \\
C_y & CPA & C_0 \\
S_7 & S_6 & S_5 & S_4 \\
S_3 & S_2 & S_1 & S_0
\end{align*}
\]
Design Procedure

- **BCD-to-Excess 3 Converter**

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
<th>$w$</th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</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>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>1</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>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</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>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</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>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$w = A + B(C + D)$

$y = (C + D)' + CD$

$x = B'(C + D) + B(C + D)'$

$z = D'$
### Seven-Segment Decoder

<table>
<thead>
<tr>
<th>$wxyz$</th>
<th>$abcdefg$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>11111110</td>
</tr>
<tr>
<td>0001</td>
<td>01100000</td>
</tr>
<tr>
<td>0010</td>
<td>10111011</td>
</tr>
<tr>
<td>0011</td>
<td>11110001</td>
</tr>
<tr>
<td>0100</td>
<td>01100111</td>
</tr>
<tr>
<td>0101</td>
<td>10110111</td>
</tr>
<tr>
<td>0110</td>
<td>10111111</td>
</tr>
<tr>
<td>0111</td>
<td>11100000</td>
</tr>
<tr>
<td>1000</td>
<td>11111111</td>
</tr>
<tr>
<td>1001</td>
<td>11110111</td>
</tr>
<tr>
<td>1010</td>
<td>xxxxxxxx</td>
</tr>
<tr>
<td>1011</td>
<td>xxxxxxxx</td>
</tr>
<tr>
<td>1100</td>
<td>xxxxxxxx</td>
</tr>
<tr>
<td>1101</td>
<td>xxxxxxxx</td>
</tr>
<tr>
<td>1110</td>
<td>xxxxxxxx</td>
</tr>
<tr>
<td>1111</td>
<td>xxxxxxxx</td>
</tr>
</tbody>
</table>

**BCD code**

![Seven-Segment Decoder Diagram](image-url)
BCD Adder

Diagram showing the BCD adder circuit with inputs A3 A2 A1 A0, B3 B2 B1 B0, and carry-in Ci. The outputs are S3 S2 S1 S0, and there is an error signal (Err) connected to the multiplexers. The diagram includes two binary adders and logic gates to handle the carry and error signals.
Binary Subtractor

- Use 2’s complement with binary adder

\[ x - y = x + (-y) = x + y' + 1 \]
Binary Adder/Subtractor

- **M**: Control Signal (Mode)
  - $M=0 \implies F = x + y$
  - $M=1 \implies F = x - y$
Overflow

• **Unsigned Binary Numbers**

• **2’s Complement Numbers**

[Diagram of binary addition with overflow indicators]
Magnitude Comparator

- Compare 4-bit number to 4-bit number
  - 3 Outputs: <, =, >
  - Expandable to more number of bits

\[
\begin{align*}
x_3 &= \overline{A_3} \overline{B_3} + A_3 B_3 \\
x_2 &= \overline{A_2} \overline{B_2} + A_2 B_2 \\
x_1 &= \overline{A_1} \overline{B_1} + A_1 B_1 \\
x_0 &= \overline{A_0} \overline{B_0} + A_0 B_0 \\
(A = B) &= x_3 x_2 x_1 x_0 \\
(A > B) &= A_3 \overline{B_3} + x_3 A_2 \overline{B_2} + x_3 x_2 A_1 \overline{B_1} + x_3 x_2 x_1 A_0 \overline{B_0} \\
(A < B) &= \overline{A_3} B_3 + x_3 \overline{A_2} B_2 + x_3 x_2 \overline{A_1} B_1 + x_3 x_2 x_1 \overline{A_0} B_0
\end{align*}
\]
Magnitude Comparator

\[
\begin{align*}
A_3 & \quad x_3 \\
B_3 & \\
A_2 & \quad x_2 \\
B_2 & \\
A_1 & \quad x_1 \\
B_1 & \\
A_0 & \quad x_0 \\
B_0 & \\
\end{align*}
\]

\((A < B)\)  
\((A > B)\)  
\((A = B)\)
Magnitude Comparator

\[ x_7 x_6 x_5 x_4 \]
\[ y_7 y_6 y_5 y_4 \]

\[ A_3 A_2 A_1 A_0 \]
\[ B_3 B_2 B_1 B_0 \]

\[ I_{(A>B)} \]
\[ I_{(A=B)} \]
\[ I_{(A<B)} \]

\[ A<B \quad A=B \quad A>B \]

\[ x_3 x_2 x_1 x_0 \]
\[ y_3 y_2 y_1 y_0 \]

\[ A_3 A_2 A_1 A_0 \]
\[ B_3 B_2 B_1 B_0 \]

\[ I_{(A>B)} \]
\[ I_{(A=B)} \]
\[ I_{(A<B)} \]

\[ A<B \quad A=B \quad A>B \]
Decoders

- Extract "Information" from the code
- Binary Decoder
  - Example: 2-bit Binary Number
Decoders

- 2-to-4 Line Decoder

<table>
<thead>
<tr>
<th>$I_1$</th>
<th>$I_0$</th>
<th>$Y_3$</th>
<th>$Y_2$</th>
<th>$Y_1$</th>
<th>$Y_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</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>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$Y_3 = I_1 I_0$

$Y_2 = I_1 \bar{I}_0$

$Y_1 = \bar{I}_1 I_0$

$Y_0 = \bar{I}_1 \bar{I}_0$
Encoders

- Put "Information" into code
- Binary Encoder
  - Example: 4-to-2 Binary Encoder

<table>
<thead>
<tr>
<th>$x_3$</th>
<th>$x_2$</th>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$y_0$</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>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Encoders

- Octal-to-Binary Encoder (8-to-3)

<table>
<thead>
<tr>
<th>$I_7$</th>
<th>$I_6$</th>
<th>$I_5$</th>
<th>$I_4$</th>
<th>$I_3$</th>
<th>$I_2$</th>
<th>$I_1$</th>
<th>$I_0$</th>
<th>$Y_2$</th>
<th>$Y_1$</th>
<th>$Y_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<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>0</td>
<td>0</td>
<td>0</td>
<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>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</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>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$Y_2 = I_7 + I_6 + I_5 + I_4$

$Y_1 = I_7 + I_6 + I_3 + I_2$

$Y_0 = I_7 + I_5 + I_3 + I_1$
Multiplexers

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_0$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$I_0$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$I_1$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$I_2$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$I_3$</td>
</tr>
</tbody>
</table>
Multiplexers

- **2-to-1 MUX**

  ![2-to-1 MUX Diagram]

- **4-to-1 MUX**

  ![4-to-1 MUX Diagram]
Multiplexers

- Quad 2-to-1 MUX

\[x_3\quad y_3\quad I_0\quad MUX\quad S\quad Y\]

\[x_2\quad y_2\quad I_0\quad MUX\quad S\quad Y\]

\[x_1\quad y_1\quad I_0\quad MUX\quad S\quad Y\]

\[x_0\quad y_0\quad I_0\quad MUX\quad S\quad Y\]

\[A_3\quad A_2\quad A_1\quad A_0\quad B_3\quad B_2\quad B_1\quad B_0\]

\[S\quad E\quad Y_3\quad Y_2\quad Y_1\quad Y_0\]
Multiplexers

- Quad 2-to-1 MUX

\[
\begin{align*}
A_3 & \quad A_2 & \quad A_1 & \quad A_0 & \quad B_3 & \quad B_2 & \quad B_1 & \quad B_0 \\
\text{MUX} & \quad Y_3 & \quad Y_2 & \quad Y_1 & \quad Y_0 & \quad S & \quad \overline{E}
\end{align*}
\]
Multiplexer Expansion

- 8-to-1 MUX using Dual 4-to-1 MUX
DeMultiplexers

DeMUX

\[ I \rightarrow \begin{array}{c}
S_1 \ \ S_0 \\
Y_3 \\
Y_2 \\
Y_1 \\
Y_0 
\end{array} \]

Truth Table:

<table>
<thead>
<tr>
<th>( S_1 )</th>
<th>( S_0 )</th>
<th>( Y_3 )</th>
<th>( Y_2 )</th>
<th>( Y_1 )</th>
<th>( Y_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</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>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Three-State Gates

• Tri-State Buffer

• Tri-State Inverter

<table>
<thead>
<tr>
<th>C</th>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>Hi-Z</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Next Class

• Sequential Circuit Design