COE 202: Digital Logic Design
Combinational Circuits
Part 4

Dr. Ahmad Almulhem
Email: ahmads@hotmail.com
Phone: 860-7554
Office: 22-324
Objectives

• Magnitude comparator
  • Design of 4-bit magnitude comparator

• Design Examples using MSI components
  • Adding Three 4-bit numbers
  • Building 4-to-16 Decoders with 2-to-4 Decoders
  • Getting the larger of 2 numbers (Maximum)
  • Excess-3 Code Converter
Magnitude Comparator

**Definition:** A magnitude comparator is a combinational circuit that compares two numbers A & B to determine whether:

- \( A > B \), or
- \( A = B \), or
- \( A < B \)

**Inputs**
- First \( n \)-bit number A
- Second \( n \)-bit number B

**Outputs**
- 3 output signals (GT, EQ, LT), where:
  - \( GT = 1 \) IFF \( A > B \)
  - \( EQ = 1 \) IFF \( A = B \)
  - \( LT = 1 \) IFF \( A < B \)

**Note:** Exactly One of these 3 outputs equals 1, while the other 2 outputs are 0's
Example 1: Magnitude Comparator (4-bit)

Problem: Design a magnitude comparator that compares 2 4-bit numbers A and B and determines whether:
- A > B, or
- A = B, or
- A < B
Example 1: Magnitude Comparator (4-bit)

Solution:
Inputs: 8-bits (A \rightarrow 4\text{-bits}, B \Rightarrow 4\text{-bits})

A and B are two 4-bit numbers
Let \( A = A_3A_2A_1A_0 \), and
Let \( B = B_3B_2B_1B_0 \)

Inputs have \( 2^8 \) (256) possible combinations (size of truth table and K-map?)

Not easy to design using conventional techniques

The circuit possesses certain amount of regularity \( \Rightarrow \) can be designed algorithmically.
Example 1: Magnitude Comparator (4-bit)

Designing EQ:

Define $X_i = A_i \text{ xnor } B_i = A_i \overline{B_i} + A_i \overline{B_i}$

$\Rightarrow X_i = 1$ IFF $A_i = B_i \forall i = 0, 1, 2$ and $3$

$\Rightarrow X_i = 0$ IFF $A_i \neq B_i$

Therefore the condition for $A = B$ or $EQ=1$ IFF

$A_3 = B_3 \rightarrow (X_3 = 1)$, and
$A_2 = B_2 \rightarrow (X_2 = 1)$, and
$A_1 = B_1 \rightarrow (X_1 = 1)$, and
$A_0 = B_0 \rightarrow (X_0 = 1)$.

Thus, $EQ=1$ IFF $X_3 \cdot X_2 \cdot X_1 \cdot X_0 = 1$. In other words,

$EQ = X_3 X_2 X_1 X_0$
Example 1: Magnitude Comparator (4-bit)

Designing GT and LT:

GT = 1 if A > B:
- If $A_3 > B_3 \rightarrow A_3 = 1$ and $B_3 = 0$
- If $A_3 = B_3$ and $A_2 > B_2$
- If $A_3 = B_3$ and $A_2 = B_2$ and $A_1 > A_1$
- If $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 > B_0$

Therefore,

$$GT = A_3B_3' + X_3 A_2 B_2' + X_3 X_2 A_1 B_1' + X_3 X_2 X_1 A_0 B_0'$$

Similarly, $LT = A_3'B_3 + X_3 A_2'B_2 + X_3 X_2 A_1'B_1 + X_3 X_2 X_1 A_0'B_0$
Example 1: Magnitude Comparator (4-bit)

EQ = $X_3 \ X_2 \ X_1 \ X_0$

GT = $A_3B_3'$
+ $X_3A_2B_2'$
+ $X_3X_2A_1B_1'$
+ $X_3X_2X_1A_0B_0'$

LT = $B_3A_3'$
+ $X_3B_2A_2'$
+ $X_3X_2B_1A_1'$
+ $X_3X_2X_1B_0A_0'$

Ahmad Almulhem, KFUPM 2009
Example 1: Magnitude Comparator (4-bit)

- Do you need all three outputs?
- Two outputs can tell about the third one
  - Example: when A is NOT GREATER THAN B, and A is NOT LESS THAN B THEN A is EQUAL TO B
- Therefore, we can save some logic gates:

Ahmad Almulhem, KFUPM 2009
Example 2: Adding three 4-bit numbers

Problem: Add three 4-bit numbers using standard MSI combinational components

Solution:
Let the numbers be $X_3X_2X_1X_0$, $Y_3Y_2Y_1Y_0$, $Z_3Z_2Z_1Z_0$,

\[
\begin{align*}
X_3X_2X_1X_0 & \quad S_3S_2S_1S_0 \\
+ \quad Y_3Y_2Y_1Y_0 & \quad + \quad Z_3Z_2Z_1Z_0 \\
\hline
C_4 & \quad S_3S_2S_1S_0 \\
D_4 & \quad F_3F_2F_1F_0
\end{align*}
\]

Note: $C_4$ and $D_4$ is generated in position 4. They must be added to generate the most significant bits of the result.
Example 2: Adding three 4-bit numbers

Ahmad Almulhem, KFUPM 2009
Example 3: 4-to-16 Decoder

Problem: Design a 4x16 Decoder using 2x4 Decoders

Solution:

- Each group combination holds a unique value for \(A_3A_2\)
- One Decoder can be therefore used with inputs: \(A_3A_2\)
- Four more decoders are needed for representing each individual color combination

<table>
<thead>
<tr>
<th>(A_3)</th>
<th>(A_2)</th>
<th>(A_1)</th>
<th>(A_0)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(D_0)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(D_1)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>(D_2)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>(D_3)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>(D_4)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(D_5)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(D_6)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(D_7)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(D_8)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(D_9)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>(D_{10})</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>(D_{11})</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>(D_{12})</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(D_{13})</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(D_{14})</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(D_{15})</td>
</tr>
</tbody>
</table>
Example 3: 4-to-16 Decoder

A_0, A_1
2x4 Decoder

D_0, D_1, D_2, D_3

A_0, A_1
2x4 Decoder

D_4, D_5, D_6, D_7

A_0, A_1
2x4 Decoder

D_8, D_9, D_10, D_11

A_0, A_1
2x4 Decoder

D_12, D_13, D_14, D_15

A_2, A_3
2x4 Decoder
Example 4: The larger of 2 numbers

**Problem:** Given two 4-bit unsigned numbers, design a circuit such that the output is the larger of the two numbers.

**Solution:** We will use a magnitude comparator and a Quad 2x1 MUX. How?
Example 4: The larger of 2 numbers

A\textsubscript{0} \rightarrow \text{A}_0
A\textsubscript{1} \rightarrow \text{A}_1
A\textsubscript{2} \rightarrow \text{A}_2
A\textsubscript{3} \rightarrow \text{A}_3
B\textsubscript{0} \rightarrow \text{B}_0
B\textsubscript{1} \rightarrow \text{B}_1
B\textsubscript{2} \rightarrow \text{B}_2
B\textsubscript{3} \rightarrow \text{B}_3

4-bit Magnitude Comparator

\text{A} > \text{B} \rightarrow \text{GT}
\text{A} < \text{B} \rightarrow \text{LT}
\text{A} = \text{B} \rightarrow \text{EQ}

\text{QUAD 2X1 MUX}

Y\textsubscript{0} \rightarrow \text{Y}_0
Y\textsubscript{1} \rightarrow \text{Y}_1
Y\textsubscript{2} \rightarrow \text{Y}_2
Y\textsubscript{3} \rightarrow \text{Y}_3

\text{For S}_0 = 1, \text{A} \text{ is selected,}
\text{For S}_0 = 0, \text{B} \text{ is selected}
Example 5: Excess-3 Code Converter

**Problem:** Design an excess-3 code converter that takes as input a BCD number, and generates an excess-3 output.

**Solution:** Use decoders and encoders

<table>
<thead>
<tr>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</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>1</td>
<td>0</td>
<td>1</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>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>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</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>1</td>
<td>1</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>0</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>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Example 5: Excess-3 Code Converter

What will be the output?
Example 5: Excess-3 Code Converter

- A decoder can be used with the inputs being W,X,Y,Z
  - It will be a 4x16 decoder, with only a single output bit equal to 1 for any input combination
- An encoder (16x4) will take as input the 16 bit output from the decoder, and will generate the appropriate output in excess-3 format
- For this to function correctly, the output from the decoder must be displaced 3 places while being connected to the encoder input
- It may be noted that outputs 10,11,12,13,14,15 of the decoder are not used – since we are dealing with BCD
Summary

• Design = Different possibilities
• Better designer = more practice
• More design examples in the textbook