King Fahd University of Petroleum & Minerals
Computer Engineering Dept

COE 202 – Fundamentals of Computer Engineering

Term 062

Dr. Ashraf S. Hasan Mahmoud

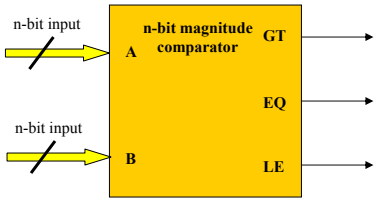
Rm 22-148-3

Ext. 1724

Email: ashraf@ccse.kfupm.edu.sa

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:
    1. GT = 1 IFF A > B
    2. EQ = 1 IFF A = B
    3. LT = 1 IFF A < B
- Note: Exactly One of these 3 outputs equals 1, while the other 2 outputs are 0`s

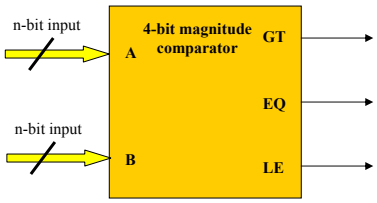


Design Example 1: Magnitude Comparator

- **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

4-bit Magnitude Comparator

- Inputs: 8-bits (A ⇒ 4-bits , B ⇒ 4-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 ⇒ *can be designed algorithmically.*

4-bit Magnitude Comparator: Design of the EQ Output

- When two binary numbers are EQUAL?
- Define $X_i = A_i$ XNOR $B_i = A_i B_i + A_i' B_i'$
  - ➔ $X_i = 1$ IFF $A_i = B_i$ $\forall$ i =0, 1, 2 and 3
  - ➔ $X_i = 0$ IFF $A_i \neq B_i$
- Therefore the condition for A = B or EQ=1 IFF
  1. A3=B3 → (X3 = 1), and
  2. A2=B2 → (X2 = 1), and
  3. A1=B1 → (X1 = 1), and
  4. A0=B0 → (X0 = 1).
- Thus, EQ=1 IFF X3 X2 X1 X0 = 1. In other words,
  EQ = X3 X2 X1 X0

4-bit Magnitude Comparator: Design of the GT Output

- When can you say that A is GREATER THAN B?
- **Case 1: If A3 > B3**, then A > B (GT=1)
  - Irrespective of the relative values of the other bits of A & B.
  - Example, A = 1000 and B = 0111 where A > B.
- How to represent A3 > B3 using logic?
  - This can be stated as GT=1 if A3 B3' =1
- **Case 2: If A3 = B3 (i.e. X3 = 1)**, we compare the next significant pair of bits (A2 & B2)
- If A2 > B2 then A > B (GT=1) irrespective of the relative values of the other bits of A & B.
  - Example, A = 0100 and B = 0011 where A > B.
- How to represent this case using logic?
  - This can be stated as GT=1 if X3 A2 B2' =1

4-bit Magnitude Comparator: Design of the GT Output – cont'd

- **Case 3: If A3 = B3 (X3 = 1) and A2 = B2 (X2 = 1)**, we compare the next significant pair of bits (A1 & B1).
- If A1 > B1, then A > B (GT=1)
  - Irrespective of the relative values of the other bits of A & B.
  - Example, A = 0010 and B = 0001 where A > B
- This can be stated as GT=1 if X3 X2 A1 B1'

- **Case 4: If A3 = B3 (X3 = 1) and A2 = B2 (X2 = 1) and A1 = B1 (X1 = 1)**, we compare the next pair of bits (A0 & B0).
- If A0 > B0 then A > B (GT=1).
- This can be stated as GT=1 if X3 X2 X1 A0 B0' = 1

4-bit Magnitude Comparator: Design of the GT Output – cont'd

- To summarize, GT =1 (A > B) IFF:
  - A3 B3' = 1, OR
  - X3 A2 B2' = 1, OR
  - X3 X2 A1 B1' = 1, OR
  - X3 X2 X1 A0 B0' = 1, OR

- In other words,

GT = A3 B3' + X3 A2 B2' + X3 X2 A1 B1'
+ X3 X2X1 A0 B0'

4-bit Magnitude Comparator: Design of the LE Output – cont'd

- In the same manner as for the GT output, we can derive the expression of the LT (A < B) output

LT = B3 A3′ + X3 B2 A2′ + X3 X2 B1 A1′
+ X3 X2 X1 B0 A0′

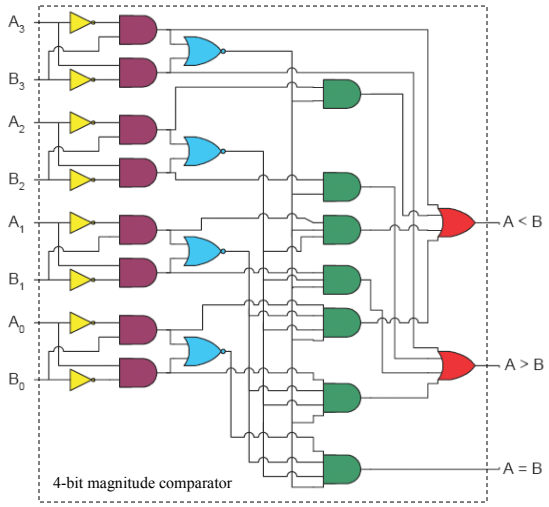
4-bit Magnitude Comparator: Circuit

- Three functions:

EQ = X3 X2 X1 X0

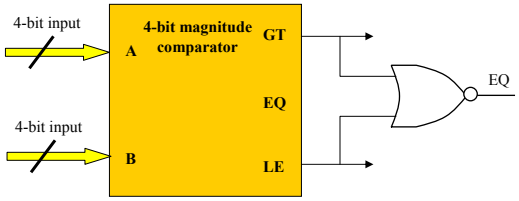
GT = A3 B3′ + X3 A2 B2′ + X3 X2
A1 B1′ + X3 X2X1 A0 B0′

LT = B3 A3′ + X3 B2 A2′ + X3 X2
B1 A1′ + X3 X2 X1 B0 A0′



4-bit Magnitude Comparator: Modification

- 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:



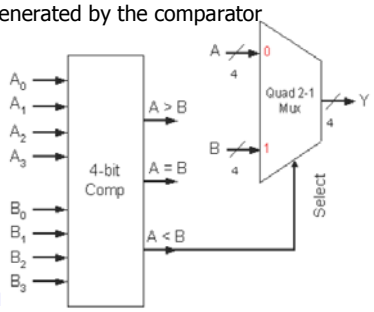
Design Example 2:

- **Problem**: Given two 4-bit unsigned numbers A and B, design a circuit which outputs the larger of the 2 numbers.

  Hint: use a quad 2-1 Mux.

- **Solution**:
- We need
  - 4-bit magnitude comparator to determine which number is greater
  - Quad 2-to-1 mux to select between A and B
  - The selection signal for the mux must be generated by the comparator
  - See circuit

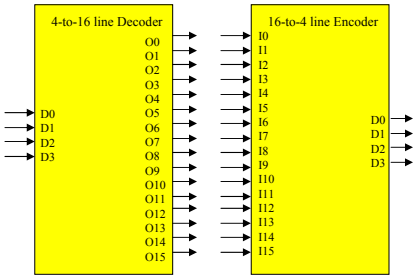
Note the relation between the selection signal and order of inputs 0 and 1 for the mux



Design Example 3: Excess-3 Code Converter

- **Problem**: Designing an Excess-3 code converter using a Decoder and an Encoder

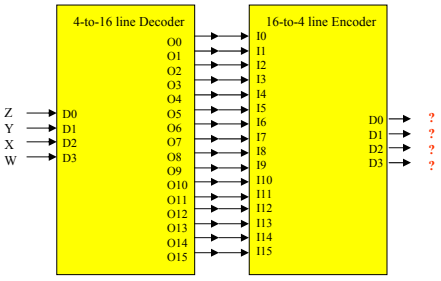
- **Solution**:
- Truth Table is as shown
- The outputs 0000, 0001, 0010, 1101, 1110, and 1111 are never generated (Why?)



	BCD input				Excess 3 code			
BCD digit	W	X	Y	Z	A	B	C	D
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Example 3: cont'd

- **Solution**:
- What do you think the output of the encoder will be?

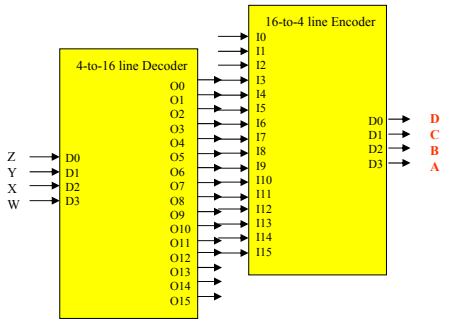


Output: D0 = Z, D1 = Y, D2 = X, D3 = W
So we get exact same code out!!

Example 3: cont'd

- **Solution**:
- We need to shift

- Example: suppose 0011 is sent as input. This will activate minterm 3 of the decoder. This output is connected to input 6 of encoder. Thus the encoder will generate the corresponding bit combination, which is 0110.



Design Example 4:

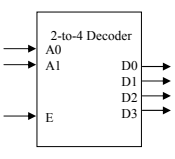
- **Problem**: Design a circuit that adds three 4-bit numbers – utilize 4-bit adder blocks and half/full adder blocks
- **Solution**: refer to the last example in the adder slides

Design Example 5:

- **Problem**: Design a 4-to-16 Decoder using **five** 2-to-4 Decoders with enable inputs. No extra logic is to be used.
- **Solution**:

Design Example 5: cont'd

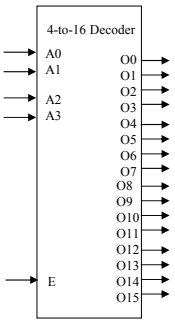
- **Solution**:
- How does a 2-to-4 decoder with Enable looklike?
- How does a 4-to-16 decoder looklike?



E	A1	A0	D0	D1	D2	D3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1
0	X	X	0	0	0	0

Truth Table

It is required to connect 4 boxes of 2-to-4 decoders to obtain the functionality of 4-to-16 decoder



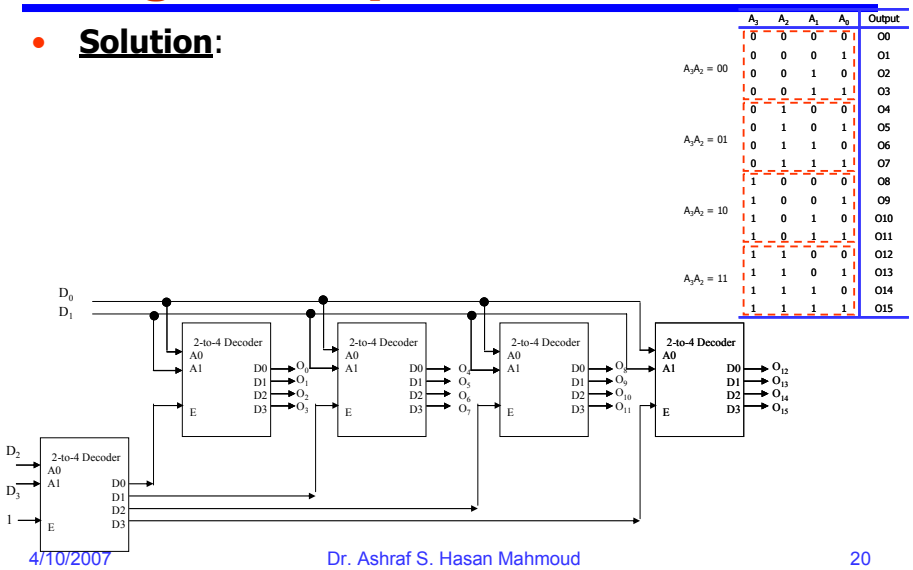
Design Example 5: cont'd

- **Solution**:
  - A3A2, when decoded, can enable the require ONE of FOUR 2-to-4 decoders
  - A1A0 always decoded into 4 outputs

	A_3	A_2	A_1	A_0	Output
$A_3A_2 = 00$	0	0	0	0	O0
	0	0	0	1	O1
	0	0	1	0	O2
	0	0	1	1	O3
$A_3A_2 = 01$	0	1	0	0	O4
	0	1	0	1	O5
	0	1	1	0	O6
	0	1	1	1	O7
$A_3A_2 = 10$	1	0	0	0	O8
	1	0	0	1	O9
	1	0	1	0	O10
	1	0	1	1	O11
$A_3A_2 = 11$	1	1	0	0	O12
	1	1	0	1	O13
	1	1	1	0	O14
	1	1	1	1	O15

Design Example 5: cont'd

- **Solution**:



Design Example 6:

- **Problem**: Designing a 16-bit adder using four 4-bit adders
- **Solution**:

Textbook Problems

4-4, 4-5, 4-6, …,4-25, 4-26
All are important problems!

Design Example 7: (similar to problem 4-21)

- **Problem**: A combinational circuit is defined by the following three Boolean functions:
  - F1 = (X+Y)' + XYZ'
  - F2 = (X+Y)' + X'YZ
  - F3 = XYZ + (X+Y')

  Design the circuit with a decoder and external OR gates

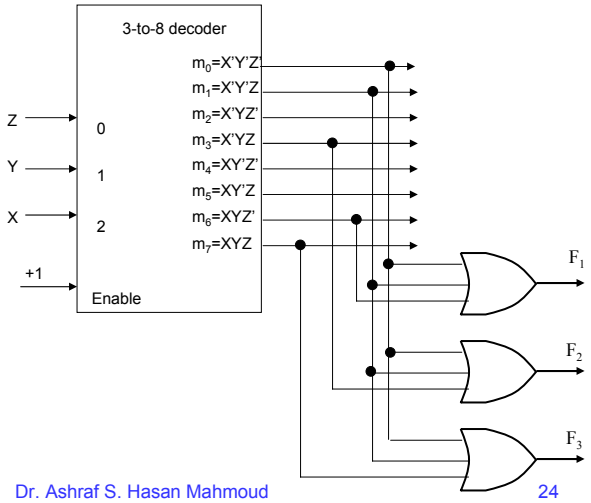
- **Solution**:

Design Example 7: cont'd

- **Solution**:

- You should write each function in terms of its minterms:
  - F1 = Σm(0,1,6)
  - F2 = Σm(0,1,3)
  - F3 = Σm(0,1,7)

- Use one OR gate for each function to sum the required minterms



Design Example 8: Problem 4-19 in textbook

- **Problem**: Construct a 15-to-1 line multiplexer with two 8-to-1 line multiplexers. Interconnect the two multiplexers and label the inputs such that any added logic required to have selection codes 0000 through 1110 is minimized.
- **Solution**:

Design Example 8: cont'd

- **Solution**:

A_3	A_2	A_1	A_0	Output
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7
1	0	0	0	D8
1	0	0	1	D9
1	0	1	0	D10
1	0	1	1	D11
1	1	0	0	D12
1	1	0	1	D13
1	1	1	0	D14
1	1	1	1