DIGITAL DESIGN AND COMPUTER ORGANIZATION

Binary Multipliers and Magnitude Comparator

Team DDCO
Department of Computer Science and Engineering

DIGITAL DESIGN AND COMPUTER ORGANIZATION

Binary Multipliers and Magnitude Comparator

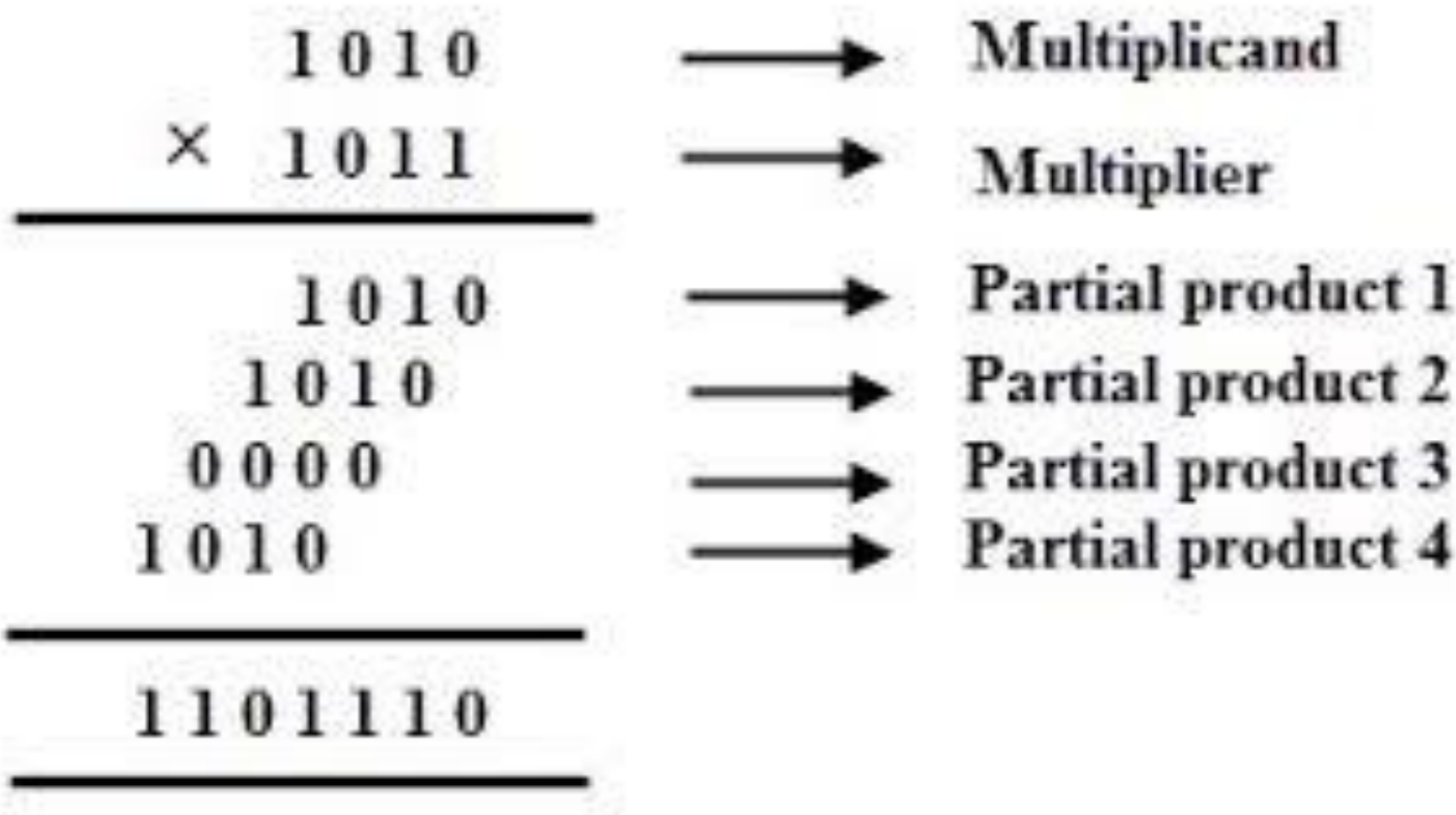
Department of Computer Science and Engineering

Combinational logic
Binary Multipliers

- Multiplication of binary numbers is performed in the same way as multiplication of decimal numbers.
- The **multiplicand is multiplied by each bit of the multiplier, starting** from the least significant bit. Each such multiplication forms a **partial product**. Successive partial products are shifted one position to the left.
- The final product is obtained from the sum of the partial products.

M. Morris Mano, Michael D. Ciletti, *Digital Design: Wi*
an Introduction to the Verilog HDL, 5th ed., Prentice Hall, 2012, Section 4.7

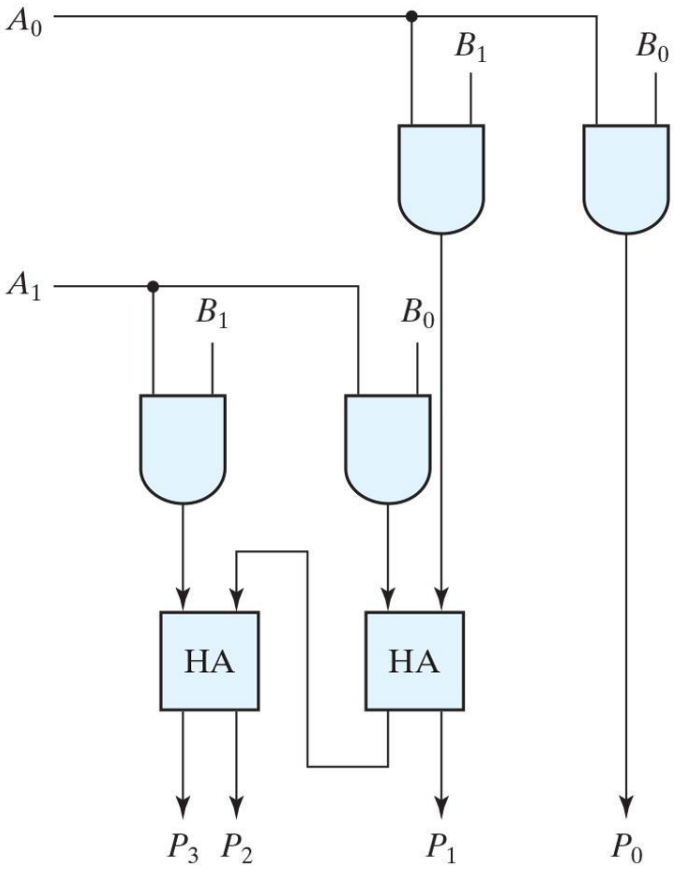
Combinational logic
Binary Multipliers



Combinational logic
Binary Multipliers

Two-bit by two-bit binary multiplier.

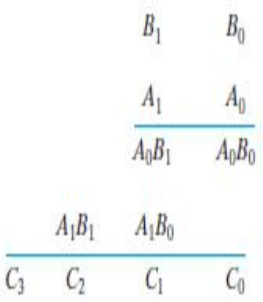
$$
\begin{array}{cccc}
 & & B_1 & B_0 \\
 & & A_1 & A_0 \\
\hline
 & & A_0B_1 & A_0B_0 \\
 & A_1B_1 & A_1B_0 & \\
\hline
P_3 & P_2 & P_1 & P_0
\end{array}
$$

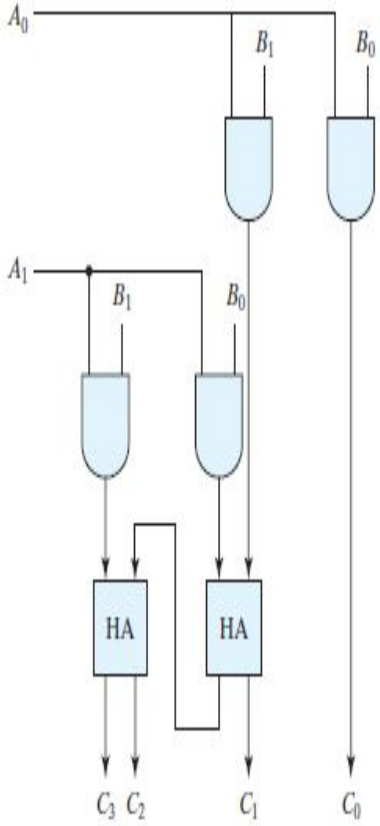


Combinational logic
Binary Multipliers

Consider the multiplication of two 2-bit numbers

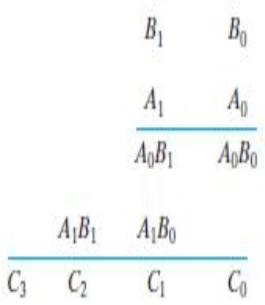
- The multiplicand bits are $B1$ and $B0$, the multiplier bits are $A1$ and $A0$,and the product is $C3C2C1C0$.
- **The first partial product is formed by multiplying $B1B0$ by $A0$.**
- The multiplication of two bits such as $A0$ and $B0$ produces a 1 if both bits are 1; otherwise, it produces a 0. This is identical to an **AND** operation. Therefore, the partial product can be implemented with AND gates as shown in the diagram.

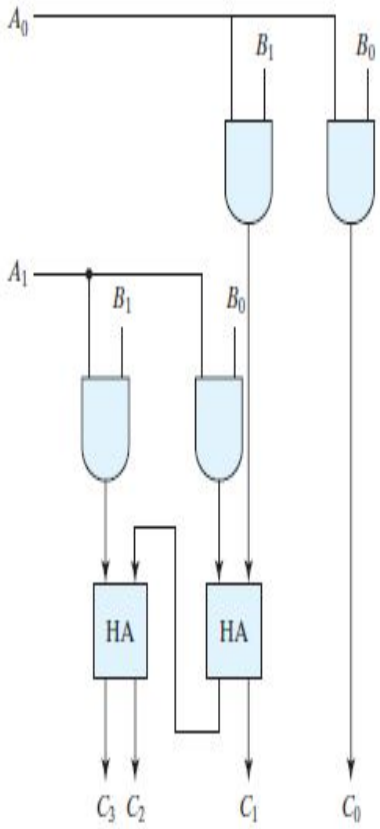




Combinational logic
Binary Multipliers

- The second partial product is formed by **multiplying** $B1B0$ by $A1$ and shifting one position to the left.
- The **two partial products** are added with two **half-adder (HA) circuits**. Usually, there are more bits in the partial products and it is necessary to use full adders to produce the sum of the partial products.
- Note that the least significant bit of the product does not have to go through an adder, since it is formed by the output of the first AND gate.





Combinational logic

Binary Multipliers

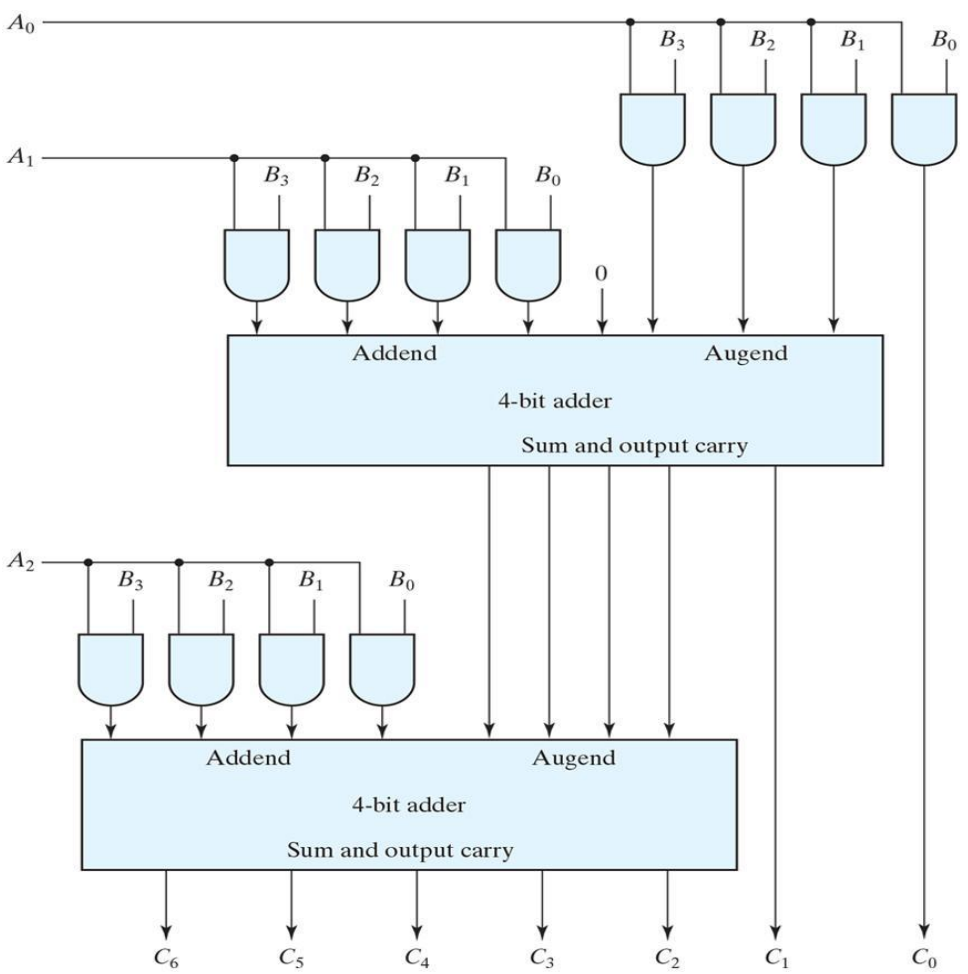
- A combinational circuit binary multiplier with more bits can be constructed in a similar fashion.
- A bit of the multiplier is ANDed with each bit of the multiplicand in as many levels as there are bits in the multiplier.
- The binary output in each level of AND gates is added with the partial product of the previous level to form a new partial product. The last level produces the product.
- For *J* multiplier bits and *K* multiplicand bits, we **need *J* * *K* AND gates** and
- **(*J* – 1) *K* -bit adders to produce a product** of **(*J* + *K*) bits**.

Combinational logic

Binary Multipliers- 1011 X 001

Four-bit by three-bit binary multiplier.

- Consider a multiplier circuit that multiplies a binary number represented by four bits by a number represented by three bits.
- Let the multiplicand be represented by $B3B2B1B0$ and the multiplier by $A2A1A0$.
- Since $K = 4$ and $J = 3$, we need 12 AND gates and two 4-bit adders to produce a product of seven bits



Combinational logic
Magnitude Comparators

- **A *magnitude comparator* is a combinational circuit that compares two numbers *A* and *B* and determines their relative magnitudes. The outcome of the comparison is specified by three binary variables that indicate whether *A* > *B*, *A* = *B*, or *A* < *B*.**
- On the one hand, the circuit for comparing two n -bit numbers has $\mathbf{2^{2n}}$ **entries** in the truth table and becomes too cumbersome, even with $n = 3$
- Consider two numbers, A and B , with four digits each. Write the coefficients of the numbers in descending order of significance:

  $A = A3\ A2\ A1\ A0$

  $B = B3\ B2\ B1\ B0$

- **The two numbers are equal if all pairs of significant digits are equal:**
- ***A*3 = *B*3, *A*2 = *B*2, *A*1 = *B*1, and *A*0 = *B*0.**

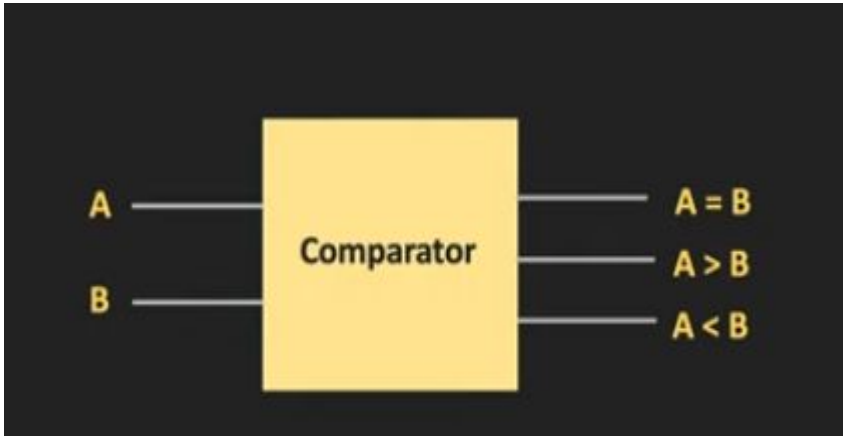
M. Morris Mano, Michael D. Ciletti, *Digital Design: With an Introduction to the Verilog HDL*, 5th ed., Prentice Hall, 2012, Section 4.8

Combinational logic
Magnitude Comparators

- When the numbers are binary, the digits are either 1 or 0, and the equality of each pair of bits can be expressed logically with an exclusive-NOR function as

$$x_i = A_i B_i + A_i` B_i` \text{ for } i = 0, 1, 2, 3$$

where $\boldsymbol{x_i}$ **= 1 only if the pair of bits in position** ***i*** **are equal** (i.e., if both are 1 or both are 0).



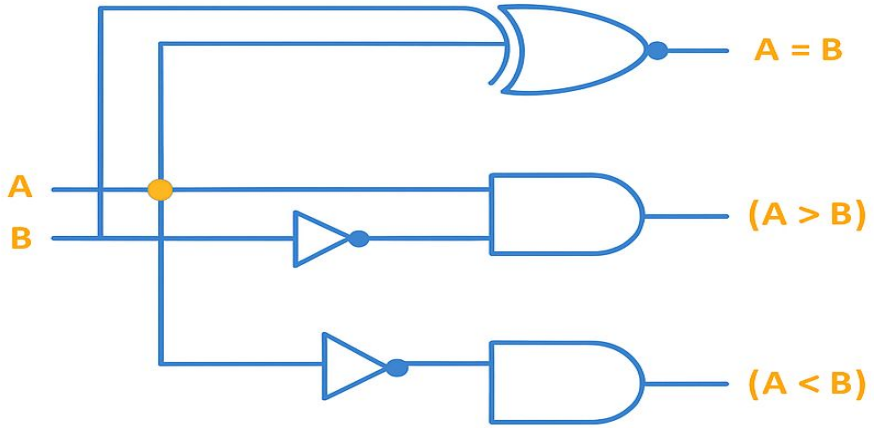
Combinational logic
Magnitude Comparators

1-bit Comparator

Truth Table

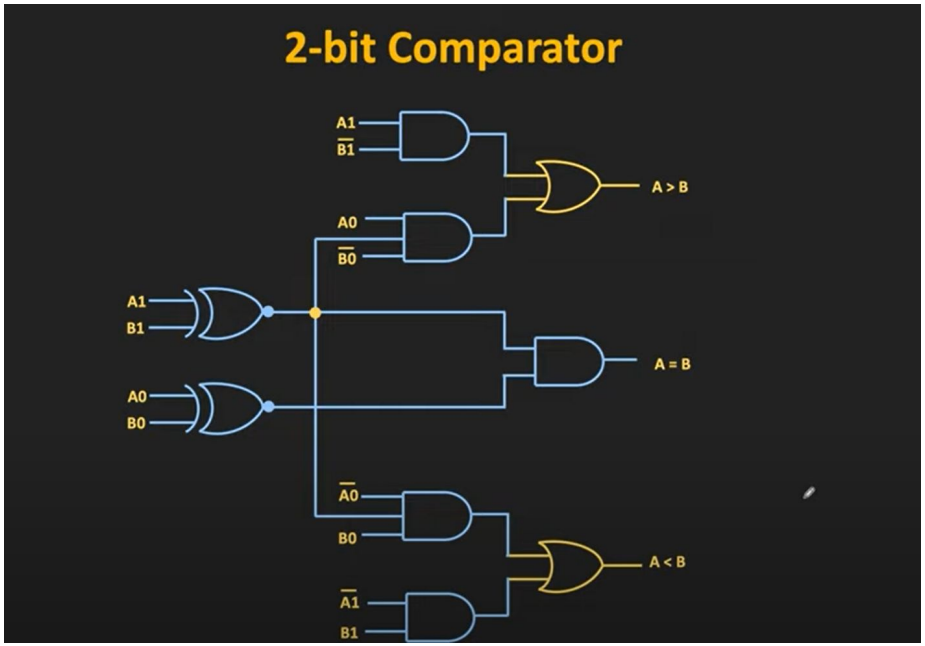
A	B	A = B	A > B	A < B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Output = $\bar{A}\,\bar{B} + AB$



2 bit comparator

1- bit Comparator	2 variables	4 rows
2- bit Comparator	4 variables	16 rows
n- bit Comparator	2n variables	2^{2n} rows



A_1	A_0	B_1	B_0	A > B	A < B	A = B
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

Combinational logic
Magnitude Comparators

4 Bit Comparator

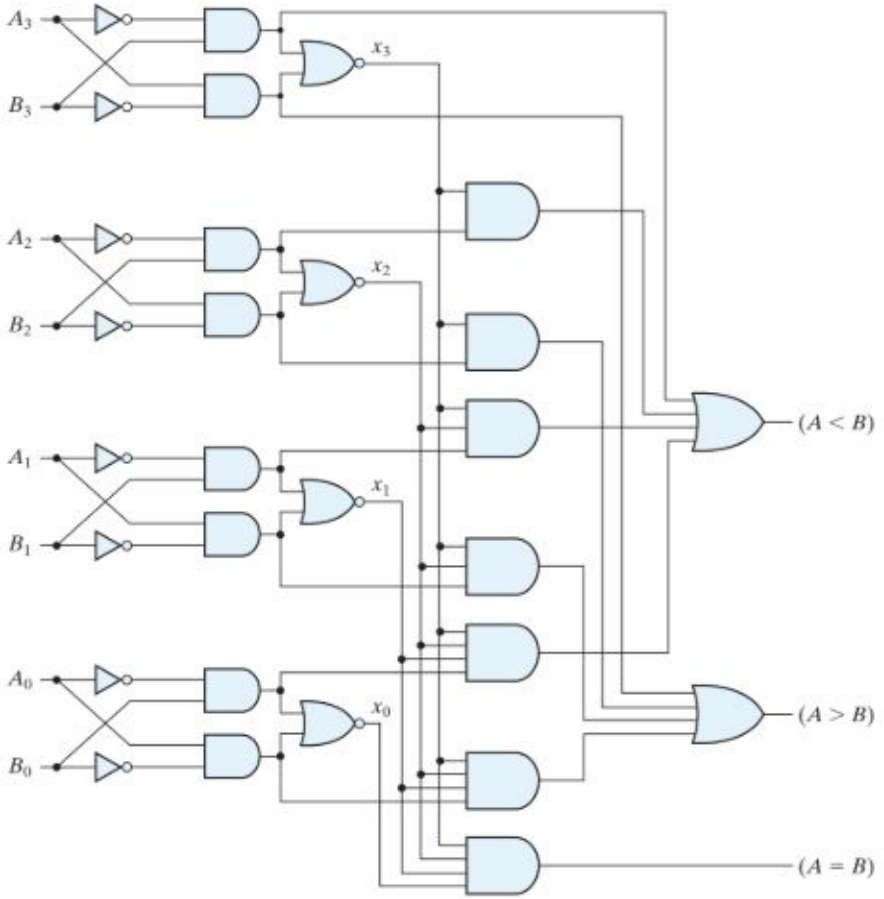


FIGURE 4.17
Four-bit magnitude comparator

$A = A_3\ A_2\ A_1\ A_0$ $\qquad$ $B = B_3\ B_2\ B_1\ B_0$

$A > B$ ⟹ $A_3 > B_3$

OR

If $A_3 = B_3$ and $A_2 > B_2$

OR

If $A_3 = B_3$ and $A_2 = B_2$ and $A_1 > B_1$

OR

If $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 > B_0$

$$(A > B) = A_3 \overline{B_3} + (A_3 \odot B_3) A_2 \overline{B_2} + (A_3 \odot B_3)(A_2 \odot B_2) A_1 \overline{B_1} +$$

$$(A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1) A_0 \overline{B_0}$$

$$(A < B) = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1' + x_3x_2x_1A'n_0B_0'$$

$A = A_3\ A_2\ A_1\ A_0$ $\qquad$ $B = B_3\ B_2\ B_1\ B_0$

$A < B$ ⇒ $A_3 < B_3$

OR

If $A_3 = B_3$ and $A_2 < B_2$

OR

If $A_3 = B_3$ and $A_2 = B_2$ and $A_1 < B_1$

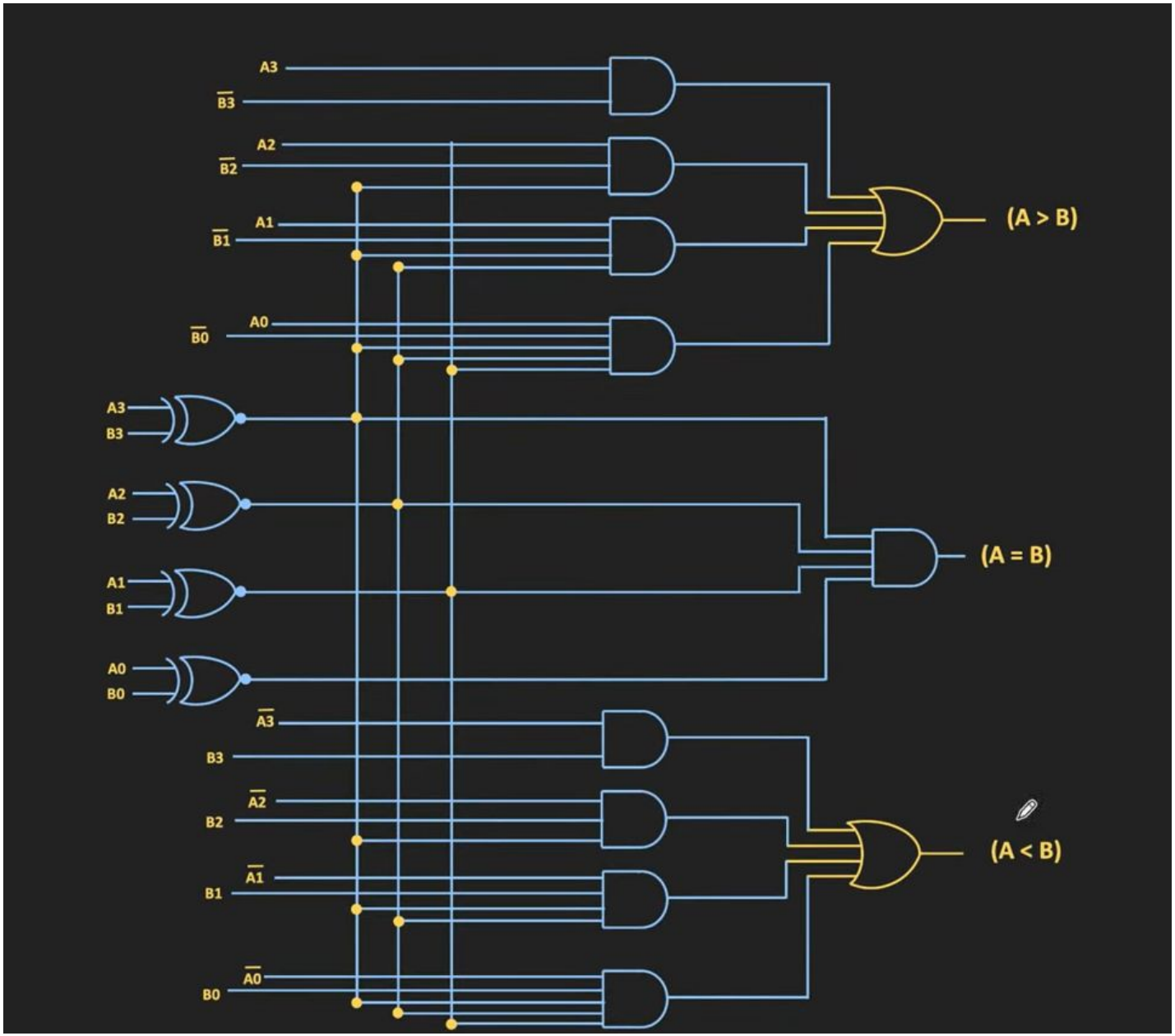
OR

If $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = B_1$ and $A_0 < B_0$

$$(A < B) = \overline{A_3}\,B_3 + (A_3 \odot B_3)\,\overline{A_2}\,B_2 + (A_3 \odot B_3)(A_2 \odot B_2)\,\overline{A_1}\,B_1 +$$

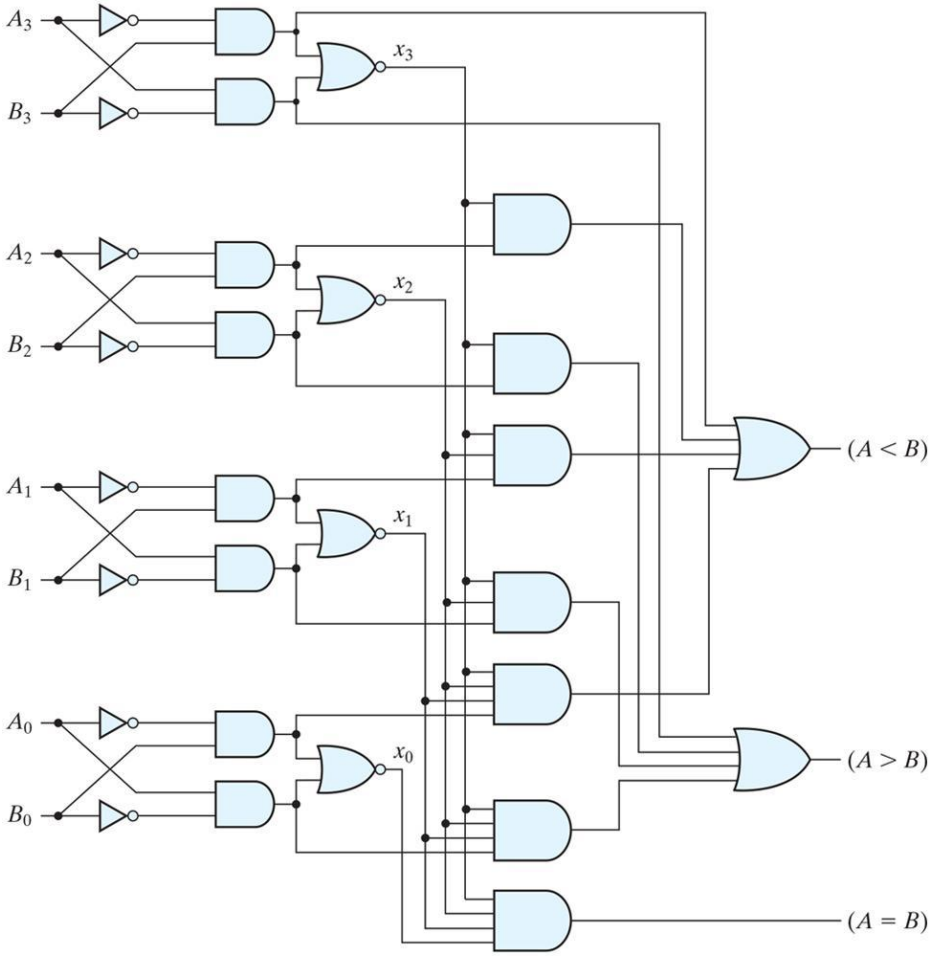
$$(A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)\,\overline{A_0}\,B_0$$

Magnitude Comparator



Combinational logic
Magnitude Comparators

Four-bit Magnitude Comparator



Combinational logic
Magnitude Comparators

- For equality to exist, all *xi* variables must be equal to 1, a condition that dictates an AND operation of all variables:

  $(A = B) = x3x2x1x0$

- The *binary* variable $(A = B)$ is equal to 1 only if all pairs of digits of the two numbers are equal.
- **To determine whether *A* is greater or less than *B* , we inspect the relative magnitudes of pairs of significant digits, starting from the most significant position.**
- If the two digits of a pair are equal, we compare the next lower significant pair of digits.
- **The comparison continues until a pair of unequal digits is reached. If the corresponding digit of *A* is 1 and that of *B* is 0, we conclude that $A > B$.**
- **If the corresponding digit of *A* is 0 and that of *B* is 1, we have $A < B$.**

Combinational logic
Magnitude Comparators

- The sequential comparison can be expressed logically by the two Boolean functions

$$(A > B) = A_3B'_3 + x_3A_2B'_2 + x_3x_2A_1B'_1 + x_3x_2x_1A_0B'_0$$
$$(A < B) = A'_3B_3 + x_3A'_2B_2 + x_3x_2A'_1B'_1 + x_3x_2x_1A'n_0B'_0$$

- The symbols (A > B) and (A < B) are binary output variables that are equal to 1 when A > B and A < B, respectively.
- The unequal outputs can use the same gates that are needed to generate the equal output
- The four x outputs are generated with exclusive-NOR circuits and are applied to an AND gate to give the output binary variable (A = B) .
- The other two outputs use the x variables to generate the Boolean functions listed previously. This is a multilevel implementation and has a regular pattern.

Combinational logic

Think about it

A 4-bit by 4-bit binary multiplier is implemented using AND gates and adders. How many AND gates are required?

A. 16
B. 8
C. 32
D. 64

For comparing two 4-bit numbers A and B, a combinational circuit gives three outputs A > B, A = B, and A < B. If each bit comparison uses XNOR and AND/OR logic, how many XNOR gates are needed?

A. 4
B. 3
C. 2
D. 1

Combinational logic

Think about it

A 4-bit by 4-bit binary multiplier is implemented using AND gates and adders. How many AND gates are required?

A. 16
B. 8
C. 32
D. 64

Answer: A. 16
Explanation: 4 × 4 = 16 partial products via AND gates.

For comparing two 4-bit numbers A and B, a combinational circuit gives three outputs A > B, A = B, and A < B. If each bit comparison uses XNOR and AND/OR logic, how many XNOR gates are needed?
A. 4
B. 3
C. 2
D. 1
Answer: A. 4
Explanation: Each bit position requires 1 XNOR gate to check equality.

Combinational logic

Think about it

Describe a real-world application where a magnitude comparator is essential. Explain how the comparator's outputs are used in decision-making.

Combinational logic

Think about it

Describe a real-world application where a magnitude comparator is essential. Explain how the comparator's outputs are used in decision-making.

Ans: **Digital Thermostat in an Air Conditioning (AC) System**

A **digital thermostat** constantly monitors the **room temperature** and compares it with the **desired set temperature**. A **magnitude comparator** is used to perform this comparison.

A: Current room temperature (in binary)

B: Desired set temperature (in binary)

Combinational logic

Think about it

Comparator Outputs:

A > B → Room is **warmer** than desired

A = B → Room is at **desired temperature**

A < B → Room is **colder** than desired

Comparator Output	System Decision
A > B	Turn ON cooling (AC compressor starts)
A = B	Maintain current state (system remains idle)
A < B	Turn OFF cooling or turn ON heating (if available)

Combinational logic

Think about it

Other Examples Where Comparators Are Used:

Speed regulators in vehicles (comparing set vs. actual speed)

Digital counters (checking if max value is reached)

Elevator control systems (comparing current floor with target floor)

Battery chargers (comparing voltage levels)

THANK YOU

Team DDCO

Department of Computer Science