Digital Logic Design (CSNB163)

Module 10

Recapss..
Combinational Logic Circuit

- In the previous module, we have been introduced to the concept of combinational logic circuits through the examples of binary adders:
  - Half adder: HA adds two binary inputs (i.e. A0 and B0) and produces two binary outputs (i.e. Sum and C_{out})
  - Full adder: FA adds three binary inputs (i.e. A0, B0 and C_{in}) and produces two binary outputs (i.e. Sum and C_{out})
- In this module, we shall learn about two more examples of combinational logic circuits:
  - **Binary multiplier**
  - **Magnitude comparator**

Binary Multiplier

- A binary multiplier performs the **multiplication of two binary numbers** (multiplicand & multiplier) whereby:
  - The **multiplicand** is multiplied by **each bit** of the **multiplier** starting from the least significant bit.
  - Each multiplication **forms a partial product** and **successive** partial products are **shifted one position to the left**.
  - The final product is obtained from the **sum** of the partial products.

2-bit by 2 bit Binary Multiplier

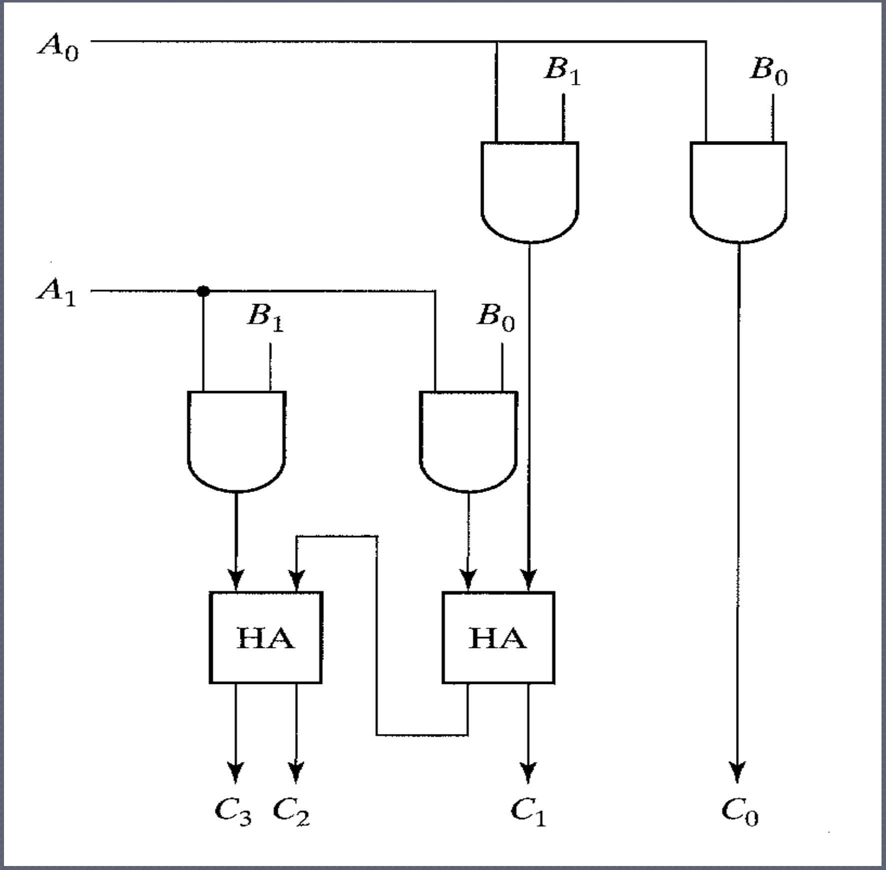
- E.g.

		B1	B0	Multiplicand
	×	A1	A0	Multiplier
		B1A0	B0A0	Partial products
+	B1A1	B0A1		
C3	C2	C1	C0	Final product

  - The multiplication of **all partial products**, say B0A0 will only be 1 if both A0 and B0 are both 1, otherwise partial product B0A0 will be 0 (i.e. similar to **AND** operation).
  - **C1** can be obtained by using **half adder** (HA1) that adds **B1A0** with **B0A0**.
  - **C2** can be obtained by using **half adder** (HA2) that adds **the output carry of HA1** with **B1A1** – whereas **C3** is the **output carry of HA2**.

2-bit by 2 bit Binary Multiplier (cont.)

- Circuit diagram for a 2-bit by 2-bit binary multiplier:

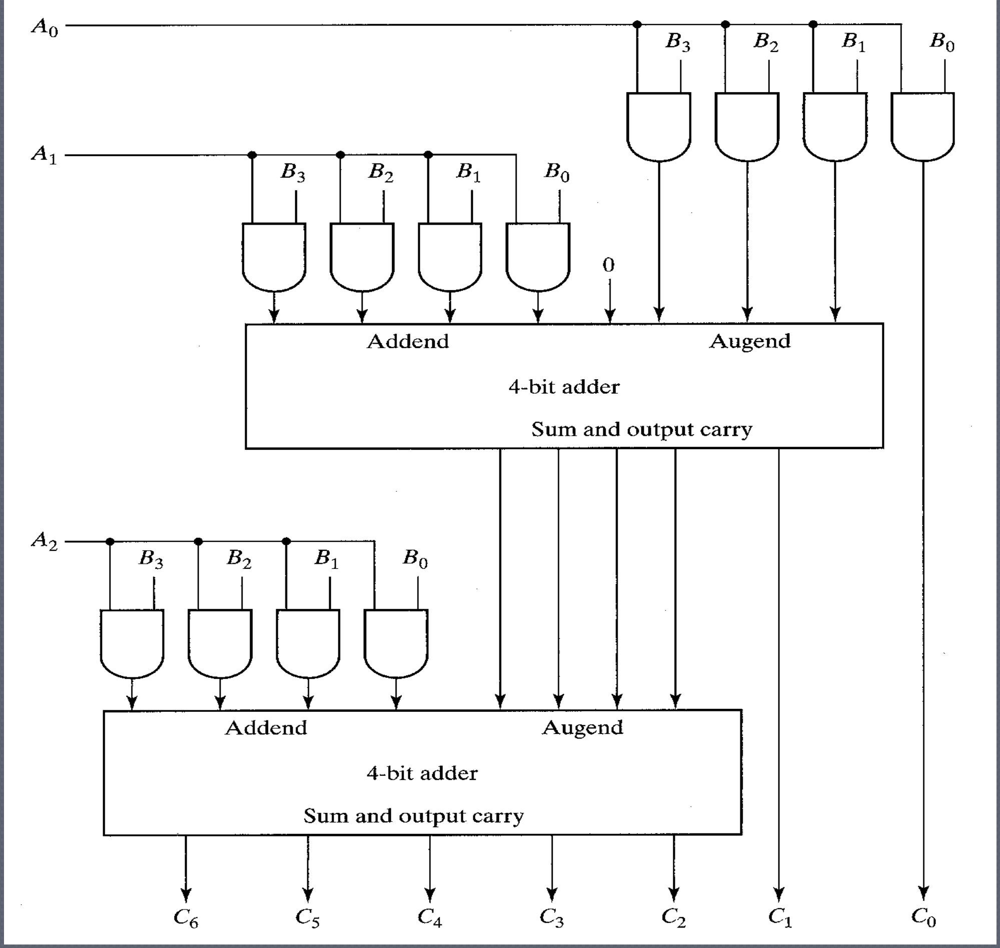


J-bit by *K*-Bit Binary Multiplier

- Usually, there are more bits in the partial products and it is necessary to use full adders to produce the sum of the partial product.
- A combinational circuit binary multiplier with more bits can be constructed in a similar fashion.
- For J multiplier and K multiplicand bits, (J x K) AND gates and (J - 1) K-bit adders are needed to produce a product of J + K bits.

J-bit by *K*-Bit Binary Multiplier (cont.)

- E.g.
  Circuit diagram for a 4-bit by 3 bit multiplier:
  - 4 bits multiplicand ($B_3B_2B_1B_0$) , K = 4
  - 3 bits multiplier ($A_2A_1A_0$)., J= 3



Magnitude Comparator

- A magnitude comparator is a combinational circuit that compares 2 binary numbers (A, B) and determines their relative magnitudes.
- The outcome of the comparison is specified by three binary variables that indicate whether:
  - ***A > B*** ***(requires test for magnitude comparison)***
  - ***A = B*** ***(requires test for equality)***
  - ***A < B*** ***(requires test for magnitude comparison)***
- For the following illustrations, we use the example of comparing two binary numbers with 4 bits each
  - $A = A_3A_2A_1A_0$
  - $B = B_3B_2B_1B_0$

Magnitude Comparator – Test for Equality

- To determine if A=B:
  - All bits of the 2 numbers are equal, $A_3 = B_3, A_2 = B_2, A_1 = B_1, A_0 = B_0$.
  - The equality of each bit can be expressed logically with an **exclusive-NOR** function as;

$$x_i = A_iB_i + A'_iB'_i \qquad \text{for } i = 0,1,2,3$$

  - Recaps… $x_i = 1$ only if both A_i and B_i equal to 1 or both A_i and B_i equal to 0.
  - For equality to exist across all bits, all x_i variables must be equal to 1 such that:

$$(A = B) = x_3 x_2 x_1 x_0$$

  - \- hence, the use of **AND** function.

Magnitude Comparator – Test for Magnitude Comparison

- To determine whether $A > B$ or $A < B$:
  - The relative magnitude of pairs of digits starting from the **most** significant position is compared.
  - If the 2 digits of a pair are equal, proceed **next** to compare the next lower significant pair of digits until a pair of unequal digit is reached.

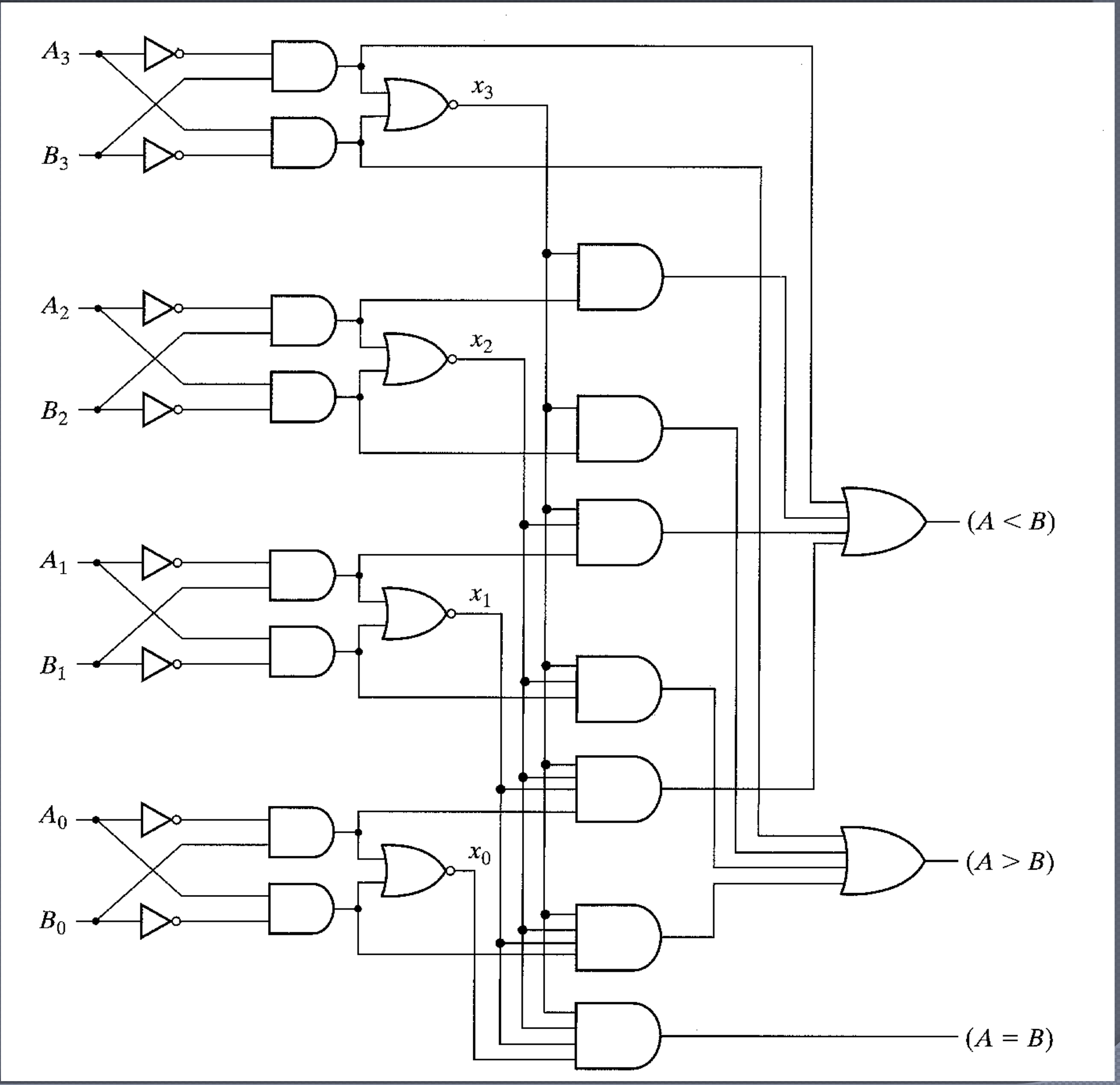
    If A = 1 and B = 0, then A > B.
    Else if A = 0 and B = 1, then A < B.

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

$$(A > B) = A_3B'_3 + x_3A_2B'_2 + x_3x_2A_1B'_1 + x_3x_2x_1A_0B'_0$$
$$(B > A) = A'_3B_3 + x_3A'_2B_2 + x_3x_2A'_1B_1 + x_3x_2x_1A'_0B_0$$

Magnitude Comparator (cont.)

- The circuit diagram for a 4 bit magnitude comparator:



Digital Logic Design (CSNB163)

End of Module 10