# Binary Multiplier and Magnitude Comparator

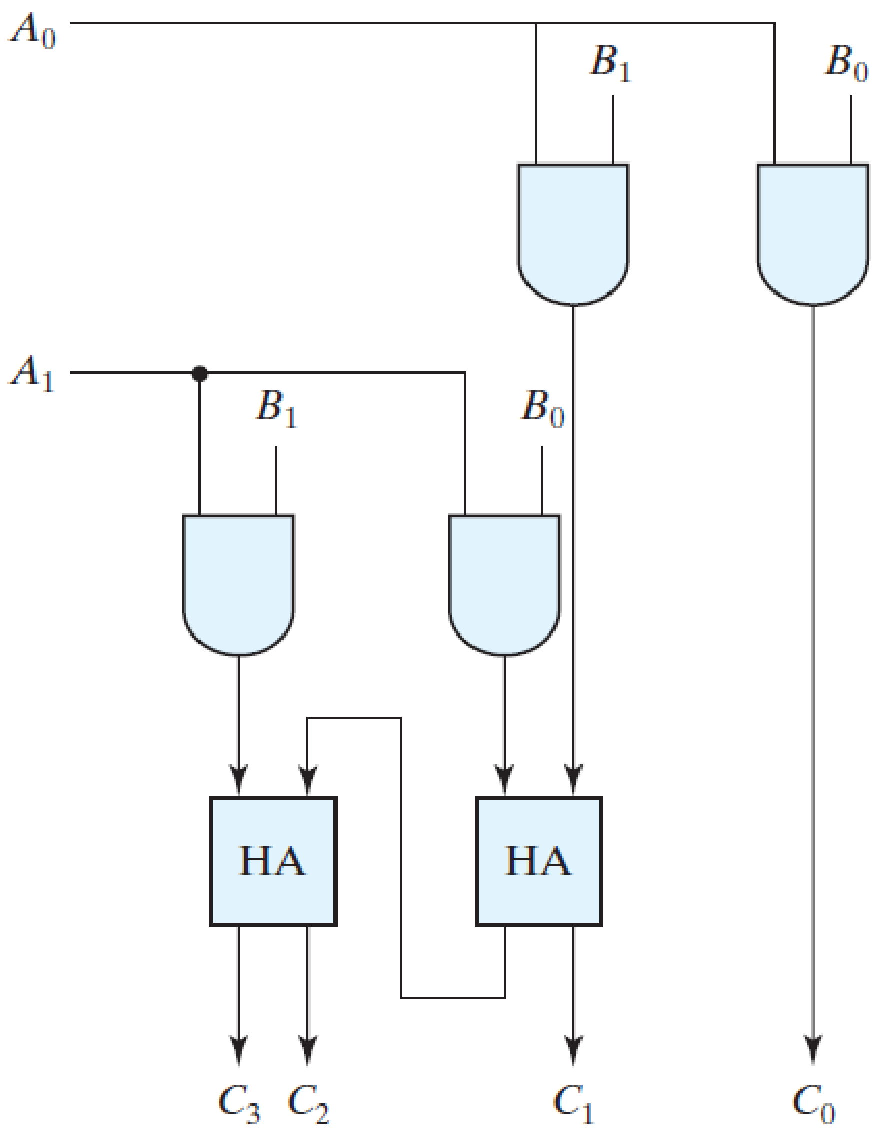
Sparsh Mittal

# 2-bit multiplier

- The first partial product is formed by multiplying $B1B0$ by $A0$.
- Multiplication of two bits such as $A0$ and $B0$ produces a 1 if both bits are 1; otherwise, it produces a 0. ➔ An AND operation.
- 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.

$$
\begin{array}{cccc}
 & & B_1 & B_0 \\
 & & A_1 & A_0 \\
\cline{3-4}
 & & A_0B_1 & A_0B_0 \\
 & A_1B_1 & A_1B_0 & \\
\hline
C_3 & C_2 & C_1 & C_0
\end{array}
$$

# Logic diagram

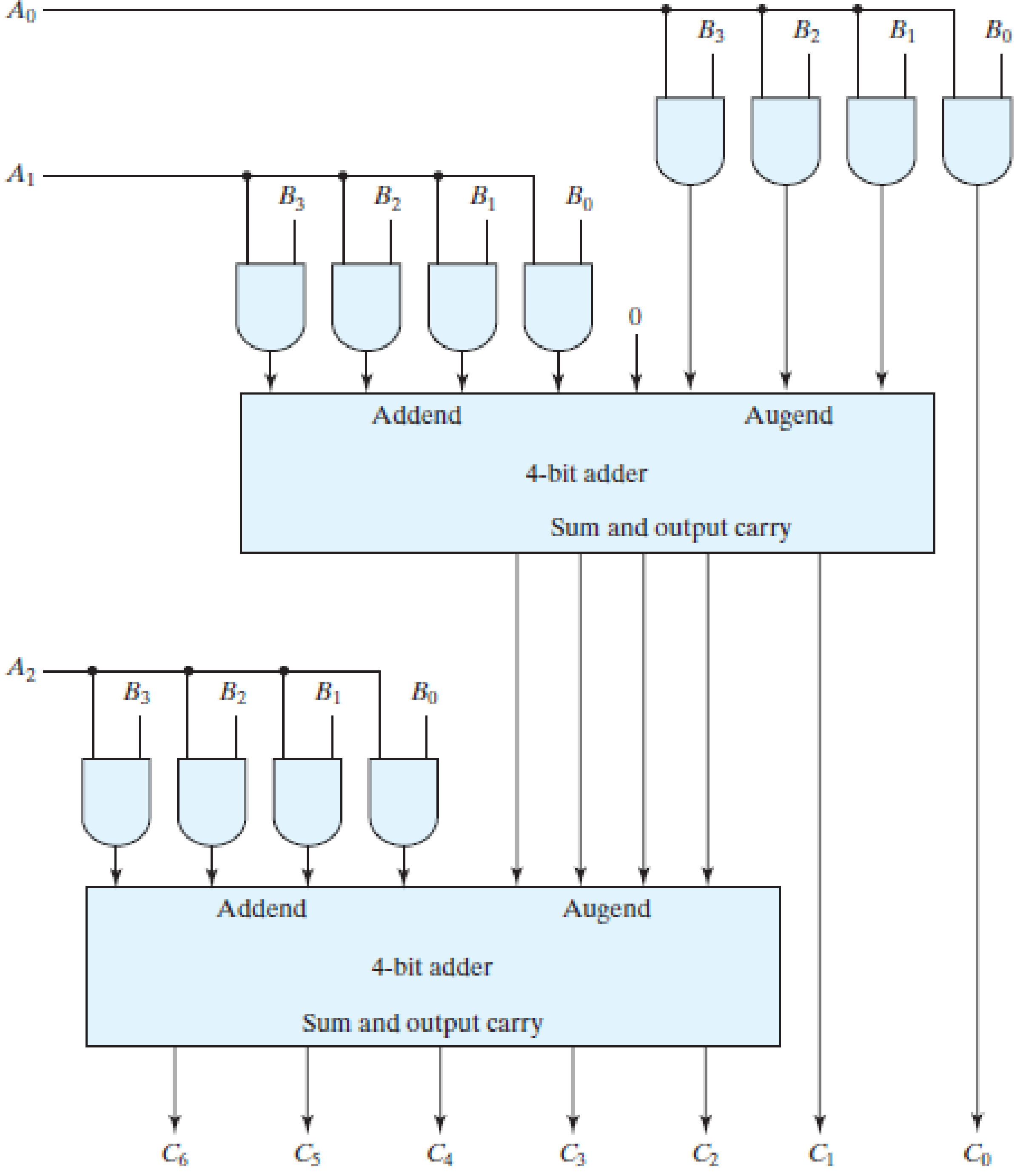


# Multiplication of >2 bits

- A bit of the multiplier is ANDed with each bit of multiplicand in as many levels as there are bits in the multiplier.
- The binary output in each level of AND gates is added with 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.

# Example: 4b*3b multiplication

- The multiplicand $B3B2B1B0$
- 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.



# Magnitude Comparator

# *Magnitude comparator*

- The outcome of the comparison is specified by three binary variables that indicate whether $A > B$, $A = B$, or $A < B$.
- Making a truth table is infeasible
- Given the regularity, we can design an algorithm

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

XNOR To check bitwise equality $\quad x_i = A_iB_i + A_i'B_i' \quad \text{for } i = 0, 1, 2, 3$

AND of bitwise equality signals to check number equality $\quad (A = B) = x_3x_2x_1x_0$

# Finding A>B or A<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 $A < B$.

# Expressions

$$(A>B) = A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0B_0'$$
$$(A<B) = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1 + x_3x_2x_1A_0'B_0$$

- The unequal outputs can use the same gates that are needed to generate the equal output.

# Circuit

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

$$(A = B) = x_3x_2x_1x_0$$

$(A>B) = A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0B_0'$

$(A<B) = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1 + x_3x_2x_1A_0'B_0$

- This is a multilevel implementation and has a regular pattern.