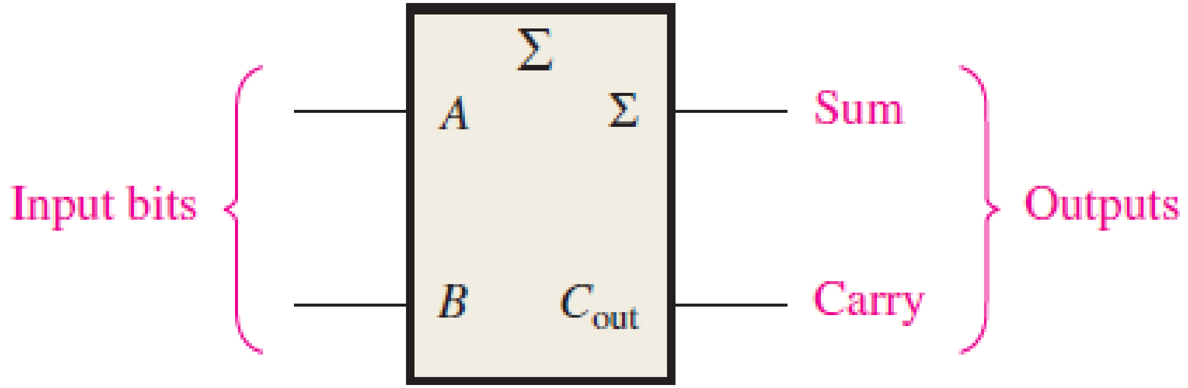
Lecture -5
Combination Circuits-1

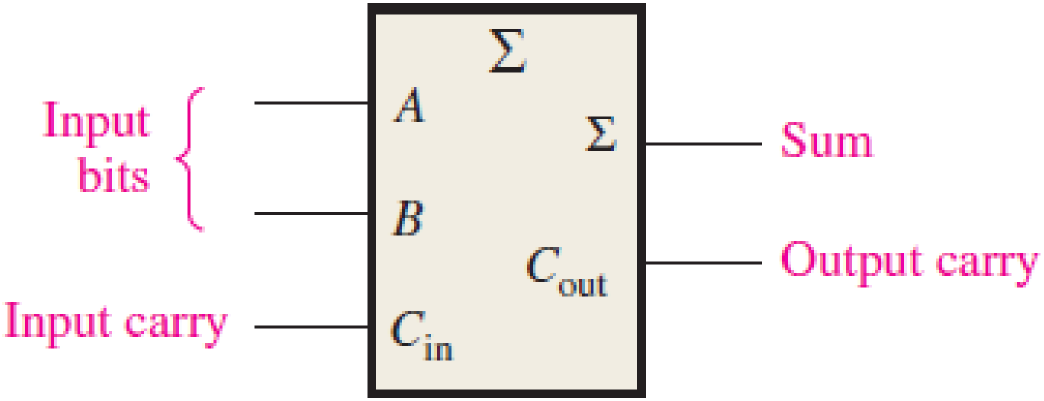
Prepared By: Asif Mahfuz

Adder

- Adders are important in computers and in other digital systems.
- To design an adder we first need to understand the basic adder operation.
- Thus we first need to know the rules of binary addition.
- The rules of binary addition are as follows:
  - $0 + 0 = \ \ 0$
  - $0 + 1 = \ \ 1$
  - $1 + 0 = \ \ 1$
  - $1 + 1 = 10$
- There are two types of adders namely, **Half-adder** and a **Full-adder**.



Block diagram of a Half-Adder



Block diagram of a Full-Adder

Half-Adder

- A half-adder accepts two binary digits on its inputs and produces two binary digits on its outputs- a sum bit and a carry bit.
- So a half-adder has two inputs and two outputs.
- Now to design a digital system we need to know its outputs for all the input combinations.
- The truth-table for the half adder can constructed following the binary addition rule.

A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

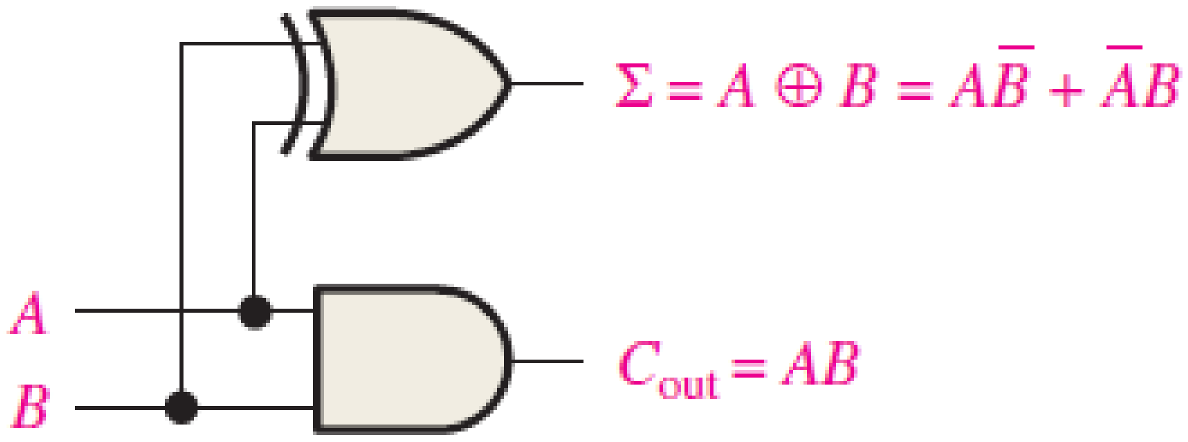
- The Boolean expression of the sum bit can be found as:

$$\mathbf{S = \bar{A}B + A\bar{B} = A \oplus B}$$

- The Boolean expression of carry bit can be found as:

$$\mathbf{C_{IN} = AB}$$

- So now a logical circuit can be drawn for a half-adder.



Full-Adder

- A full-adder accepts two input bits and an input carry and generates a sum output and a carry output.
- So a full adder has three inputs and two outputs.
- The truth table can be constructed the same way as we constructed for a half-adder.

A	B	C_{IN}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Full-Adder

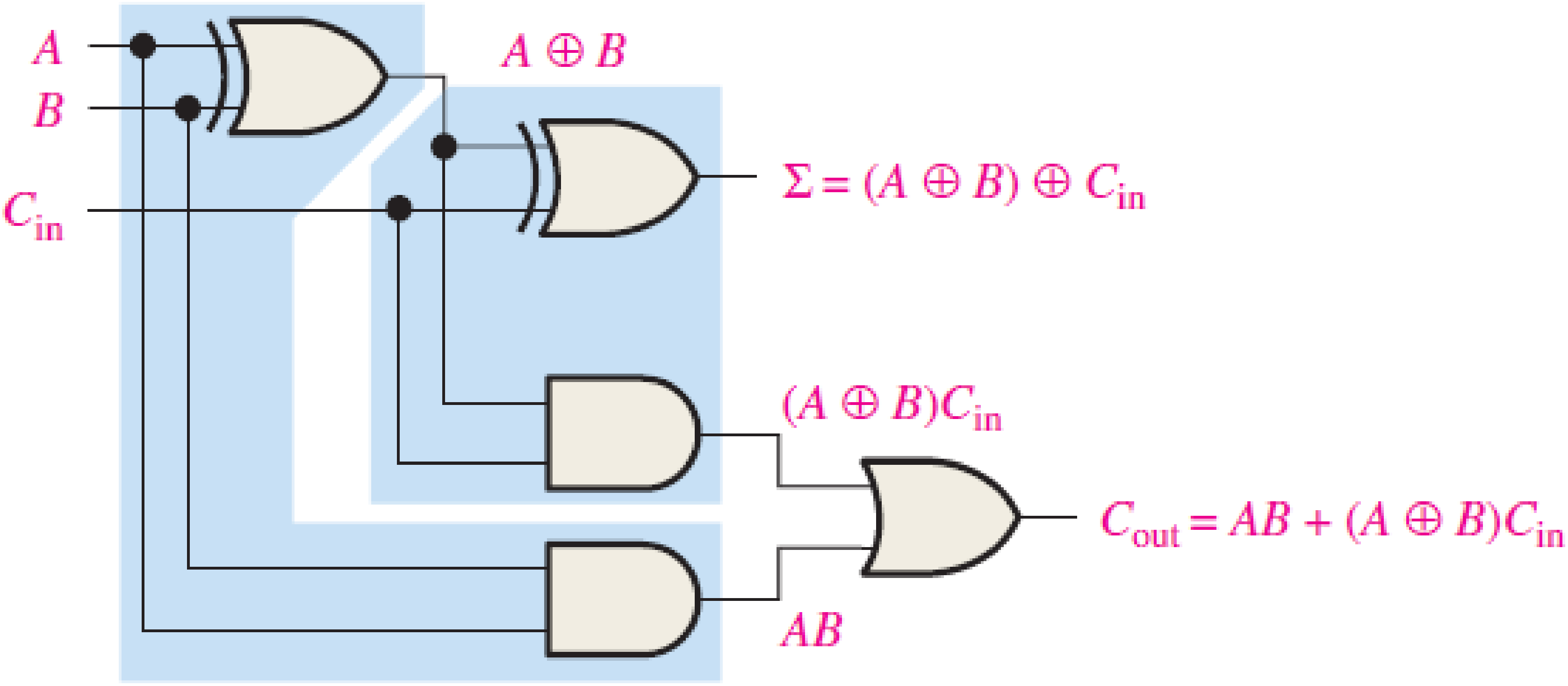
- After constructing the truth, now we can find the Boolean expressions

$$\mathbf{S = \bar{A}\bar{B}C_{IN} + \bar{A}B\overline{C_{IN}} + A\bar{B}\,\overline{C_{IN}} + ABC_{IN}}$$

$$\mathbf{S = A \oplus B \oplus C}$$

$$\mathbf{C_{OUT} = \bar{A}BC_{IN} + A\bar{B}C_{IN} + AB\overline{C_{IN}} + ABC_{IN}}$$

$$\mathbf{C_{OUT} = AB + (A \oplus B)C_{IN}}$$



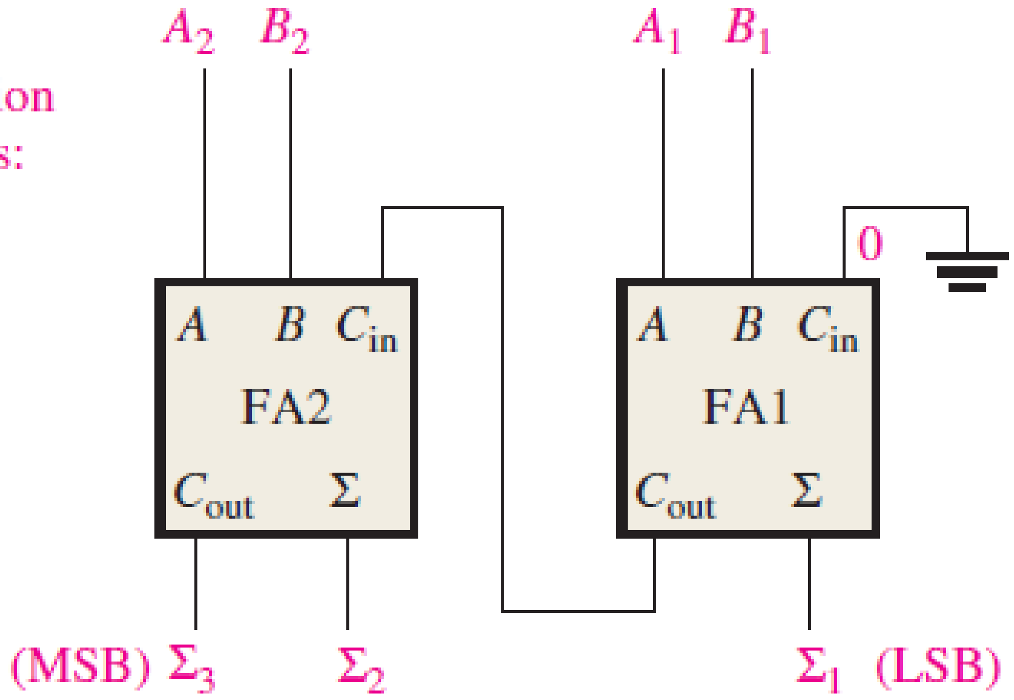
Parallel Adder

- Till now we have learned how add two numbers of 1 bit.
- So how do we add two number of two bits.
- The principle is same as we add two decimal numbers of two digits.
- We sum the first digits and below the first digits we write the sum and the generated carry is summed with the next digits.

Carry	1	
Number1	2	9
Number2	3	5
Sum	6	4

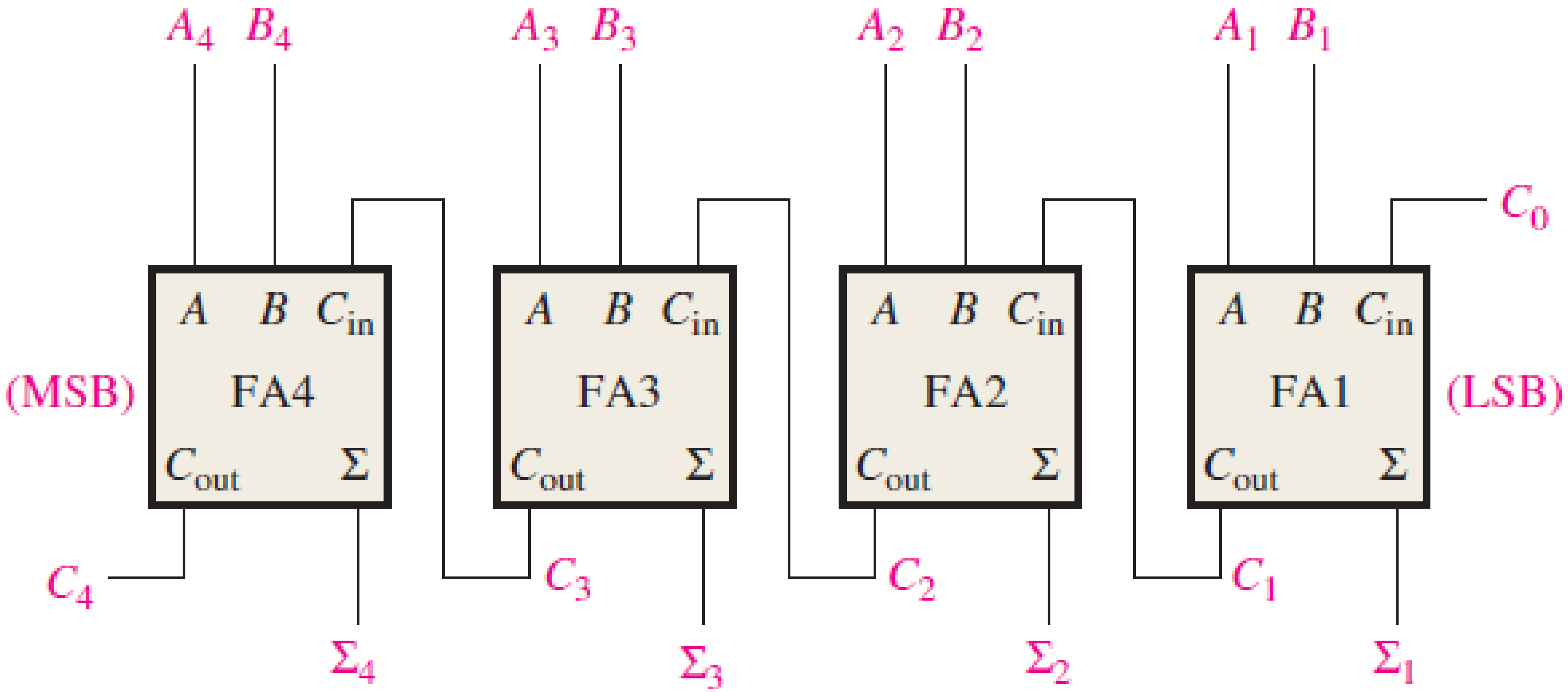
General format, addition of two 2-bit numbers:

$$\begin{array}{r} A_2A_1 \\ +\,B_2B_1 \\ \hline \Sigma_3\Sigma_2\Sigma_1 \end{array}$$



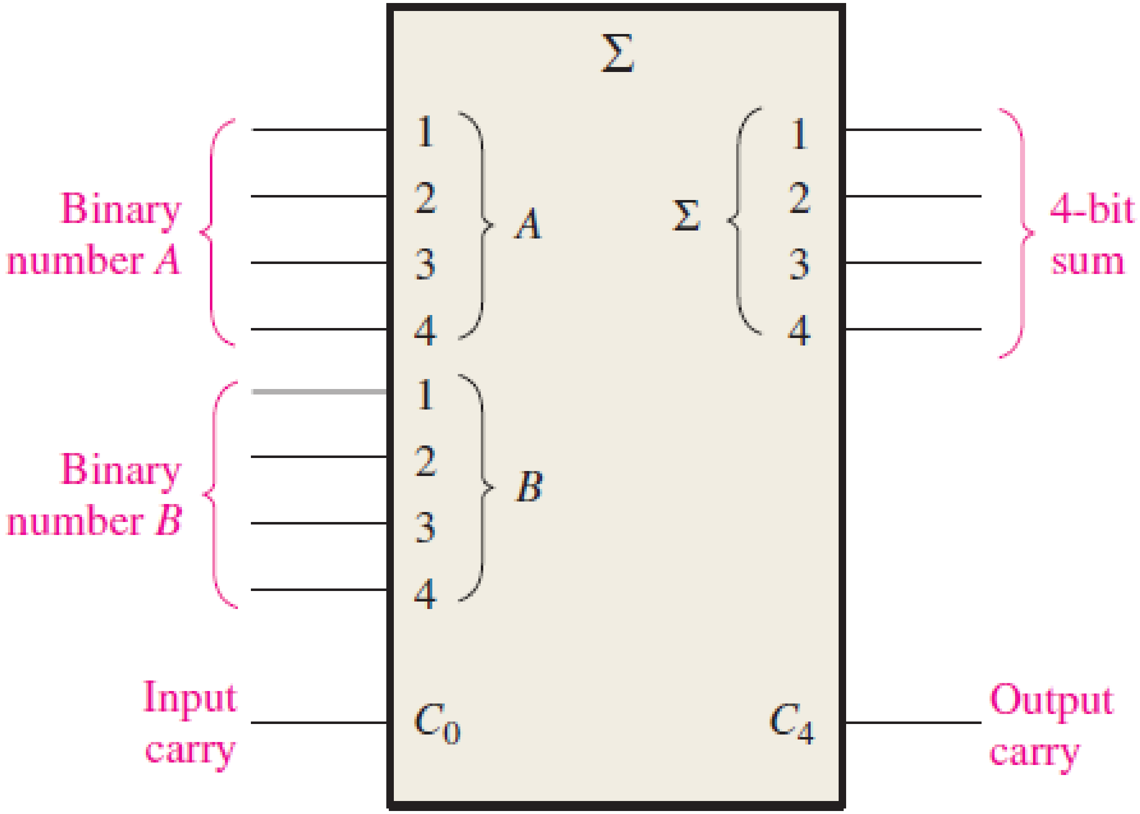
Parallel Adder

- A 4-bit Parallel adder to add two 4bit numbers can be designed using the same principle.



Adder Expansion

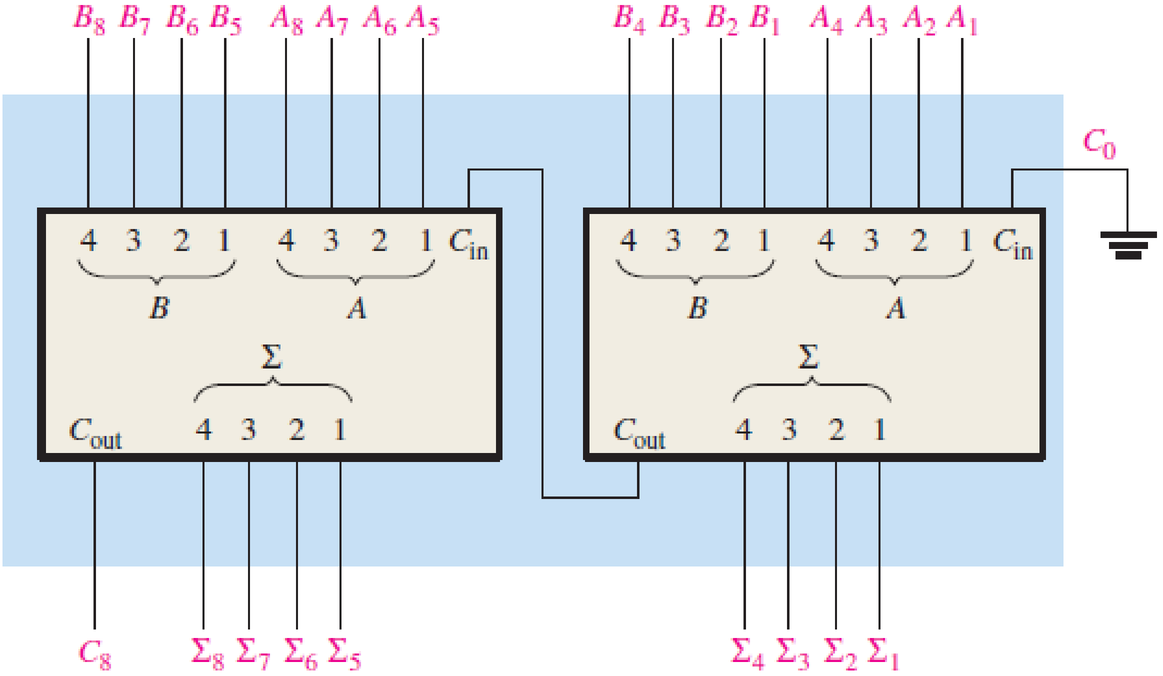
- We have learned how to add two 4-bit binary numbers.
- A 4-bit is called a nibble.
- So how do we add two numbers of 8-bits or even higher.
- That is when we use the concept of adder expansion.
- The following is the logic symbol used to represent a 4-bit parallel adder.



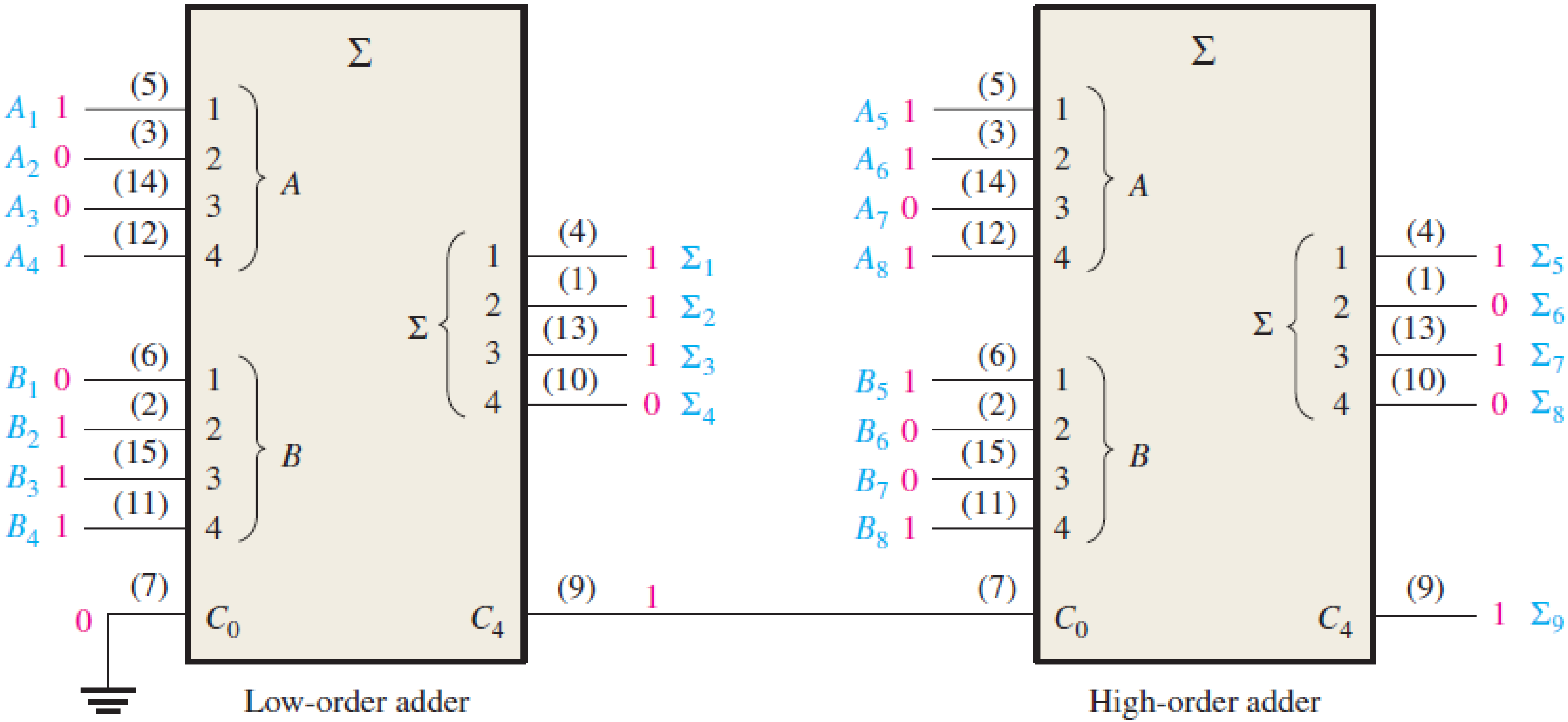
Adder Expansion

- We can cascade two blocks of 4-bit parallel adder to add two numbers of 8-bits.

		1	1	1					
	1	0	1	1		1	0	0	1
	1	0	0	1		1	1	1	0
1	0	1	0	1		0	1	1	1
C	S_8	S_7	S_6	S_5		S_4	S_3	S_2	S_1



Example of Adder Expansion



Subtractor

- A subtractor is a combinational circuit that take two numbers as input and produces their difference.
- It also has an output to specify if a 1 has been borrowed.
- The minuend is designated by **x** and the subtrahend is designated by **y**.
- Like adders, subtractors are also of two types namely, Half-subtractor and Full-subtractor.
- The rules of binary subtraction are:

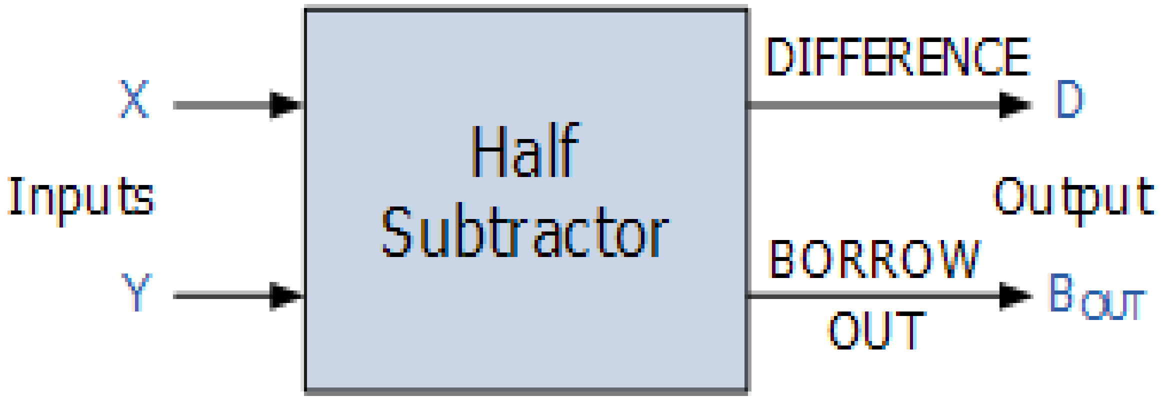
X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

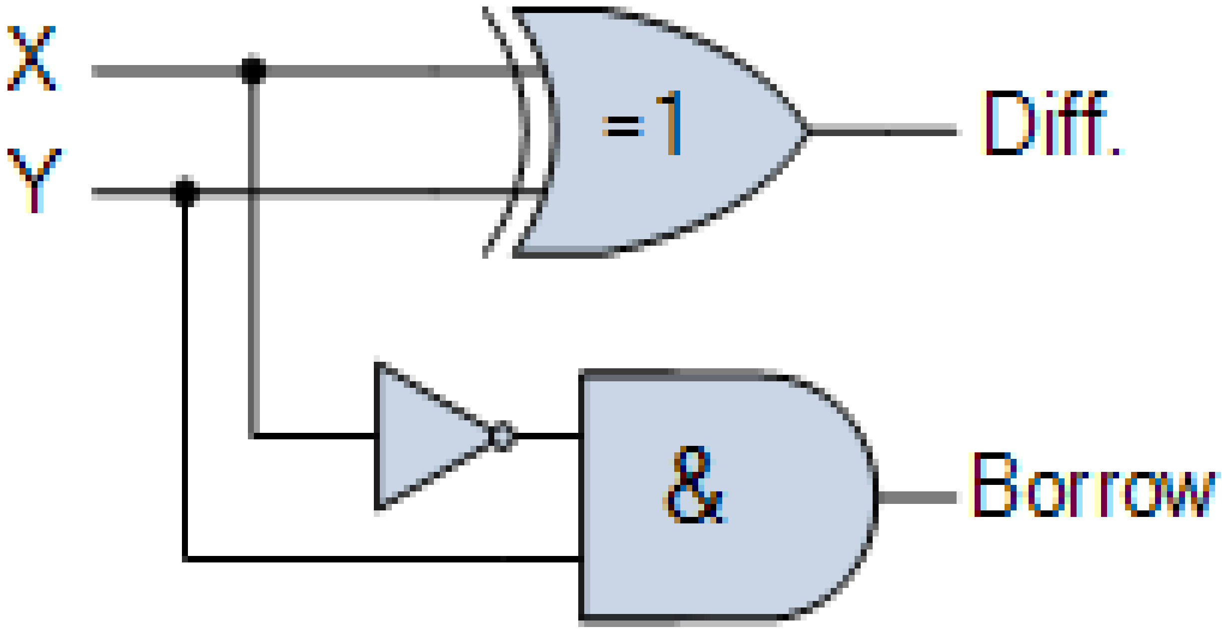
Half-Subtractor

- A half-subtractor take two binary bits as input and outputs the difference and borrow.
- The truth table for a half-subtractor is as follows:

X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$\mathbf{D = \overline{X}Y + X\overline{Y} = X \oplus Y}$$
$$\mathbf{B = \overline{X}Y}$$

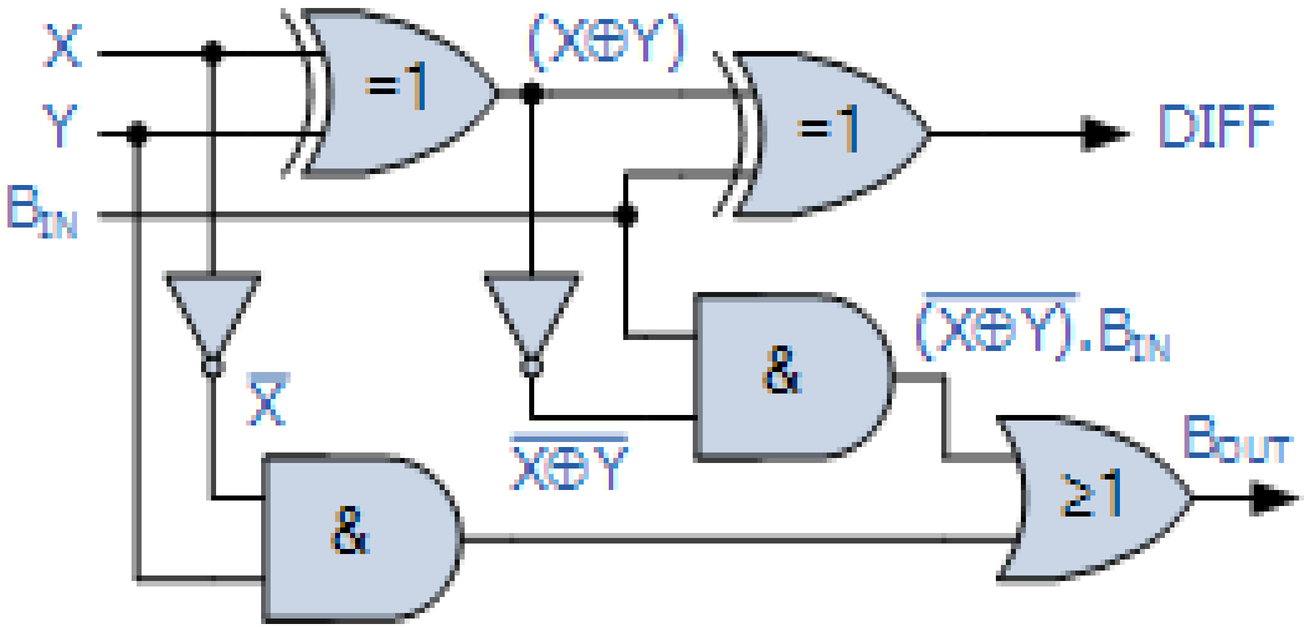




Full-Subtractor

- A full-subtractor take the binary bits as input and outputs the difference and borrow.
- The truth table for a full-subtractor is as follows:

X	Y	B_{IN}	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

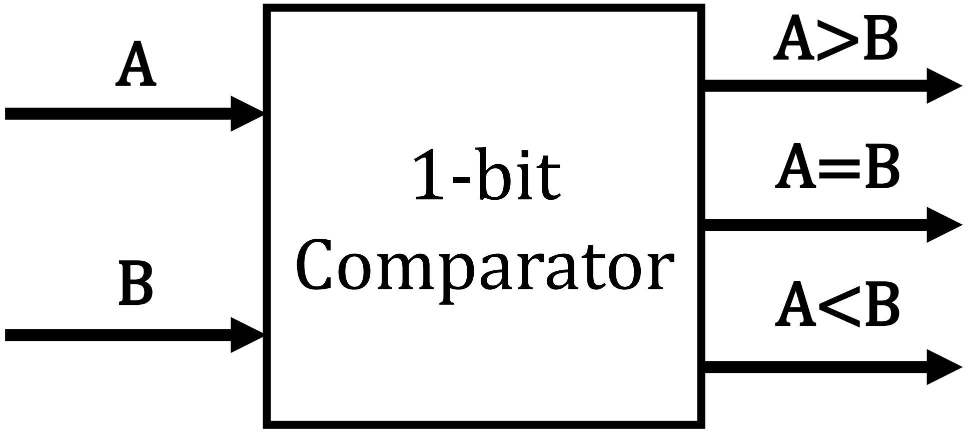


$$D = X \oplus Y \oplus B_{IN}$$

$$B = \bar{X}Y + \overline{(X \oplus Y)}B_{IN}$$

Digital Magnitude Comparator

- A comparator is a combinational circuit which can be used to compare between two number.
- A magnitude comparator has three possible outputs; A is greater than B, A is equal to B and A is less than B.
- The truth-table for a 1-bit comparator can be constructed as:



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

The Boolean expression for the outputs:

$$F_{A=B} = \bar{A}\bar{B} + AB$$
$$F_{A<B} = \bar{A}B$$
$$F_{A>B} = A\bar{B}$$

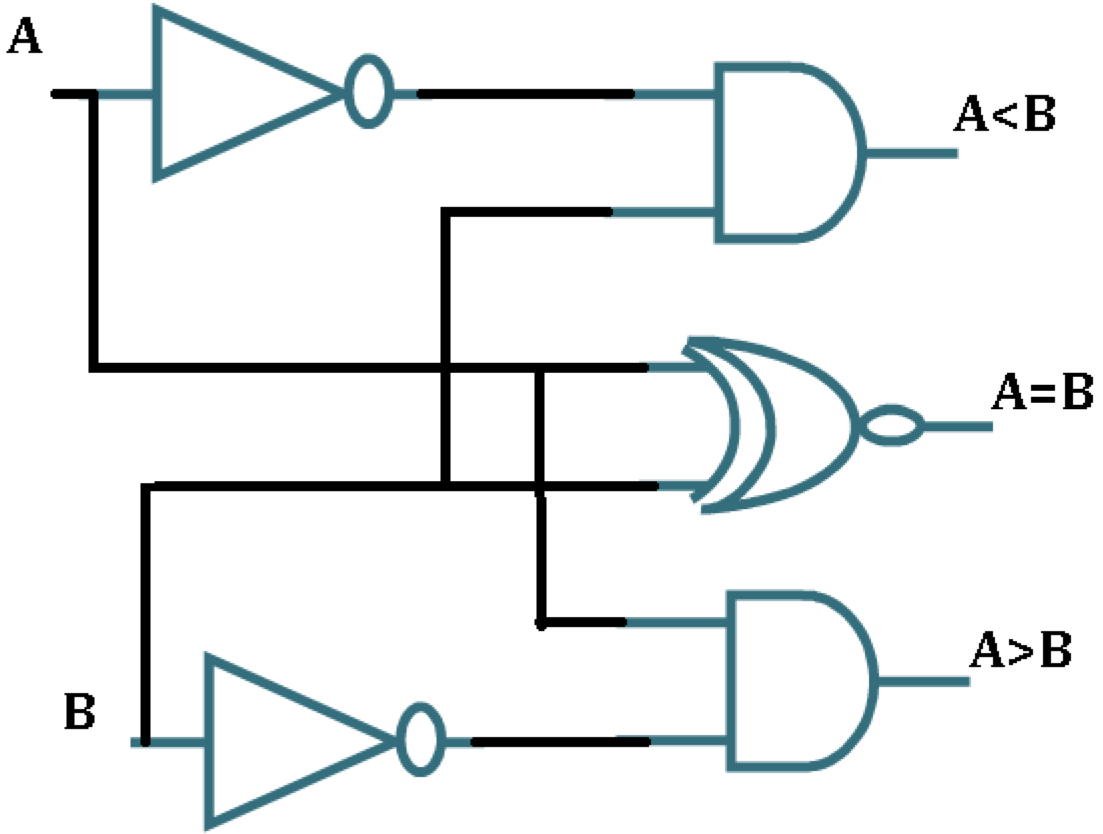
Digital Magnitude Comparator

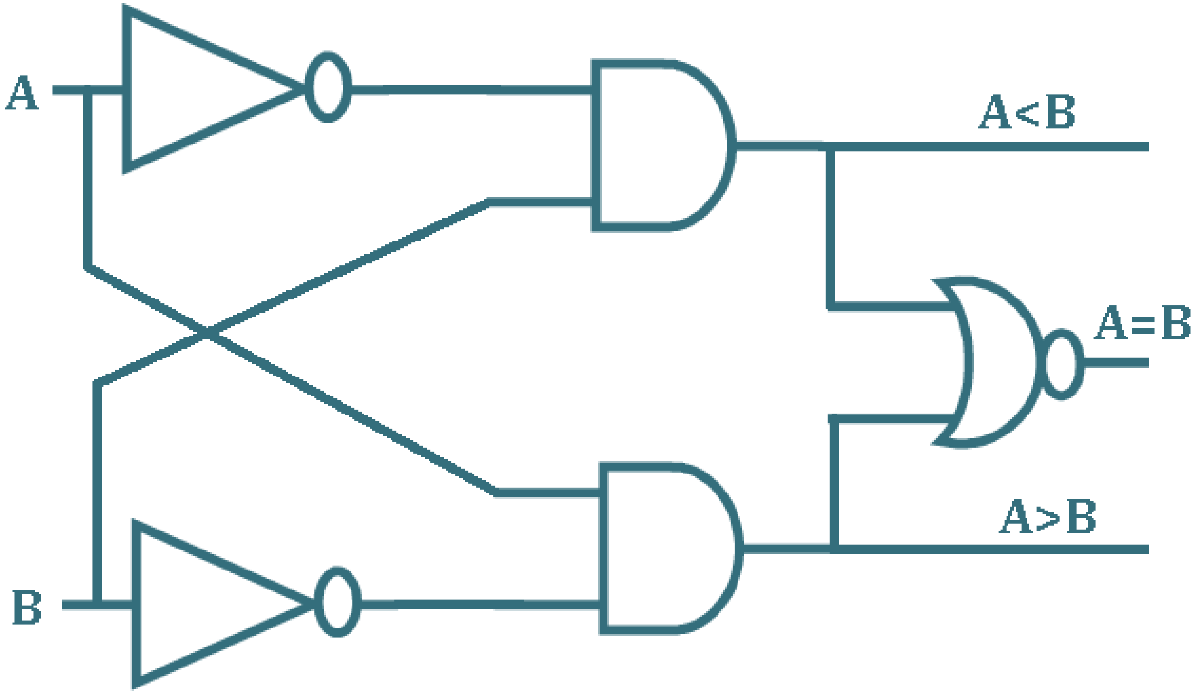
The Boolean expression for the outputs:

$$F_{A=B} = \bar{A}\bar{B} + AB = \overline{A \oplus B}$$

$$F_{A<B} = \bar{A}B$$

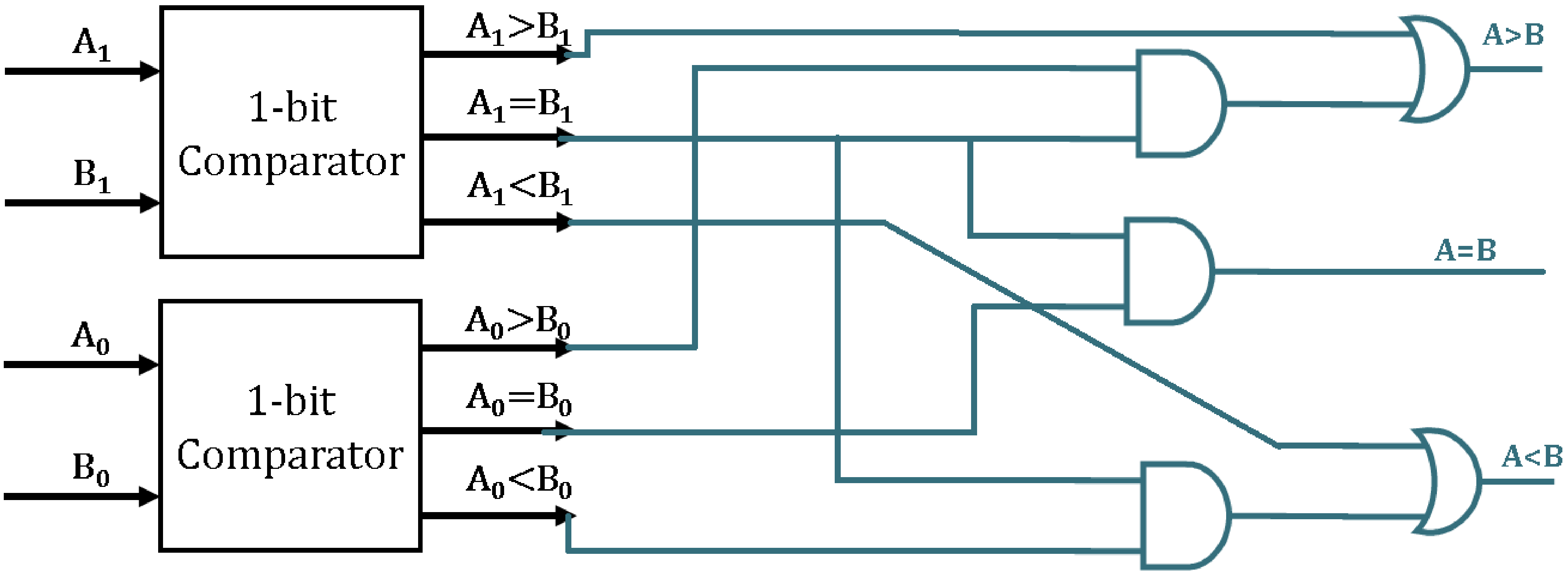
$$F_{A>B} = A\bar{B}$$





Digital Magnitude Comparator

- Two 1-bit comparators can be logically connected to make a 2-bit number comparator.
- The logic behind the connections are:
  - Let $\mathbf{X_0} = (\mathbf{A_0} == \mathbf{B_0})$ and $\mathbf{X_1} = (\mathbf{A_1} == \mathbf{B_1})$
  - For $\mathbf{A} = \mathbf{B}$, the Boolean expression is X_1X_0
  - For $\mathbf{A} > \mathbf{B}$, the Boolean expression is $\mathbf{A_1\overline{B_1}} + \mathbf{X_1(A_0\overline{B_0})}$
  - For $\mathbf{A} < \mathbf{B}$, the Boolean expression is $\mathbf{\overline{A_1}B_1} + \mathbf{X_1(\overline{A_0}B_0)}$



References

1. Thomas L. Floyd, “Digital Fundamentals” 11th edition, Prentice Hall – Pearson Education.

Thank You