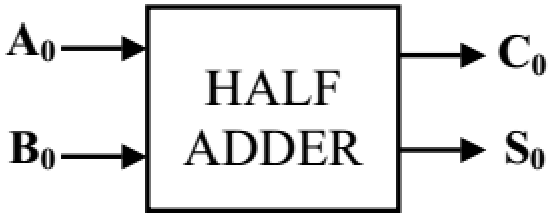
CHAPTER 5: LOGIC CIRCUITS

I. HALF ADDER

The half adder: to add 2 least significant bits (LSB) we do not need a carry input from a previous stage. >> We only need a half adder this will have two inputs A_0, B0 and two outputs S_0 and C_0



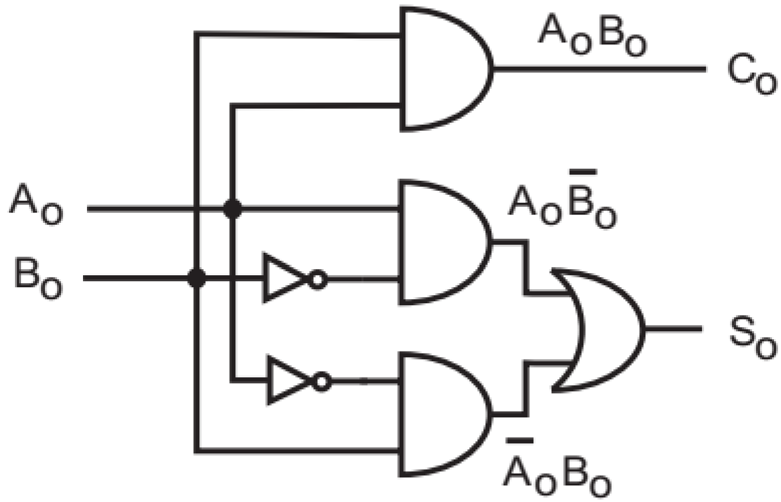
A_0	B_0	S_0	C_0
0	0	**0**	**0**
0	1	**1**	**0**
1	0	**1**	**0**
1	1	**0**	**1**

S_0: sum out
C_0: carry out

The **logic** equations are:

$$\boldsymbol{S0} = \overline{A_0}B_0 + A_0\overline{B_0} = A \oplus B > (XOR\ Gate)$$
$$\boldsymbol{C0} = A_0B_0 > (AND\ Gate)$$

The **logic diagram** can also be expressed as follows:



II. FULL ADDER

a. Using normal method

For all other bits (except the LSB) a half adder **will not** suffice because there **may be a carry input** from a previous stage.
A full adder has **3 inputs**: **A_K**, **B_K , C_{K-1}** AND **2 OUTPUTS**: **S_K**, **C_K C_{K-1} =** carry in from the **previous** stage **C_K**= carry out to the **next** stage

A_k	B_k	C_{k-1}	S_k	C_k
0	0	0	**0**	**0**
0	0	1	**1**	**0**
0	1	0	**1**	**0**
0	1	1	**0**	**1**
1	0	0	**1**	**0**
1	0	1	**0**	**1**
1	1	0	**0**	**1**
1	1	1	**1**	**1**

The **logic** equations are:

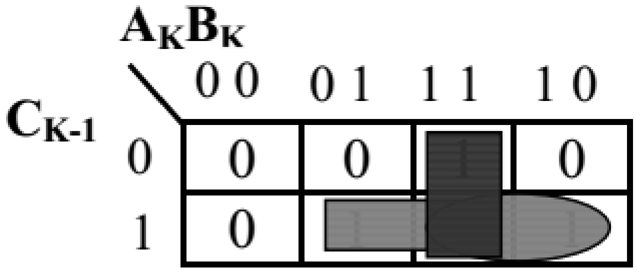
$$\boldsymbol{S_K} = \bar{A}_K\bar{B}_K C_{K-1} + \bar{A}_K B_K \bar{C}_{K-1} + A_K\bar{B}_K\bar{C}_{K-1} + A_K B_K C_{K-1}$$
$$\boldsymbol{C_K} = \bar{A}_K B_K C_{K-1} + A_K\bar{B}_K C_{K-1} + A_K B_K C_{K-1} + A_K B_K \bar{C}_{K-1}$$

K-MAP FOR **S_K:**

C_{K-1} \ A_KB_K	0 0	0 1	1 1	1 0
0	0	1	0	1
1	1	0	1	0

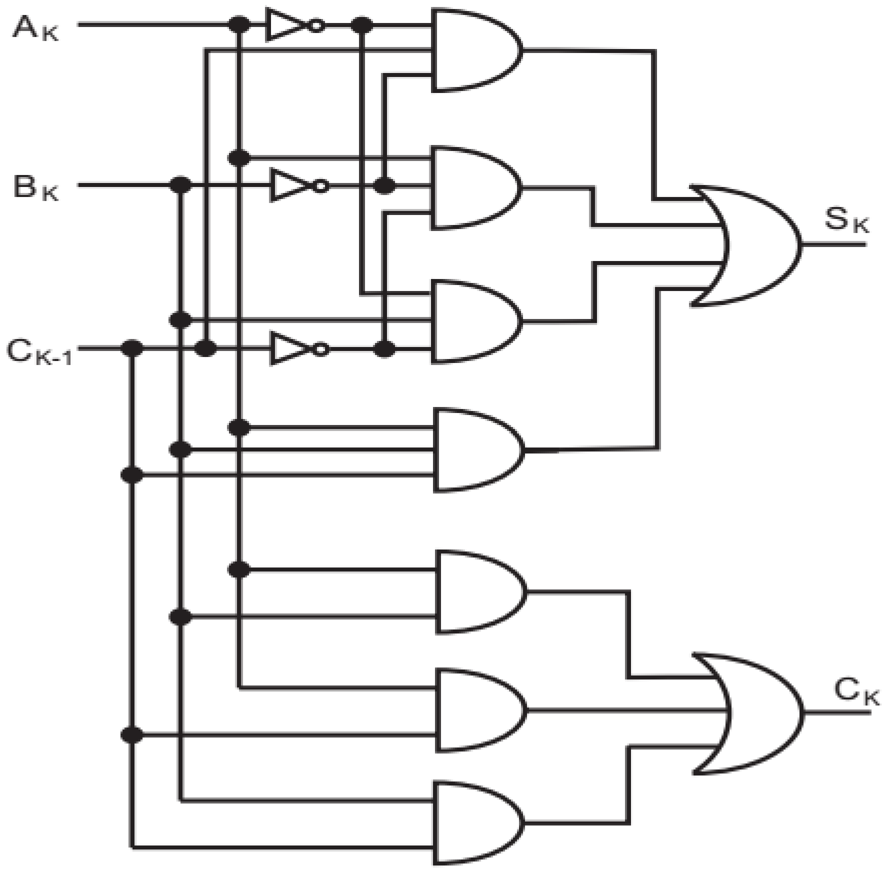
No simplification possible

k-map for **C_K**



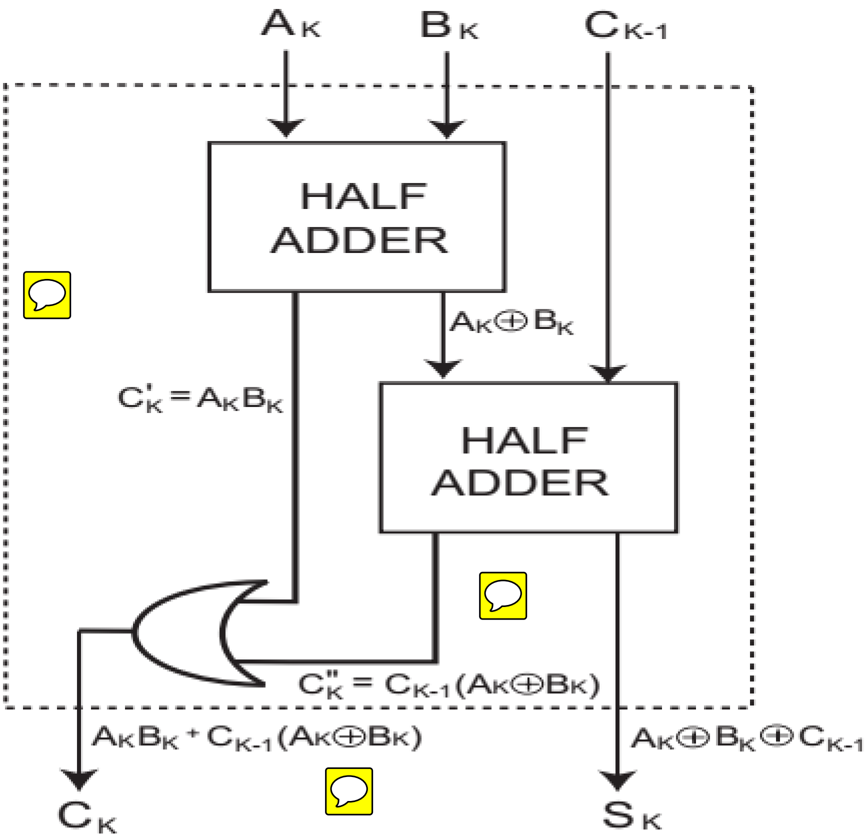
$$\boldsymbol{C_K} = B_K C_{K-1} + A_K C_{K-1} + A_K C_{K-1}$$

The **logic** diagram for a **full adder** is:



b. **Full adder using half adders**

One half adder adds $\mathbf{A_K}$ to $\mathbf{B_K}$ to give an intermediate sum S'_K and carry C'_K.
Another half adder adds S'_K and C_{K-1} to give the final sum S_K and another intermediate carry C''_K
there will be a final carry if either C'_K or C''_K are "**1**"



Remember the Boolean operation for **half adder**:
$S'_K = A_K \oplus B_K$
$C'_K = A_K B_K$

For full adder:

$S_K = A_K \oplus B_K \oplus C_{K-1}$
$C_K = A_K B_K + C_{K-1}(A_K \oplus B_K)$

PROOF:

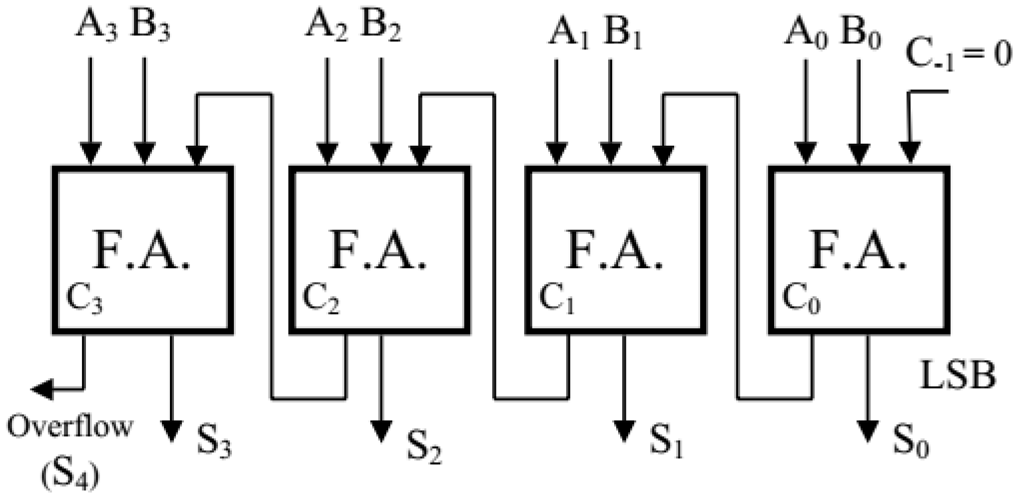
$$\boldsymbol{S_K = \bar{A}_K \bar{B}_K C_{K-1} + \bar{A}_K B_K \bar{C}_{K-1} + A_K \bar{B}_K \bar{C}_{K-1} + A_K B_K C_{K-1}}$$
$$= C_{K-1}(\bar{A}_K \bar{B}_K + \boldsymbol{A_K B_K}) + \bar{C}_{K-1}(\bar{A}_K B_K + \boldsymbol{A_K \bar{B}_K})$$
$$= C_{K-1}\left(\overline{A_K \oplus B_K}\right) + \bar{C}_{K-1}(A_K \oplus B_K)$$
$$= A_K \oplus B_K \oplus C_{K-1}$$

$$\boldsymbol{C_K = \bar{A}_K B_K C_{K-1} + A_K \bar{B}_K C_{K-1} + A_K B_K \bar{C}_{K-1} + A_K B_K C_{K-1}}$$
$$= \boldsymbol{A_K B_K}(\boldsymbol{C_{K-1}} + \bar{C}_{K-1}) + C_{K-1}(\bar{A}_K B_K + \boldsymbol{A_K \bar{B}_K})$$
$$= \boldsymbol{A_K B_K} + C_{K-1}(A_K \oplus B_K)$$

III. THE PARALLEL ADDER:

Also called ripple carry adder used to add **two n-bit** numbers. it consists of n full adders where the carry output of each stage is the carry in of the next stage.

Example: 4-bit parallel adder add $\mathbf{A_3A_2A_1A_0}$ AND $\mathbf{B_3B_2B_1B_0}$



$C_3 = 1$ >>> over flow

E.G. $A = \mathbf{1010}$ (10_{10})

$B = \underline{\mathbf{1001}}$ (9_{10})

Overflow $\leftarrow$ (**1**)0011 $\rightarrow$ 3_{10}