Binary Adders (Half adder and full adder)

Half Adder:

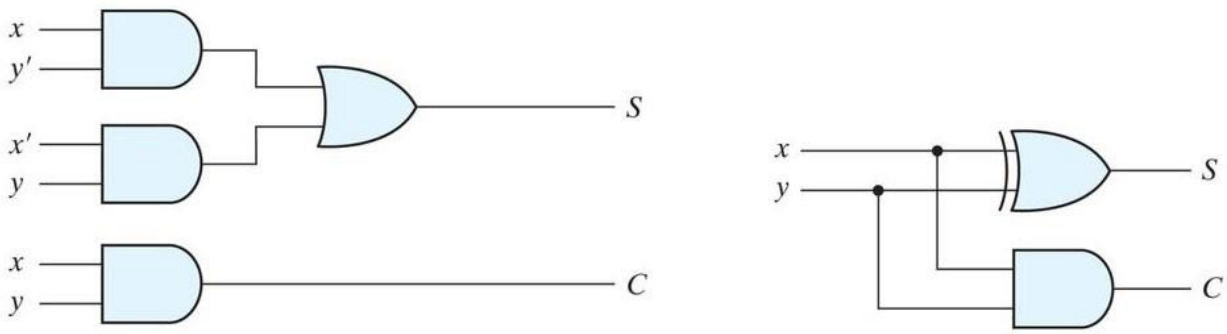
Specification: 2 inputs (X,Y)
2 outputs (C,S)

From the verbal explanation of a half adder, we find that this circuit needs two binary inputs and two binary outputs. The input variables designate the augend and addend bits; the output variables produce the sum and carry. We assign symbols x and y to the two inputs and S (for sum) and C (for carry) to the outputs. The truth table for the half adder is listed in Table.

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The C output is 1 only when both inputs are 1. The S output represents the least significant bit of the sum.

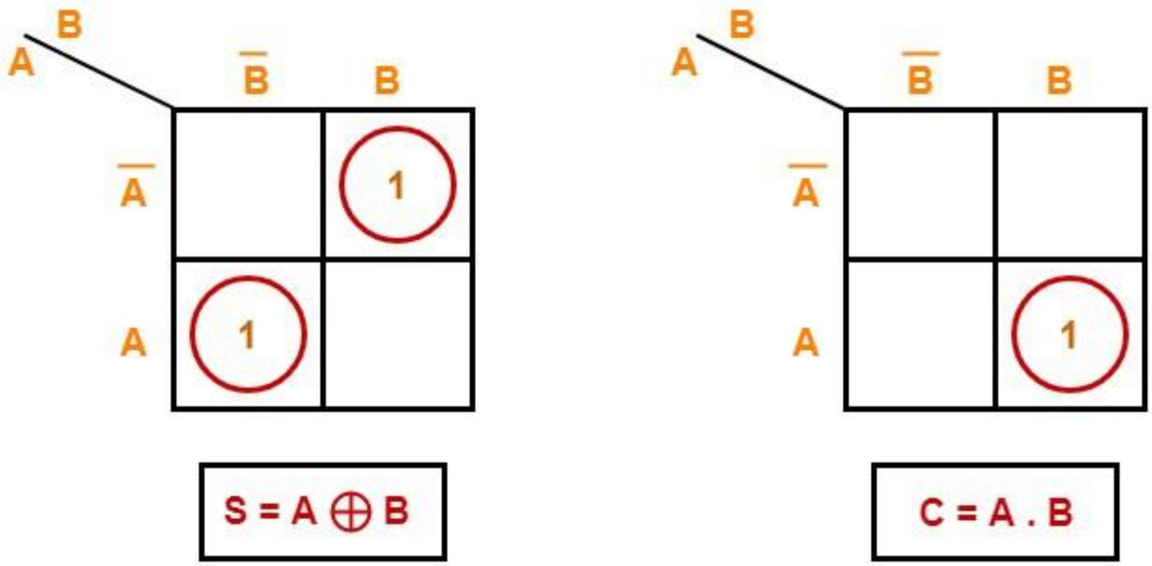
FIGURE 4.5 Implementation of half adder



(a) $S = xy' + x'y$
$C = xy$

(b) $S = x \oplus y$
$C = xy$

For S: For C:



K Maps

Adder

Design an Adder for 1-bit numbers?

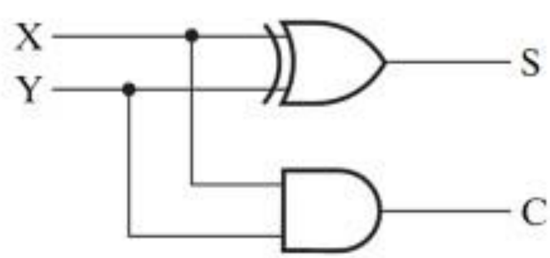
1. Specification:

2 inputs (X,Y)
2 outputs (C,S)

2. Formulation:

X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

3. Optimization/Circuit



Full Adder

A combinational circuit that adds 3 input bits to generate a Sum bit and a Carry bit

X	Y	Z	C	S
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

Sum

X \ YZ	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$S = X'Y'Z + X'YZ'$
$+ XY'Z' + XYZ$
$= X \oplus Y \oplus Z$

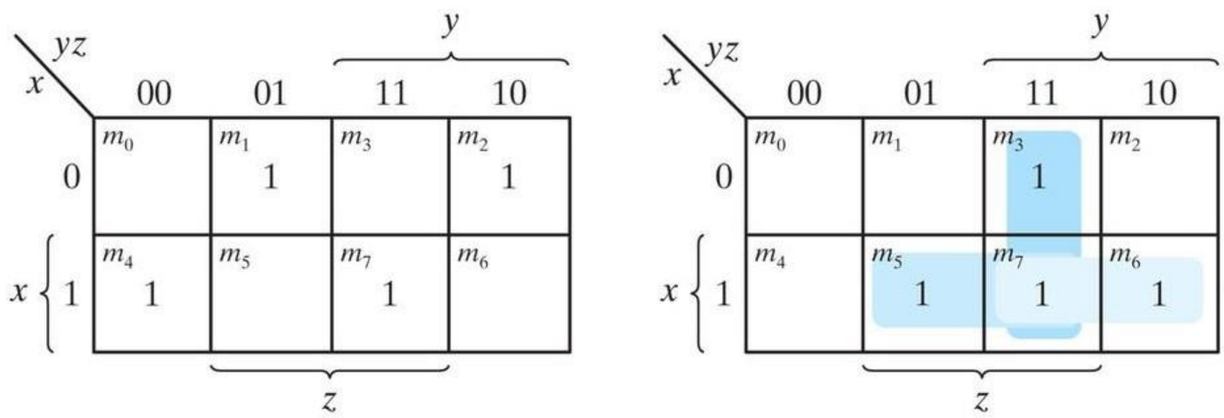
Carry

X \ YZ	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$C = XY + YZ + XZ$

x	*y*	*z*	*C*	*S*
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

FIGURE 4.6 K-Maps for full adder



(a) $S = x'y'z + x'yz' + xy'z' + xyz$

(b) $C = xy + xz + yz$

FIGURE 4.7 Implementation of full adder in sum-of-products form

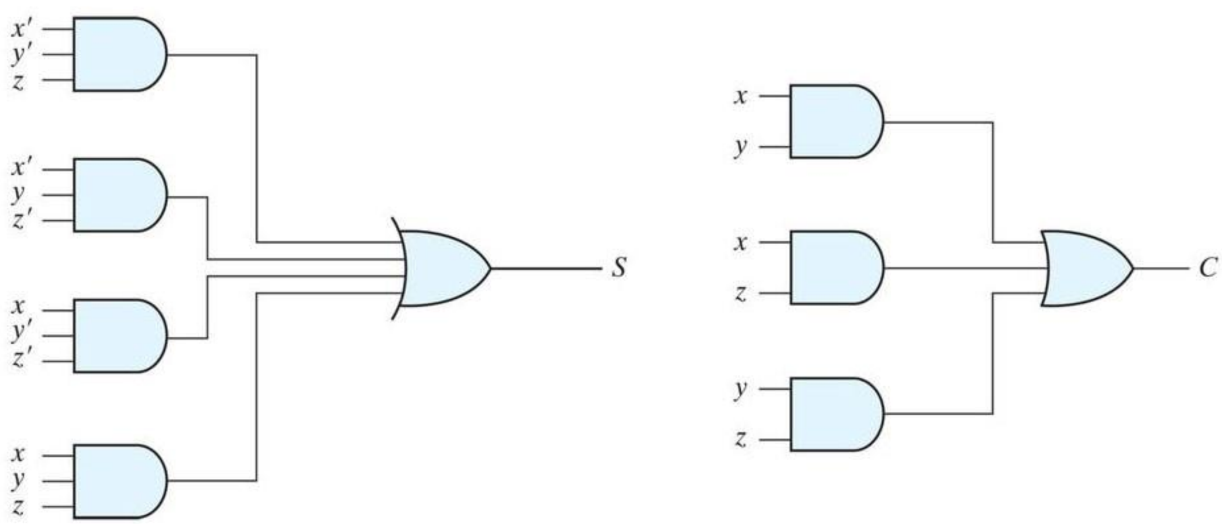
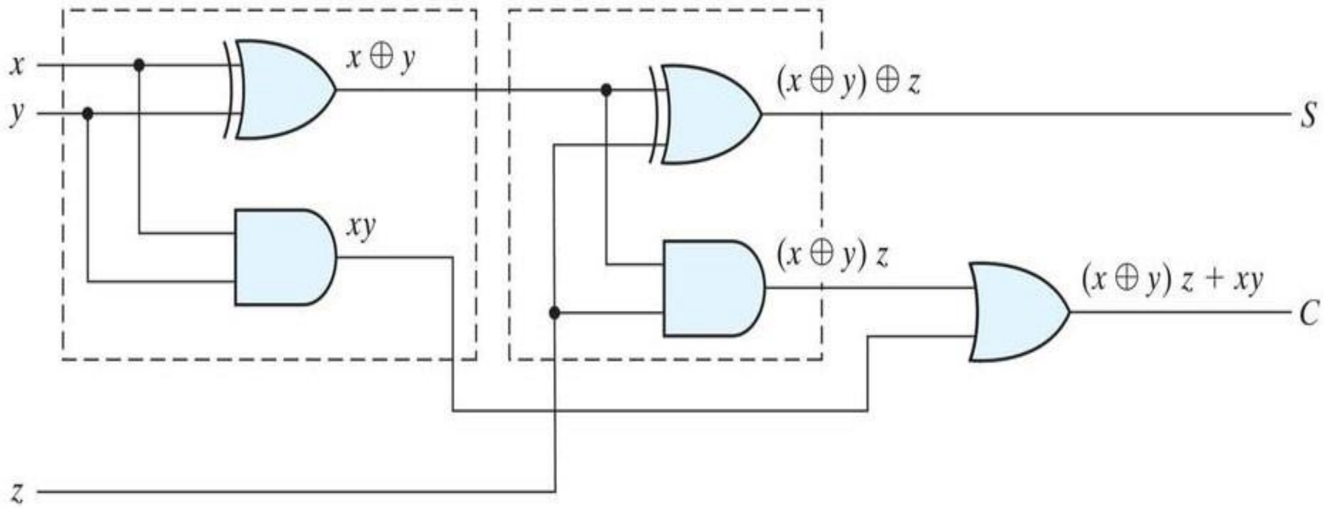


FIGURE 4.8 Implementation of full adder with two half adders and an OR gate



Full Adder = 2 Half Adders

Manipulating the Equations:

$S = (X \oplus Y) \oplus Z$

$C = XY + XZ + YZ$

$= XY + XYZ + XY'Z + X'YZ + XYZ$

$= XY(1 + Z) + Z(XY' + X'Y)$

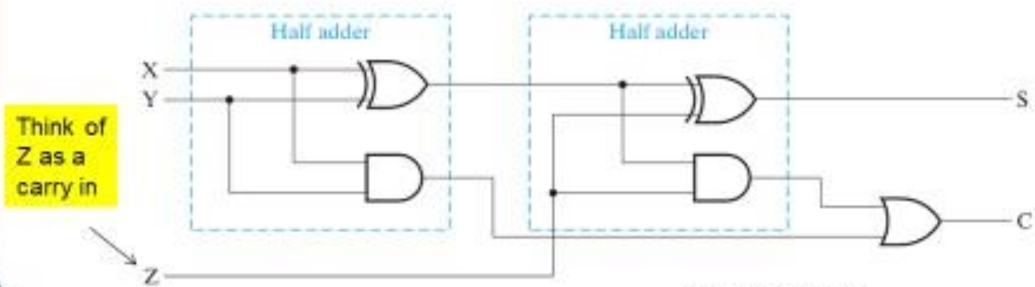
$= XY + Z(X \oplus Y)$

Full Adder = 2 Half Adders

Manipulating the Equations:

$S = (X \oplus Y) \oplus Z$

$C = XY + XZ + YZ = XY + Z(X \oplus Y)$

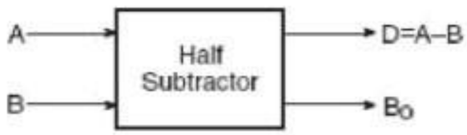


Src: Mano's Book

Binary Subtractors (Half and full Subtractor)

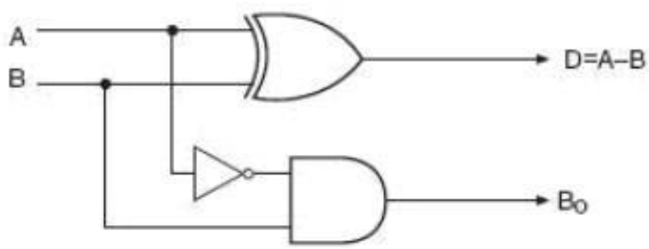
$$D = \overline{A}.B + A.\overline{B}$$

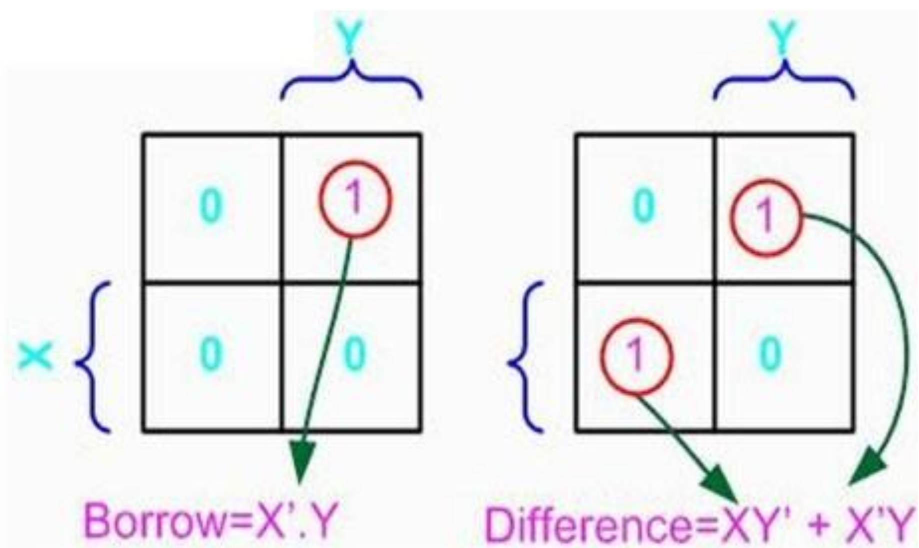
$$B_o = \overline{A}.B$$



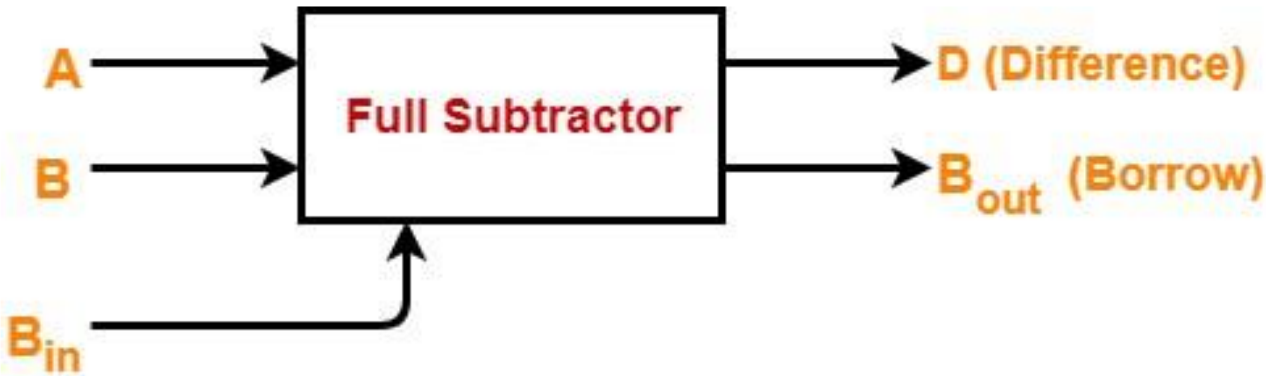
A	B	D	B_O
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Half Subtractor





Full Subtractor



Full Subtractor-Truth Table				
Input			Output	
A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

For D

$A \backslash BB_{in}$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$D = \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in} + ABB_{in}$

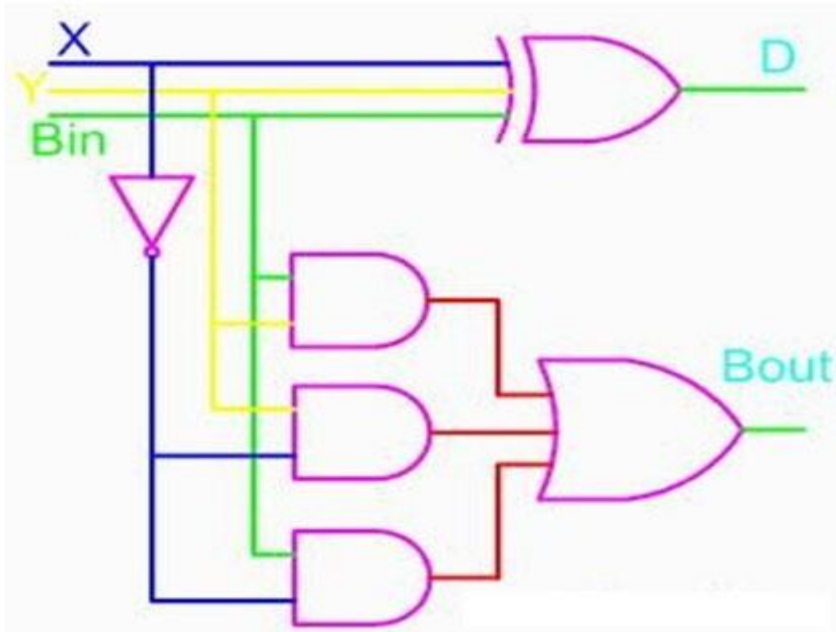
For B_{out}

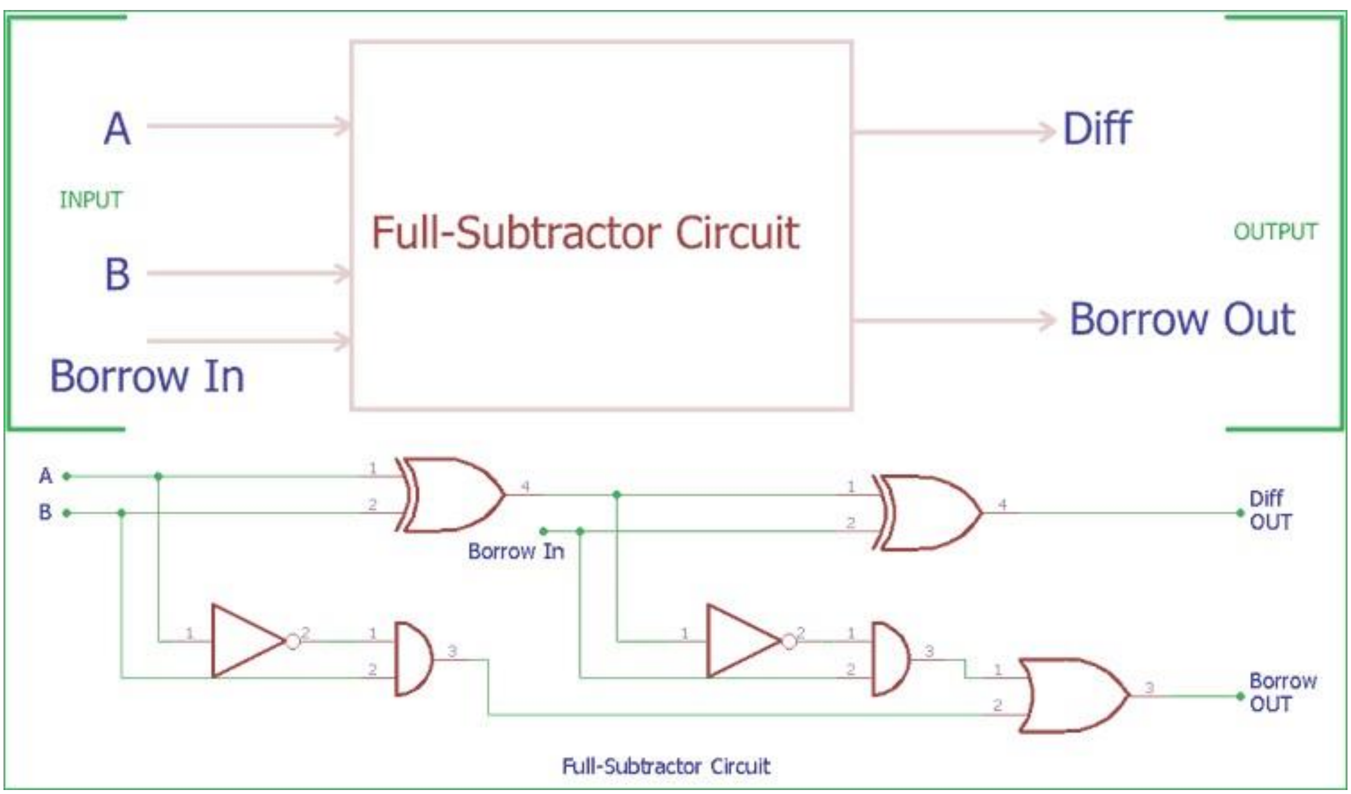
$A \backslash BB_{in}$	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$B_{out} = \bar{A}B_{in} + \bar{A}B + BB_{in}$

Fig. 3.21 Maps for full-subtractor

$$
\begin{aligned}
Difference &= \bar{A}\,\bar{B}\,C + \bar{A}\,B\,\bar{C} + A\,\bar{B}\,\bar{C} + ABC \\
&= C\,(\bar{A}\,\bar{B} + AB) + \bar{C}\,(\bar{A}\,B + A\,\bar{B}) \\
&= C\,(A \odot B) + \bar{C}\,(A \oplus B) \\
&= C\,(\overline{A \oplus B}) + \bar{C}\,(A \oplus B) \\
&= C \oplus (A \oplus B)
\end{aligned}
$$


X
Y
Bin
D
Bout


A
INPUT
B
Borrow In
Full-Subtractor Circuit
Diff
OUTPUT
Borrow Out
A
B
Borrow In
Diff OUT
Borrow OUT
Full-Subtractor Circuit