Sairam INSTITUTIONS

Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

YEAR	SEM
II	III

CS8351
DIGITAL PRINCIPLES AND SYSTEM DESIGN

UNIT NO.2

2.4 SUBTRACTOR

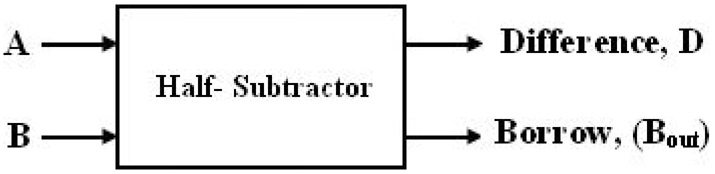
Version: 1.XX

4 QUALITY EDUCATION

NBA NAAC A+ nirf

Half -Subtractor:

A half-subtractor is a combinational circuit that can be used to subtract one binary digit from another to produce a DIFFERENCE output and a BORROW output. The BORROW output here specifies whether a ‗1‘ has been borrowed to perform the subtraction.

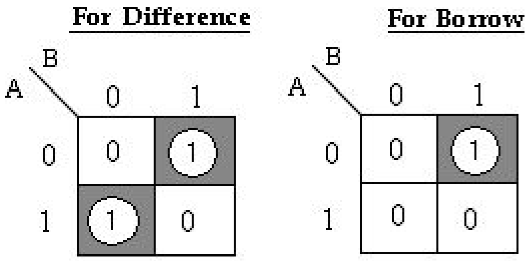


Block schematic of half-subtractor

The truth table of half-subtractor, showing all possible input combinations and the corresponding outputs are shown below.

Input		Output	
A	**B**	**Difference (D)**	**Borrow (Bout)**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-map simplification for half subtractor:



Difference = AB'+ A'B
= A⊕B

Borrow = $\overline{A}$.B

The Boolean expressions for the DIFFERENCE and

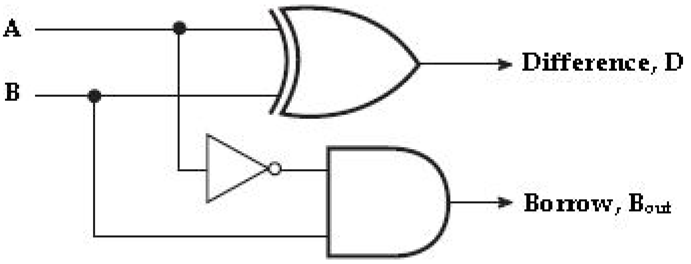
BORROW outputs are given by the equations,

Difference, D = A'B+ AB'= A $\oplus$ B Borrow,

Bout= A' . B

The first one representing the DIFFERENCE (**D**)output is that of an exclusive-OR gate, the expression for the BORROW output (**Bout**) is that of an AND gate with input A complemented before it is fed to the gate.

The logic diagram of the half adder is,



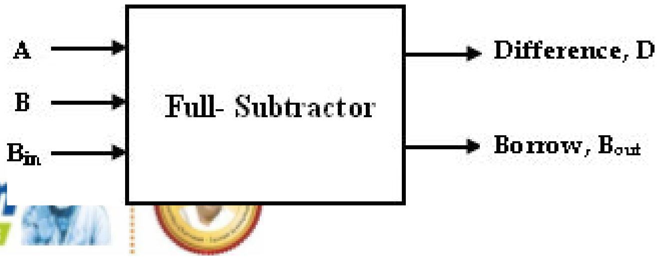
Logic Implementation of Half-Subtractor

Comparing a half-subtractor with a half-adder, we find that the expressions for the SUM and DIFFERENCE outputs are just the same. The expression for BORROW in the case of the half-subtractor is also similar to what we have for CARRY in the case of the half-adder. If the input A, ie., the minuend is complemented, an AND gate can be used to implement the BORROW output.

Full Subtractor:

A *full subtractor* performs subtraction operation on two bits, a minuend and a subtrahend, and also takes into consideration whether a'1'has already been borrowed by the previous adjacent lower minuend bit or not.

As a result, there are three bits to be handled at the input of a full subtractor, namely the two bits to be subtracted and a borrow bit designated as Bin. There are two outputs, namely the DIFFERENCE output D and the BORROW output Bo. The BORROW output bit tells

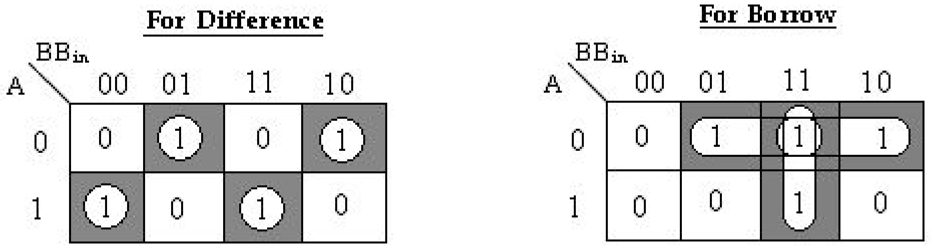


whether the minuend bit needs to borrow a _1‘ from the next possible higher minuend bit.

Block schematic of full-adder

The truth table for full-subtractor is,

Inputs			Outputs	
A	**B**	**Bin**	**Difference(D)**	**Borrow(Bout)**
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



Difference, D = $A'B'B_{in}$+ $A'BB'_{in}$+ $AB'B'_{in}$+ ABB_{in}

Borrow, B_{out} = $A'B$+ $A'B_{in}$ + BB_{in}

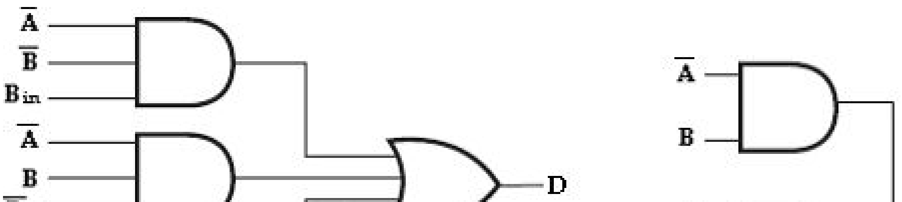
K-map simplification for full-subtractor:

The Boolean expressions for the DIFFERENCE and BORROW outputs are given by the equations,

Difference, D = A’B’Bin+ A’BB’in + AB’B’in + ABBin

Borrow, Bout= A’B+ A’Cin + BBin .

The logic diagram for the above functions is shown as,



Implementation of full-adder in Sum of Products

The logic diagram of the full-subtractor can also be implemented with two half- subtractors and one OR gate. The difference,D output from the second half subtractor is the exclusive-OR of Bin and the output of the first half-subtractor, giving

Difference,

$D= Bin \oplus (A \oplus B)$ $\quad [x \oplus y = x'y+ xy']$

$= Bin \oplus (A'B+AB')$

$= B'in\ (A'B+AB') + Bin\ (A'B+AB')'$ $\quad [(x'y+xy')'= (xy+x'y')]$

$= B'in\ (A'B+AB') + Bin\ (AB+A'B')$

$= A'BB'in + AB'B'in + ABBin + A'B'Bin$.

and the borrow output is,

Borrow, Bout = $A'B+ Bin\ (A'B+AB')'$ $\quad [(x'y+xy')'= (xy+x'y')]$

$= A'B+ Bin\ (AB+A'B')$

$= A'B+ ABBin+ A'B'Bin$

$= A'B\ (Bin+1) + ABBin+ A'B'Bin$ $\quad [Cin+1= 1]$

$= A'BBin+ A'B+ ABBin+ A'B'Bin$

$= A'B+ BBin\ (A+A') + A'B'Bin$ $\quad [A+A'= 1]$

$= A'B+ BBin+ A'B'Bin$

$= A'B\ (Bin+1) + BBin+ A'B'Bin$ $\quad [Cin+1= 1]$

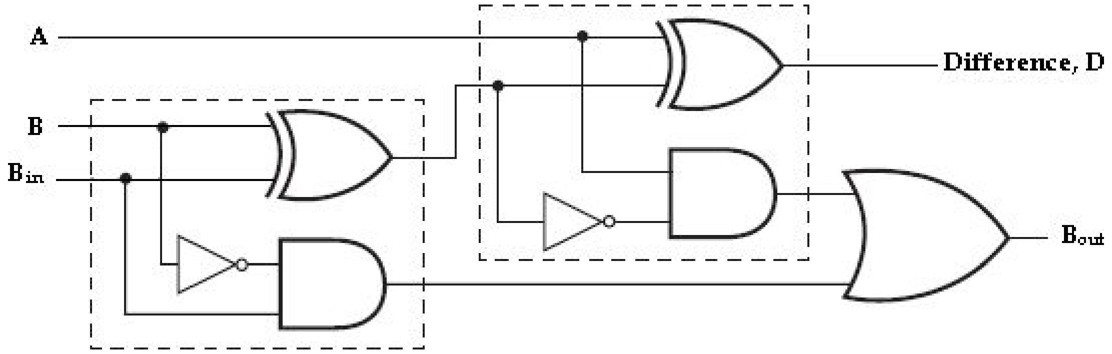
$= A'BBin+ A'B+ BBin+ A'B'Bin$

$= A'B+ BBin+ A'Bin\ (B +B')$

$= A'B+ BBin+ A'Bin.$

Therefore,

we can implement full-subtractor using two half-subtractors and OR gate as,



Implementation of full-subtractor with two half-subtractors and an OR gate