LAB 2 → Design and Implement Half Adder, Full Adder, Half Subtractor, Full Subtractor using basic gates and Implement the Same in HDL.

Adder

An adder is a combinational logic circuit in electronics that implements addition of numbers.

Adders are classified into 2 types:
1> half adder
2> full adder.

Half Adder: Adds two single binary digits A and B. It has 2 outputs, sum(S) and carry (C).
The carry signal represents an overflow into the next digit of multi-digit addition.

Truth - Table:

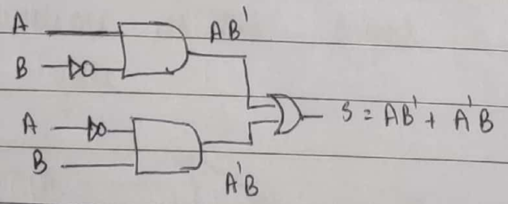
Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

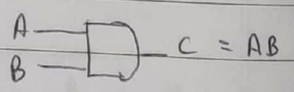
K-MAP: Sum

B \ A	0	1
0	0: 0	2: 1
1	1: 1	3: 0

B \ A	0	1
0	0:	2:
1	1:	3: 1

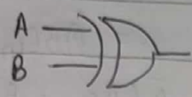
$sum = AB' + A'B = A \oplus B$ $\qquad$ $carry = AB$





Circuit Diagram for Sum $\qquad$ Circuit Diagram for carry.

Half Adder implementation using AND - XOR.



$AB' + A'B = Sum$

$AB = Carry$

Full Adder: Adds binary numbers and accounts for values carried in as well as out.

Difference between half-adder and a full adder is that the full-adder has 3 inputs and 2 outputs

A 1-bit full adder adds three one-bit numbers, often written as A, B and C_{in}. It has 2 outputs, sum (S) and carry (C).

Truth-Table.

Inputs			Outputs	
A	B	C_{in}	Carry	Sum
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

K-MAP for full Adder

BC_{in} \ A	0	1
00	0	4
01	1	5 1
11	3 1	7 1
10	2	6 1

BC_{in} \ A	0	1
00	0	4 1
01	1 1	5
11	3	7 1
10	2 1	6

Carry $= AB + \bar{A}BC_{in} + A\bar{B}C_{in}$ Sum $= A \oplus B \oplus C_{in}$

$= AB + C_{in}(\bar{A}B + A\bar{B})$

$= AB + C_{in}(A \oplus B)$

$= \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + \bar{A}B\bar{C}$

$= \bar{A}(\bar{B}C_{in} + B\bar{C}_{in}) + A(\bar{B}\bar{C}_{in} + BC_{in})$

$= A \oplus B \oplus C_{in}$

Subtractor

A subtractor is a digital logic circuit in electronics that implements subtraction of numbers.

Subtractor are classified into 2 types:

1> half subtractor

2> full subtractor.

Half subtractor : The half subtractor subtracts 2 single binary digits A and B.

It has 2 outputs, Difference(D) and Borrow(B)

The borrow signal represents an overflow into the next digit of a multi-digit subtraction.

Truth-Table.

Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-MAP:

Difference

B \ A	0	1
0	⁰	² 1
1	¹ 1	³

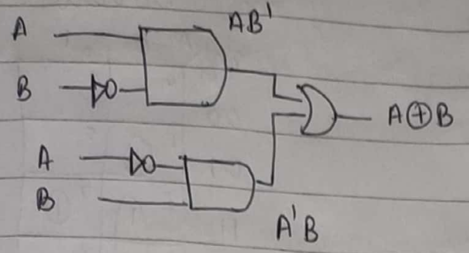
Borrow

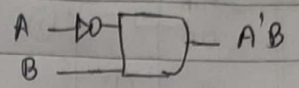
B \ A	0	1
0	⁰	²
1	¹ 1	³

$B = \bar{A}B$

$Diff = AB' + A'B = A \oplus B$

Circuit Diagram using Basic Gate:

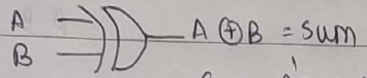


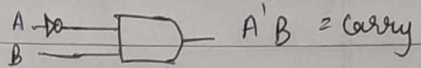


carry.

Sum

Circuit Diagram using EX-OR





Sum = A'B + AB'

Full Subtractor: Subtracts binary numbers and accounts for values borrowed in as well as out.

Difference between half-subtractor and a full-subtractor is that the full-subtractor has 3 inputs and 2 outputs

A one-bit full subtractor subtracts three one-bit numbers, often written as A, B and C_{in}.
It has 2 output Difference (D) and Borrow (B)

Truth Table

Inputs			Outputs	
A	B	C_{in}	Borrow	Difference
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

K-MAP

Difference

BC_{in} \ A	0	1
00	0:	4: 1
01	1: 1	5:
11	3:	7: 1
10	2: 1	6:

Borrow

BC_{in} \ A	0	1
00	0:	4:
01	1: 1	5:
11	3: 1	7: 1
10	2: 1	6:

$Diff = A \oplus B \oplus C_{in}$

$Br = \bar{A}B + \bar{A}\bar{B}C_{in} + ABC_{in}$

$= \bar{A}B + C_{in}(\bar{A}\bar{B} + AB)$

$= \bar{A}B + C_{in}(\overline{A \oplus B})$

VHDL

→ A hardware description language allows a digital system to be designed debugged at a higher-level before implementation at the gate level.

→ The two most popular hardware description languages are VHDL and verilog.

VHDL	verilog
ADE based	C-based
More powerfull	less
Difficult to learn	Easy

VHDL → Can describe digital system at several different levels - Behavioral level
Dataflow level
Structural level

Behavioral level: In terms of its function of adding two binary numbers without giving any implementation details.

Dataflow level - by giving the logic equations.

Structural level - by specifying the inter-connections of gates which makeup the adder.

Xilinx ISE (Integrated Synthesis Environment)

→ Is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs perform timing analysis simulate a design's reaction to different stimuli and configure the target device with the programmer.