# DAY 7

# FULL SUBTRACTOR

## Theory

A Full Subtractor is a combinational logic circuit that is designed to perform the subtraction of two binary numbers. It is considered "full" because it not only considers the subtraction of two binary inputs but also accounts for a borrow input from the previous stage. In other words, it can subtract three binary bits - minuend, subtrahend, and borrow - to produce the difference and a borrow output.

### The Inputs

A Full Subtractor typically has three inputs and two outputs:

A (Minuend): This is the binary number from which you want to subtract.

B (Subtrahend): This is the binary number that you want to subtract from the minuend.

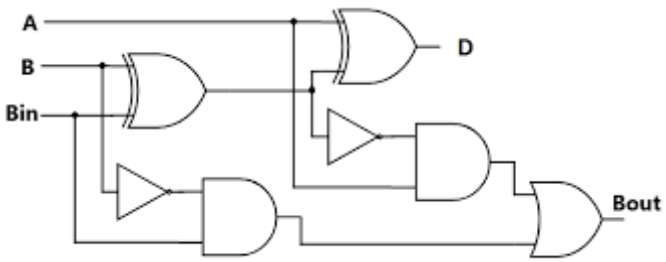
B_in (Borrow In): This input considers any borrow from the previous stage.

### The outputs :

Difference (D): This is the result of the subtraction of A and B.

Borrow Out (B_out): This output provides the borrow for the next stage in case it's needed.

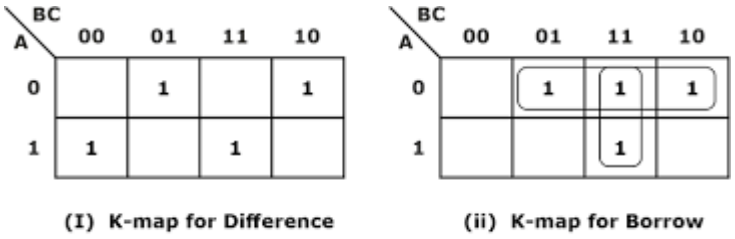
## Circuit Diagram :



Truth Table

| A | B | Bin = C | Diff | 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 |

K -Map :




(I) K-map for Difference (ii) K-map for Borrow

K-maps for full subtractor

**Therefore the expression is :**

**Difference = AB'C'+ABC+A'B'C+A'BC'**

**Borrow = A'C+BC+A'B**

**Verilog code for Full Subtractor using structural or gate level:**

```
module Full_Substractor(A,B,Bin,Dout,Bout);
 input A,B,Bin;
 output Dout,Bout;
 wire w1,w2,w3,w4,w5;
 xor gate1(w1,A,B);
 xor gate2(Dout,w1,Bin);
 not gate3(w3,A);
 not gate4(w4,w1);
 and gate5(w5,w3,B);
 and gate6(w6,w4,Bin);
 or gate7(Bout,w5,w6);
endmodule
```

**Dataflow :**

```
module Full_Sub(A,B,Bin,Dout,Bout);
 input A,B,Bin;
 output Dout,Bout;
 assign Dout = (A^B^Bin);
 assign Bout = ((~A&C)|(B&C)|(~A&C));
endmodule
~
```

**Behavioral**

```
module Full_Sub(A,B,Bin,Dout,Bout);
 input A,B,Bin;
 output Dout,Bout;
 always @(A,B,Bin)
  begin
      assign Dout = (A^B^Bin);
      assign Bout = ((~A&C)|(B&C)|(~A&C));
  end
endmodule
~
~
```

**Simulator for Full_Substractor :**

https://circuitverse.org/simulator/edit/full_substractor-2155eed9-6dca-417d-b2e7-9bd8a5f48afc