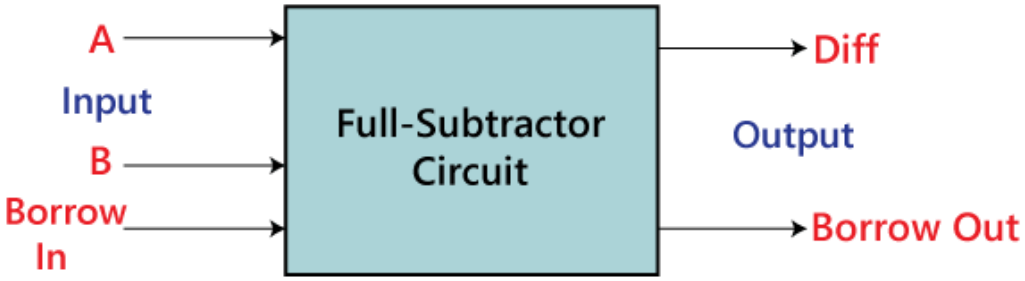
FULL SUBTRACTOR

EXPLANATION:

The Half Subtractor is used to subtract only two numbers. To overcome this problem, a full subtractor was designed. The full subtractor is used to subtract three 1-bit numbers A, B, and C, which are minuend, subtrahend, and borrow, respectively. The full subtractor has three input states and two output states i.e., diff and borrow.



DIFF: (A XOR B) XOR Bin

BORROW OUT: A'Bin + A'B + BBin

APLLICATIONS:

Some of the **applications of full-subtractor** include the following

- These are generally employed for ALU (Arithmetic logic unit) in computers to subtract as CPU & GPU for the applications of graphics to decrease the circuit difficulty.
- Subtractors are mostly used for performing arithmetical functions like subtraction, in electronic calculators as well as digital devices.
- These are also applicable for different micro controllers for arithmetic subtraction, timers, and the program counter (PC)
- Subtractors are used in processors to compute tables, addresses, etc.
- It is also useful for DSP and networking based systems.
- These are used mainly for ALU within computers for subtracting like CPU & GPU for graphics applications to reduce the complexity of the circuit.
- These are mainly used to perform arithmetical functions such as subtraction within digital devices, calculators, etc.
- These subtractors are also appropriate for various microcontrollers for timers, PC (program counter) & arithmetic subtraction
- These are employed for processors to calculate addresses, tables, etc.
- The implementation of this with logic gates like NAND & NOR can be done with any full subtractor logic circuit because both the NOR & NAND gates are called universal gates.

RTL CODE:

DATA METHOD:

```
module full_sub(d,barr,a,b,c);
  output d,barr;
  input a,b,c;
  assign d=a^b^c;
  assign barr =(c& (~(a^b)))|((~a)&b);
endmodule
```

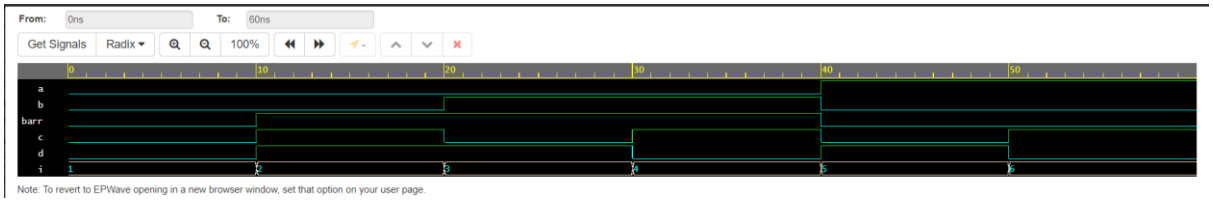
BEHAVIOURAL METHOD:

```
module full_sub(d,barr,a,b,c);
  output reg d,barr;
  input a,b,c;
  always@(*)
    begin
      d=a^b^c;
      barr =(c& (~(a^b)))|((~a)&b);
    end
endmodule
```

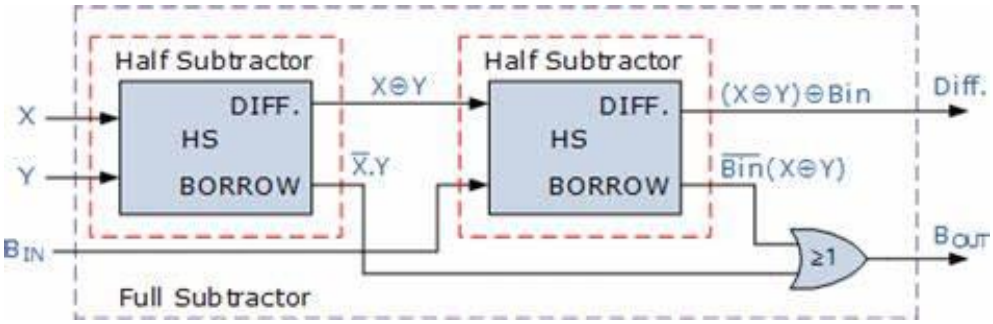
TEST BENCH:

```
module testbench;
  reg a,b,c;
  wire d,barr;
  integer i;
  full_sub a1(d,barr,a,b,c);
  initial
    begin
      $dumpfile("dump.vcd");
      $dumpvars(1);
    end
  initial
    begin
      a=0;b=0;c=0;
```

```
    end
  initial
    begin
      for(i=1;i<8;i=i+1)
        begin
          #10 {a,b,c}=i;
        end
    end
  initial
    begin
      #60 $finish();
    end
endmodule
```



FULLSUBRACTOR USING HALF SUBRACTOR



RTL CODE:

```
module full_subtractor(d,barr,a,b,c);

input a,b,c;

output d,barr;
```

```
wire p, q, r;

half_subtractor u4(p,q,a,b);

half_subtractor u5(d,p,c);

or_gate u6(barr, p,q);

endmodule
```

TEST BENCH:

```
module testbench;
  reg a,b,c;
  wire d,barr;
  integer i;
  full_sub a1(d,barr,a,b,c);
  initial
    begin
      $dumpfile("dump.vcd");
      $dumpvars(1);
    end
  initial
    begin
      a=0;b=0;c=0;
    end
  initial
    begin
      for(i=1;i<8;i=i+1)
        begin
```

```
        #10 {a,b,c}=i;
      end
    end
  initial
    begin
      #60 $finish();
    end
endmodule
```