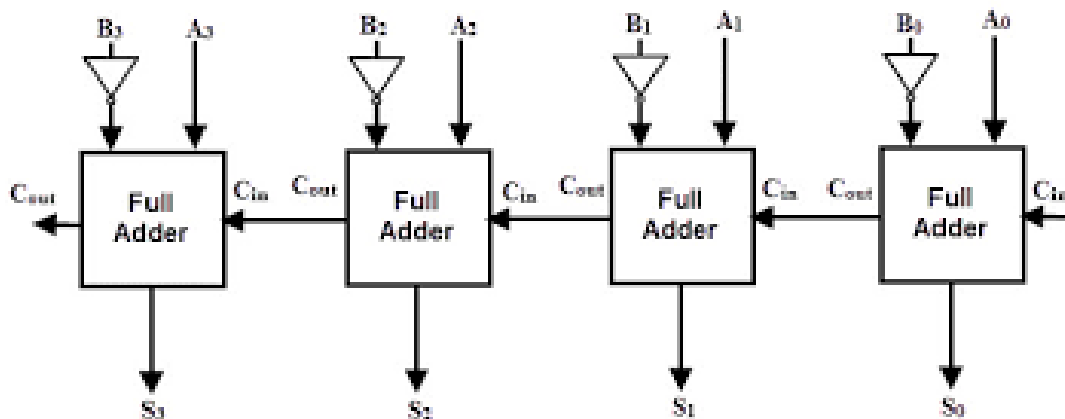


4BIT PARALLEL SUBTRACTOR

EXPLANATION:

The combinatorial circuits which are used to subtract two **binary numbers** are called Subtractors. When the binary numbers to be subtracted are of single bits, then, we can use a half subtractor to accomplish the task while if we need to subtract three binary numbers of single bits (among which two will generally be inputs while the other will be the borrow), we will have to use full subtractor. Now what if we desire to subtract two n-bit binary numbers? This is the case which demands for the use of n-bit *parallel subtractor*



Step 1 – Firstly, the 1's complement of bit B₁ obtained using an inverter and a 1 (C_{in}) are added to obtain the 2's complement of the bit B₁. Then, this 2's complemented B₁ is further added to A₁. This will produce first bit of the output difference designated by S₁, and a carry bit C₁ which is connected to the input carry of the FA₂.

Step 2 – The full adder FA₂ uses the input carry bit C₁ to add with its input bit A₂ and the 2's complement of the input bit B₂ to produce the second difference bit (S₂) and the carry bit C₂.

Step 3 – The full adder FA₃ uses the input carry bit C₂ to add with its input bit A₃ and the 2's complement of the input bit B₃ to produce the third difference bit (S₃) and the carry bit C₃.

Step 4 – Finally, the full adder FA₄ uses the carry bit C₃ to add with its input bit A₄ and the 2's complement of the input bit B₄ to produce the last difference bit (S₄) and last carry bit C₄.

RTL CODE:

```
module FA(input a, b, Cin, output sum, Cout);  
    assign sum = a ^ b ^ Cin;  
    assign Cout = (a & b) | (b & Cin) | (Cin & a);  
endmodule
```

```
module Parallel_subtractor(input [3:0] a, b, Cin, output [3:0] sum, Cout);  
    wire [2:0] w;  
    FA f1(a[0], !b[0], Cin, sum[0], w[0]);  
    FA f2(a[1], !b[1], w[0], sum[1], w[1]);  
    FA f3(a[2], !b[2], w[1], sum[2], w[2]);  
    FA f4(a[3], !b[3], w[2], sum[3], Cout);  
Endmodule
```

TESTBENCH:

```
module testbench;  
    reg [3:0] a = 4'b0000, b = 4'b0000;  
    reg Cin = 1'b0;  
    wire [3:0] sum;  
    wire Cout;
```

```
Parallel_subtractor pa2(a, b, Cin, sum, Cout);
```

```
initial
```

```
begin
```

```
#10 a = 4'b0001; b = 4'b1010; Cin = 1'b0;
```

```
#10 a = 4'b0111; b = 4'b1110; Cin = 1'b1;
```

```
end
```

```
initial
```

```
begin
```

```
$dumpfile("dump.vcd");
```

```
$dumpvars(1);
```

```
end
```

```
initial
```

```
#30 $finish();
```

```
Endmodule
```

