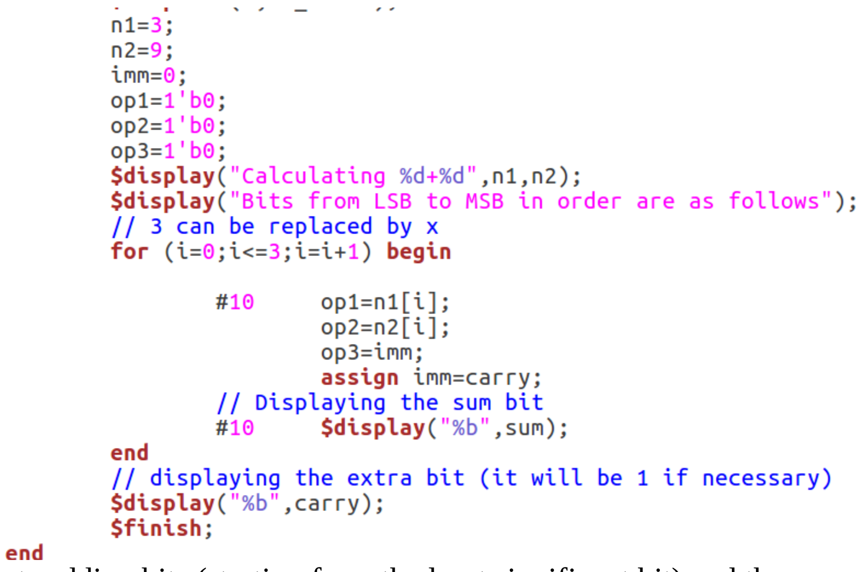
Adder and Subtractor (n-bit) using a single full adder

Adder

n bit adder can be made using a for loop



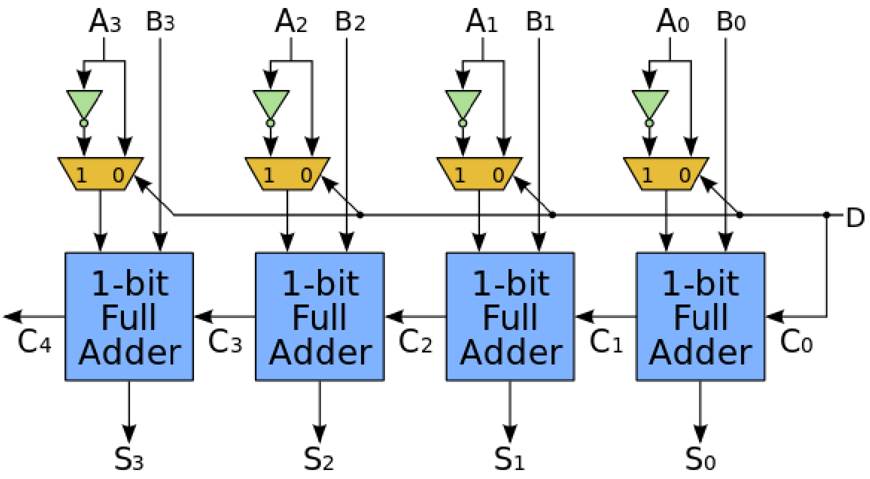
```
        n1=3;
        n2=9;
        imm=0;
        op1=1'b0;
        op2=1'b0;
        op3=1'b0;
        $display("Calculating %d+%d",n1,n2);
        $display("Bits from LSB to MSB in order are as follows");
        // 3 can be replaced by x
        for (i=0;i<=3;i=i+1) begin

                #10     op1=n1[i];
                        op2=n2[i];
                        op3=imm;
                        assign imm=carry;
                // Displaying the sum bit
                #10     $display("%b",sum);
        end
        // displaying the extra bit (it will be 1 if necessary)
        $display("%b",carry);
        $finish;
end
```

The full adder starts adding bits (starting from the least significant bit) and the carry is stored in a intermediate variable ‘imm’. Hence, using a single adder multiple bits can be added.

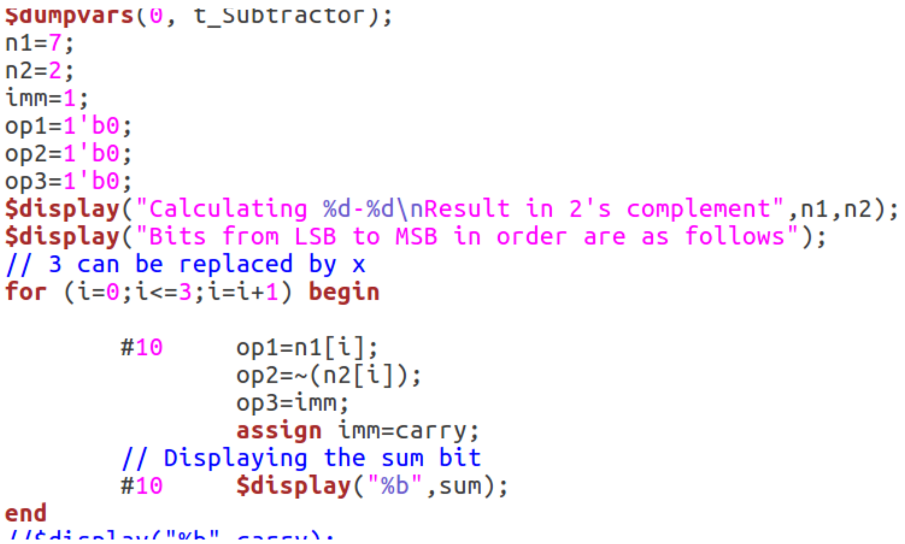
To change the number of bits the size of array and the limits in for loop must be changed in the test bench.

Subtractor



Adder can be modified into a subtractor by negating the input bits and changing the initial carry to 1.

Rest of addition can be done using a for loop.



```
$dumpvars(0, t_subtractor);
n1=7;
n2=2;
imm=1;
op1=1'b0;
op2=1'b0;
op3=1'b0;
$display("Calculating %d-%d\nResult in 2's complement",n1,n2);
$display("Bits from LSB to MSB in order are as follows");
// 3 can be replaced by x
for (i=0;i<=3;i=i+1) begin

        #10     op1=n1[i];
                op2=~(n2[i]);
                op3=imm;
                assign imm=carry;
        // Displaying the sum bit
        #10     $display("%b",sum);
end
//$display("%b",carry);
```

Just like in the case of adder.