2)

Eight-bit adder/subtractor

To add/subtract two eight-bit two's complement numbers

Verilog Implementation

It is implemented in two modules. First, the module implements a one-bit adder/subtractor with four inputs a, b, cin, and opcode, and two outputs sum and carry.
For the addition operation the input opcode will be 0 and 1 for subtraction operation.
For opcode as 1, we take 2's complement of b as
b = ~b + 8'b00000001
Which is implemented as the addition of 1 in last bit by making initial carry input 1 and taking bit wise negation of b
Otherwise, carry input initially is 0 and b is same as input
Then we take sum bitwise and also take carry out 1 when we have any two of a,b,carry_in or all are 1.
These bitwise sum and carry outputs are stored in 8-bit wire outputs using the "for loop" to call 1-bit adder/subtractor "`oneBit_Adder(input a, input b, input cin, input opcode, output sum, output cout);`" within the module

“`eightBit_adder(a,b,cin,opcode,sum,cout,overflow);`” keeps on storing the bits of sum so obtained.. The last carry out is stored in “cout” as given as final output to display.

Also we assign the overflow whenever the last and second last carry in is the same.

Truth Table of one bit adder/subtractor

Inputs				Outputs	
A	B	Carry in	Op code	sum	cout
0	0	0	0	0	0
		1		1	0
		0	1	1	0
		1		0	1
0	1	0	0	1	0
		1		0	1
		0	1	0	0
		1		1	0
1	1	0	0	0	1
		1		1	1
		0	1	1	0
		1		0	1
1	0	0	0	1	0
		1		0	1
		0	1	0	1
		1		1	1