1 BIT ALU

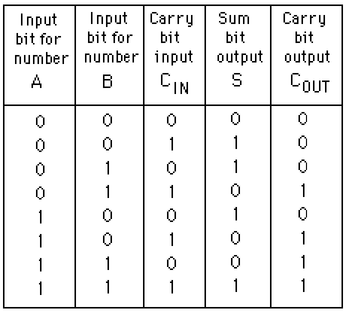
1-bit ALU. Build a 1-bit ALU that can perform the following logical and arithmetic operations: AND, OR, NAND, NOR, ADD, SUBTRACT. Include two additional flags: (a) a flag to indicate if the result of the ALU operation is zero (Z), (b) a flag to indicate if the first input is greater than the second input.

OPCODES for different functions (3 bit)

ADD:	000
SUBTRACT:	001
AND operation:	010
OR operation:	011
NAND operation:	100
NOR operation:	101

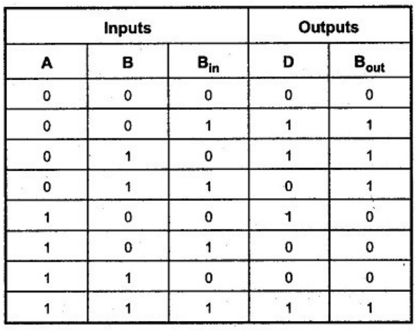
The main module accepts 4 inputs from the user. The input 1, input 2, carry_in and the opcode. The result is calculated using the following truth tables.

Adder



Input bit for number A	Input bit for number B	Carry bit input C_{IN}	Sum bit output S	Carry bit output C_{OUT}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

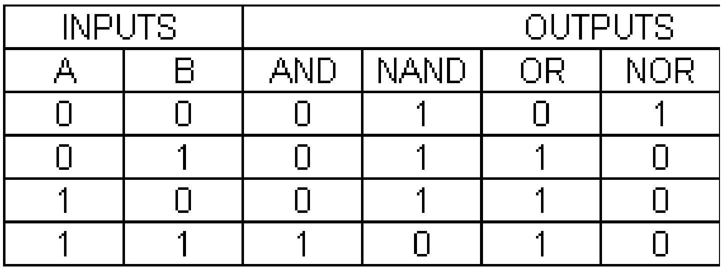
Subtractor



Inputs			Outputs	
A	B	B_{in}	D	B_{out}
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

Table 3.9 Truth table for full-subtractor

AND, OR, NAND AND NOR



INPUTS		OUTPUTS			
A	B	AND	NAND	OR	NOR
0	0	0	1	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	0	1	0

The two additional flags which check if the output is 0 or not, and to check if the 1 st input is greater or not are:

FLAG 1

flag1= not(output_bit)

FLAG 2

flag2= (input_1) and (not(input_2))