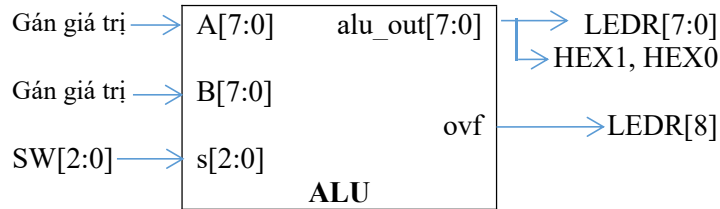


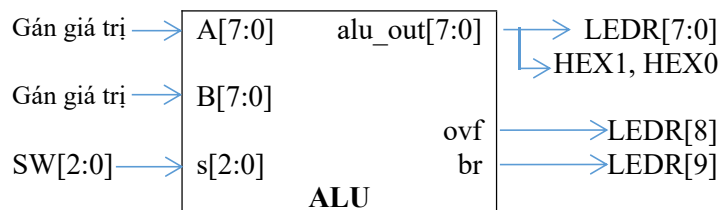
BÀI 9: THIẾT KẾ ALU (TT)

Câu 1: Viết code verilog mô tả ALU với ngõ vào A, B 8bits, thực hiện các phép toán số học và logic như sau:



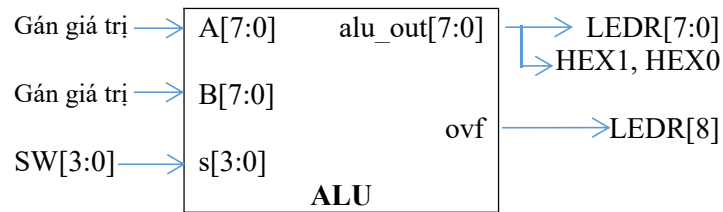
Opcode s[2:0]	ALU Operation alu_out[7:0]	Overflow ovf
000	A + B	ovf
001	A - B	0
010	A - 1	0
011	A + 1	ovf
100	A & B	0
101	A B	0
default	~A	0

Câu 2: Viết code verilog mô tả ALU với ngõ vào A, B 8bits, thực hiện các phép toán số học và logic như sau:



Opcode s[2:0]	ALU Operation alu_out[7:0]	Overflow ovf	Branch br
000	A + B	ovf	0
001	~A	0	0
010	A ^ B	0	0
011	~(A & B)	0	0
100	A >> 1	0	0
101	B << 1	0	0
110	A == B	0	1 : đúng 0 : sai
111	A > B	0	1 : đúng 0 : sai

Câu 3: Viết code verilog mô tả ALU với ngõ vào A, B 8bits, thực hiện các phép toán số học và logic như sau:



Opcode s[3:0]	ALU Operation alu_out[7:0]	Overflow ovf	Description
0000	A + B	ovf	Addition
0001	A - B	0	Subtraction
0010	A + 1	0	Increase
0011	~ B	0	Invert
0100	A << 1	0	Logical shift left
0101	A >> 1	0	Logical shift right
0110	{A[6:0],A[7]}	0	A Rotate left by 1
0111	{A[0],A[7:1]}	0	A Rotate right by 1
1000	A & B	0	Logical and
1001	A B	0	Logical or
1010	A ^ B	0	Logical xor
1011	~(A B)	0	Logical nor
1100	~(A & B)	0	Logical nand
1101	~(A ^ B)	0	Logical xnor
1110	(A>B)? alu_out=11111111: nếu đúng alu_out=00000000: nếu sai	0	Greater comparison
1111	(A==B)? alu_out=11111111: nếu đúng alu_out=00000000: nếu sai	0	Equal comparison