Cairo University

Faculty of Engineering

Computer Engineering

Computer Architecture

Lab 1

You are required to design a 8-bit ALU that accepts a 2 8-bit input values “A” and “B” and provides a 8-bit output “F” and a 1-bit output Cout. The ALU has 4-bit selection inputs “S” (S0->S3) and Cin input. The ALU will provide 16 operations according to the following table:

	S3	**S2**	**S1**	**S0**	
Part A	0	0	0	0	Next time
	0	0	0	1	
	0	0	1	0	
	0	0	1	1	
Part B	0	1	0	0	F = A xor B, Cout = 0
	0	1	0	1	F = A nand B, Cout = 0
	0	1	1	0	F = A or B, Cout = 0
	0	1	1	1	F = Not A, Cout = 0
Part C	1	0	0	0	F = Logic shift left A, Cout = shifted bit
	1	0	0	1	F = Rotate left A, Cout = rotated bit
	1	0	1	0	F = Rotate Left A with carry (cin), Cout = rotated bit
	1	0	1	1	F = 0000, Cout = 0
Part D	1	1	0	0	F = Logic shift right A, Cout = shifted bit
	1	1	0	1	F = Rotate right A, Cout = rotated bit
	1	1	1	0	F = Rotate right A with carry (cin), Cout = rotated bit

	1	1	1	1	F = Arithmetic Shift A

Requirement:

1- Implement part B
2- Compile your code without any errors or warning
3- Simulate the code with the given inputs below
4- Save the do files to submit
5- Write a testbench to test the code you implemented

S	Operation	A	B	Cin	F	Cout
0100	**XOR**	F0	B0	-	40	0
0101	**NAND**	F0	0B	-	FF	0
0110	**OR**	F0	B0	-	F0	0
0111	**NOT**	F0	-	-	0F	0

Assignment:

1- Implement part C and part D each one in a separate file

2- Compile your code without any errors or warning

3- Create a new component to integrate part B,C,D using a multiplexer

4- Write a testbench to test the code you implemented using the testcases in the table below **“in order”** where S = 01 chooses part B, S= 10 chooses part C, S = 11 chooses part D.

	Operation	A	B	Cin	F	Cout
1000	**Logic shift left**	F0	-	-	E0	1
1001	**Rotate left**	F0	-	-	E1	1
1010	**Rotate left with cin**	F0	-	0	E0	1
1011	**F=0000**	F0	-	-	00	0
1100	**Logic shift right**	F0	-	-	78	0
1101	**Rotate right**	F0	-	-	78	0
1110	**Rotate right with cin**	F0	-	0	78	0
1111	**Arithmetic shift right**	F0	-	-	F8	0
1010	**Rotate left with cin**	F0	-	1	E1	1
1110	**Rotate right with cin**	F0	-	1	F8	0