

Computer Architecture  
Lab 1

You are required to design a 16-bit ALU that accepts a 2 16-bit input values “A” and “B” and provides a 16-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 or B, Cout = 0
	0	1	0	1	F = A and B, Cout = 0
	0	1	1	0	F = A nor 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 C
- 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

Operation	A	B	Cin	F	Cout
Logic shift left	A00A	-	-	4014	1
Logic shift left	000A	-	-	0014	0
Rotate left	B00C	-	-	6019	1
Rotate left	000C	-	-	0018	0
Rotate left with cin	A00A	-	0	4014	1
Rotate left with cin	A00A	-	1	4015	1
F=0000	A00A	-	-	0000	0

### Assignment:

- 1- Implement part B 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 where S = 01 chooses part B, S= 10 chooses part C, S = 11 chooses part D.

Operation	A	B	Cin	F	Cout
OR	F000	00B0	-	F0B0	0
AND	F000	000B	-	0000	0
NOR	F000	B000	-	0FFF	0
NOT	F000	-	-	0FFF	0

Operation	A	B	Cin	F	Cout
Logic shift left	A00A	-	-	4014	1
Rotate left	B00C	-	-	6019	1
Rotate left with cin	A00A	-	0	4014	1
F=0000	A00A	-	-	0000	0

Operation	A	B	Cin	F	Cout
Logic shift right	000F	-	-	0007	1
Rotate right	0F0F	-	-	8787	1
Rotate right with cin	0F0F	-	0	0787	1
Arithmetic shift right	F000	-	-	F800	0
Logic shift left	000A	-	-	0014	0
Rotate left	000C	-	-	0018	0
Rotate left with cin	A00A	-	1	4015	1
Rotate right with cin	0F00	-	1	8780	0