Cairo University
Faculty of Engineering
Computer Engineering

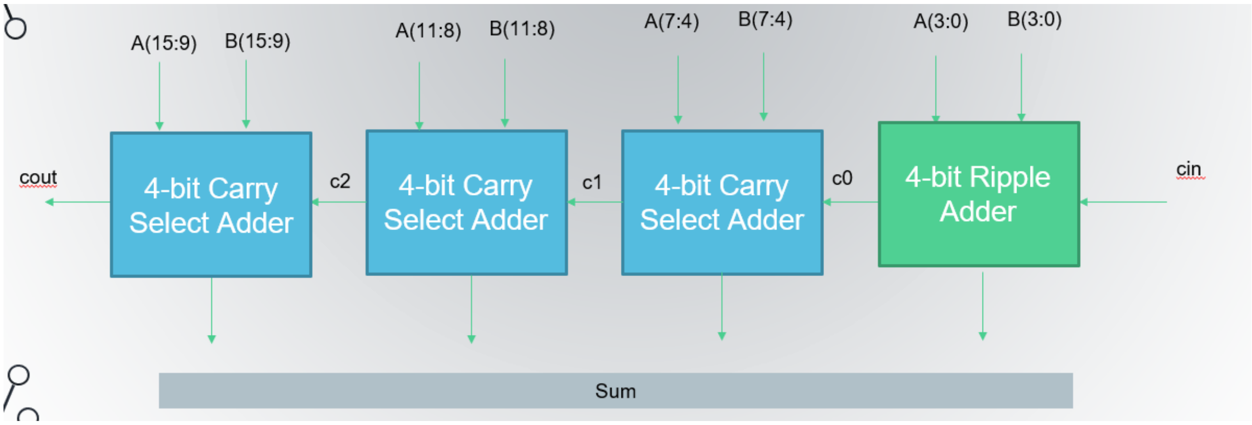
Computer Architecture
Lab 2

Continue the design of a 16-bit ALU that accepts a 2 16-bit input values “A” , “B” and a 1-bit Cin 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**	**Cin =0**	Cin =1
Part A	0	0	0	0	F=A	F=A+1
	0	0	0	1	F=A+B	F=A+B+1
	0	0	1	0	F=A-B-1	F=A-B
	0	0	1	1	F=A-1	F=B
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 16-bit-select-adder using design below



2- Implement part A using adder implemented in step. 1.

3- Compile your code without any errors or warning.

4- Integrate PartA with last time assignment (rest of the parts).

5- Add the following cases to your testbench.

Note: Borrow is the opposite of carry, you report carry.

Operation	A	B	Cin	F	Cout
F=A	0F0F	-	0	0F0F	0
F=A+B	0F0F	0001	0	0F10	0
F=A-B-1	FFFF	0001	0	FFFD	1
F=A-1	FFFF	-	0	FFFE	1
F=A+1	0F0E	-	1	0F0F	0
F=A+B+1	FFFF	0001	1	0001	1
F=A-B	0F0F	0001	1	0F0E	1
F=B	0F0F	0001	1	0001	0

Assignment:

1- Modify ALU to be generic N-bit ALU.

2- Use quartus to synthesize your implementation.

3- Take a screen shot of the RTL design of PartA.

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