BLG222E - Computer Organization
Project 1 – Part B

This project (Part B) will constitute 7/10 points of project 1. Design a 16-bit ALU. Your design should be able to perform the following operations:

- Arithmetic
    1. Add
    2. Add with carry
    3. Subtract
    4. Subtract with borrow
    5. Increment
    6. Decrement
    7. Transfer
- Logic
    1. AND
    2. NAND
    3. OR
    4. NOR
    5. XOR
    6. XNOR
    7. NOT
- Shift
    1. Logical shift left&right
    2. Circular shift&right
    3. Arithmetic shift&right

Detailed information of the given operations can be found in course book and lecture notes. You should start your implementation by building 1 bit logic unit and 1 bit arithmetic unit circuits that support given operations. Eventually, you should build the 16 bit ALU by using 1 bit ALU blocks in parallel together with some design for shift operations. As shown in Figure 2, ALU has A, B 16 bit parameter inputs, an 8-bit condition code register (CCR) and S 4 bit select input. The only output is F 16 bit result.

The bits of the condition code register (CCR) will be used in some operations, and at the end of operations some bits of the CCR should be updated. CCR is shown in Fig. 1. In CCR, bits 5-7 are reserved for future use, bit 4 is the interrupt flag, bit 3 is zero flag, bit 2 is the negative flag, bit 1 is the overflow flag, and bit 0 is the carry flag.

Table 1 explains how selection bits determine the operation, how CCR bits are used in the operations, and which CCR flags are updated at the end of the operations. The carry bit (bit 0) is used in add with carry and subtract with barrow operations. If an arithmetic operation produces an overflow, the overflow flag in CCR (bit 1) must be set. If they produce carry (or barrow) the carry bit in CCR (bit 0)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R	R	R	I	Z	N	O	C

Figure 1: Bit of the condition code register. R - reserved for future use, I - interrupt flag, Z - zero flag, N - negative flag, O - overflow flag, C -carry flag.

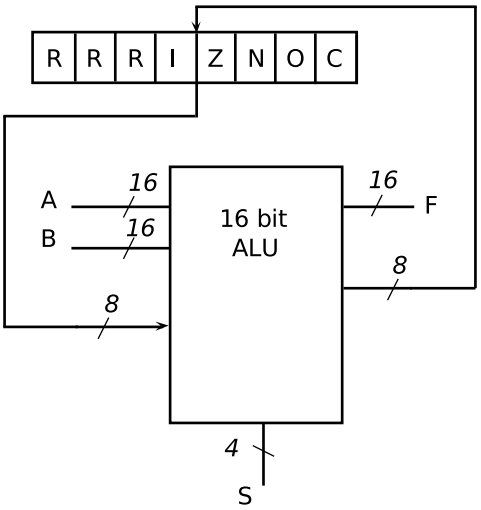

Figure 2: Inputs and outputs of the 16 bit ALU

S_3	S_2	S_1	S_0	C (bit 0) in CCR	Operation	Function	Flag updates			
							Z	N	O	C
1	0	0	0	0	$F \leftarrow A$	Transfer A	√	√	√	√
1	0	0	0	1	$F \leftarrow A + 1$	Increment A	√	√	√	√
1	0	0	1	0	$F \leftarrow A + B$	Addition	√	√	√	√
1	0	0	1	1	$F \leftarrow A + B + 1$	Add with carry	√	√	√	√
1	X	1	0	0	$F \leftarrow A + \bar{B}$	Subtract with borrow	√	√	√	√
1	0	1	0	1	$F \leftarrow A + \bar{B} + 1$	Subtraction	√	√	√	√
1	0	1	1	0	$F \leftarrow A - 1$	Decrement A	√	√	√	√
1	0	1	1	1	$F \leftarrow A$	Transfer A	√	√	√	√
0	1	0	0	0	$F \leftarrow A \wedge B$	AND	√	√	–	–
0	1	0	0	1	$F \leftarrow \overline{A \wedge B}$	NAND	√	√	–	–
0	1	1	0	0	$F \leftarrow A \vee B$	OR	√	√	–	–
0	1	1	0	1	$F \leftarrow A \oplus B$	XOR	√	√	–	–
0	1	0	1	1	$F \leftarrow \overline{A \vee B}$	NOR	√	√	–	–
0	1	0	1	1	$F \leftarrow \overline{A \oplus B}$	XNOR	√	√	–	–
0	1	1	1	X	$F \leftarrow \bar{A}$	Complement A	√	√	–	–
0	0	0	0	0	$F \leftarrow shrA$	Logical shift right A into F	√	√	–	–
0	0	0	0	1	$F \leftarrow ashrA$	Arithmetic shift right A into F	√	√	–	–
0	0	0	1	0	$F \leftarrow cshrA$	Circular shift right A into F	√	√	–	–
0	0	0	1	1	$F \leftarrow shlA$	Logical shift left A into F	√	√	–	–
0	0	1	X	0	$F \leftarrow ashlA$	Arithmetic shift left A into F	√	√	√	–
0	0	1	0	1	$F \leftarrow cshlA$	Circular shift left A into F	√	√	–	–

Table 1: Operations of the ALU from the lecture notes. × means don't care, √ means the flag is updated, and – means the flag is not updated (effected) by the corresponding operation.

must be updated. If the result of the operations produces a zero value or a negative value, the zero flag or the negative flag should be updated accordingly.

Implement your design in **logisim** software, upload a single compressed (zip or rar) file to ninova before the deadline. Only one student from each group should submit the project file. This compressed file should contain your design files (.circ) and a report that contains:

- the number&names of the students in the group
- list of control inputs and corresponding functions of the ALU

Group work is expected for this project. Form groups of 4 students, and design together. You will be asked to make a 10-minute demonstration of your design with a few test cases.

Make sure to connect pins (under Wiring group of logisim) to the inputs and control inputs of your design, so that different inputs and functions can be tested. Similarly connect your inputs and outputs to a "Hex Digit Display" in logisim (under Input/output group of logisim) so that the test outputs can be observed and use proper labelling to improve the clarity of your circuits.