

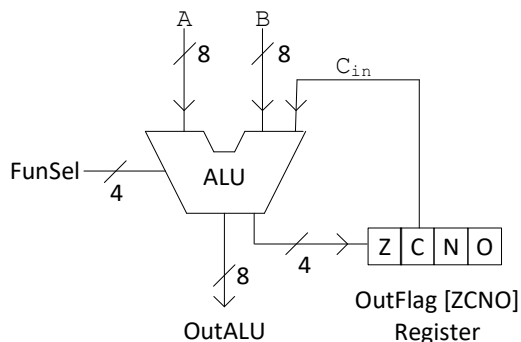
## BLG222E Computer Organization

### Project 2

Due Date: 03.04.2019, 23:00

**(Part-1)** Design an Arithmetic Logic Unit (ALU) that has two 8-bit inputs and an 8-bit output. The ALU is shown in the left side of Figure 1. The ALU functions and the flags that will be updated (i.e., - means that the flag will not be affected and  $\checkmark$  means that the flag changes based on the OutALU) are given in the right side of Figure 1:

- **FunSel** selects the function of the ALU.
- **OutALU** shows the result of the operation that is selected by **FunSel** and applied on A and/or B inputs.
- **Z (zero)** bit is set if **OutALU** is zero (e.g., **NOT B** is zero).
- **C (carry)** bit is set if **OutALU** sets the carry (e.g., **LSL A** produces carry).
- **N (negative)** bit is set if the ALU operation generates a negative result (e.g., **A-B** results in a negative number).
- **O (overflow)** bit is set if an overflow occurs (e.g., **A+B** results in an overflow).



| FunSel | OutALU        | Z            | C            | N            | O            |
|--------|---------------|--------------|--------------|--------------|--------------|
| 0000   | A AND B       | $\checkmark$ | -            | $\checkmark$ | -            |
| 0001   | A OR B        | $\checkmark$ | -            | $\checkmark$ | -            |
| 0010   | NOT A         | $\checkmark$ | -            | $\checkmark$ | -            |
| 0011   | A XOR B       | $\checkmark$ | -            | $\checkmark$ | -            |
| 0100   | A             | $\checkmark$ | -            | $\checkmark$ | -            |
| 0101   | A + B         | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ |
| 0110   | A + B + Carry | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ |
| 0111   | A - B         | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ |
| 1000   | B             | $\checkmark$ | -            | $\checkmark$ | -            |
| 1001   | NOT B         | $\checkmark$ | -            | $\checkmark$ | -            |
| 1010   | LSL A         | $\checkmark$ | $\checkmark$ | $\checkmark$ | -            |
| 1011   | LSR A         | $\checkmark$ | $\checkmark$ | $\checkmark$ | -            |
| 1100   | ASL A         | $\checkmark$ | -            | $\checkmark$ | $\checkmark$ |
| 1101   | ASR A         | $\checkmark$ | -            | $\checkmark$ | -            |
| 1110   | CSL A         | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ |
| 1111   | CSR A         | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ |

Figure 1: The ALU (Left) and its characteristic table (Right)

(Circular | Arithmetic | Logical) Shift (Left | Right) operations are depicted in Figure 2, Figure 3, and Figure 4.

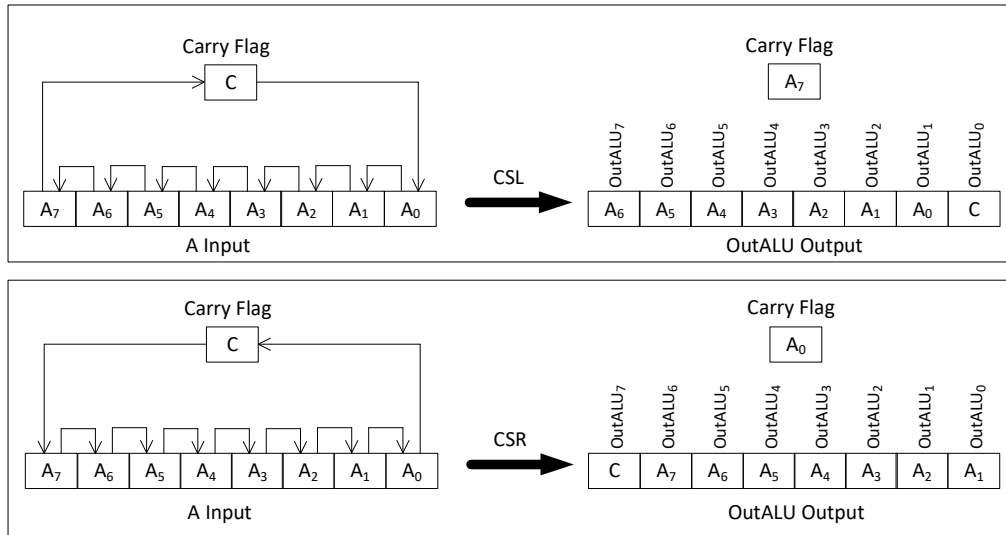


Figure 2: Circular Shift Operations

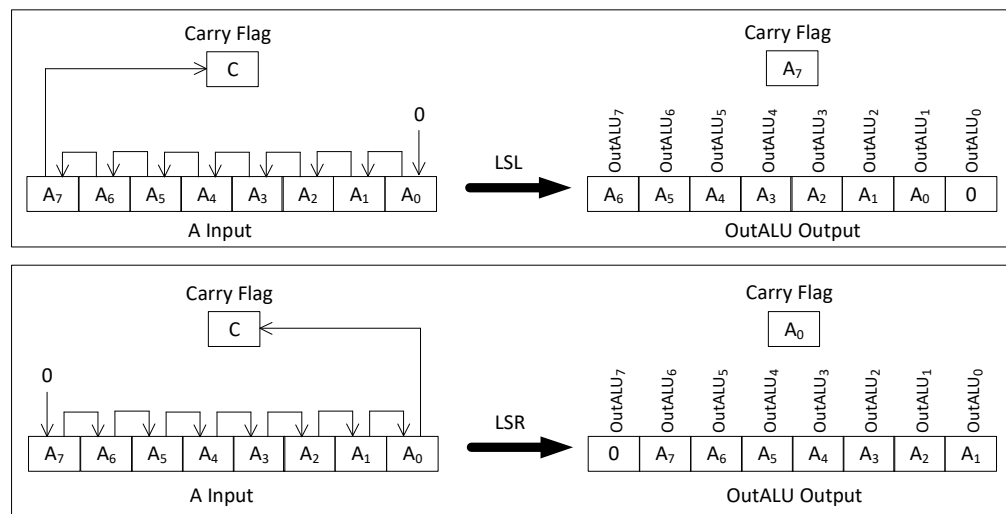


Figure 3: Logical Shift Operations

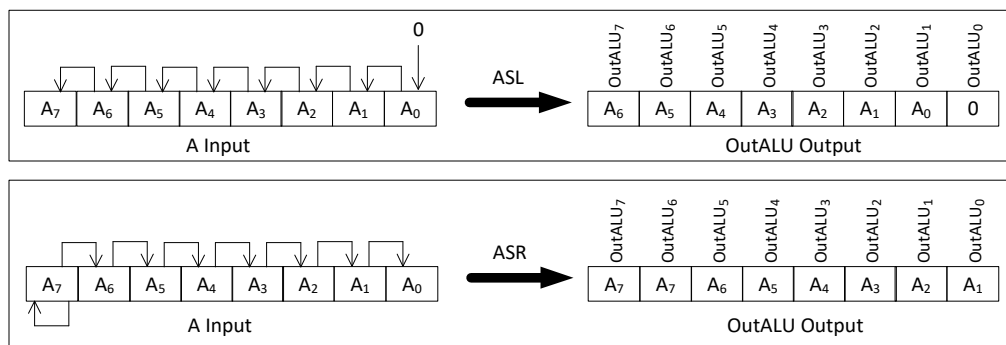


Figure 4: Arithmetic Shift Operations

**(Part-2)** Implement the organization in Figure 5.

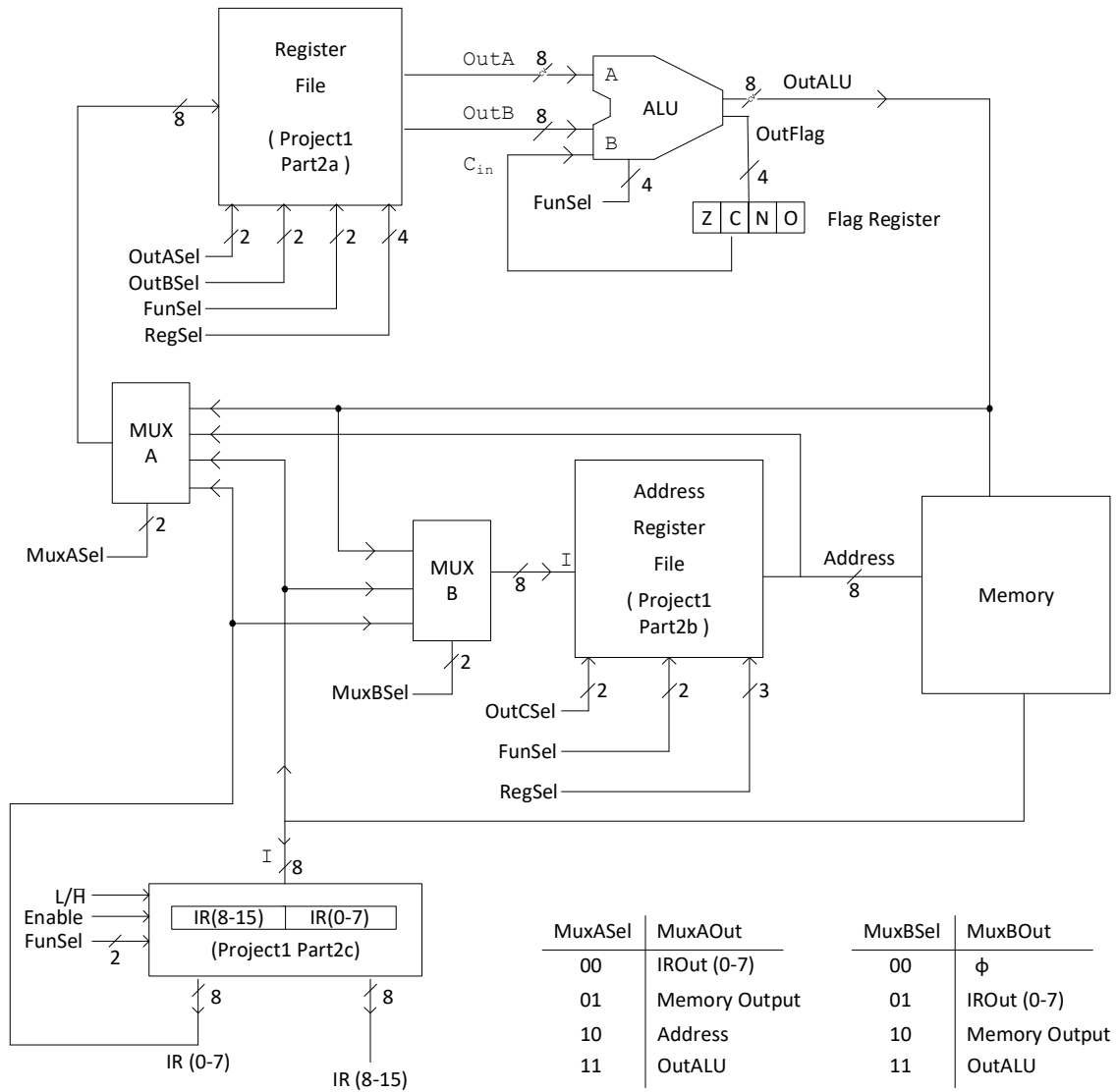


Figure 5: ALU System