# BLG222E Computer Organization

Project 2

Due Date: 12.04.2017, 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 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.
- **C (carry)** bit is set if **OutALU** is zero.
- **N (negative)** bit is set if the ALU operation generates a carry.
- **O (overflow)** bit is set if an overflow occurs**.**
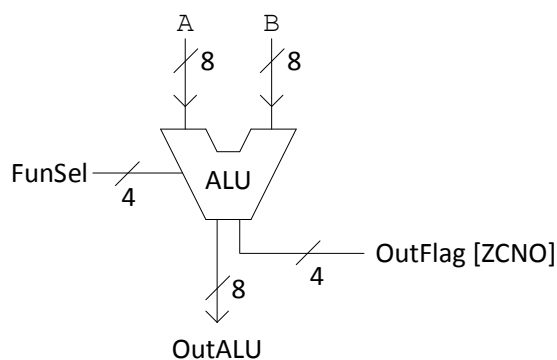
FunSel	OutALU	Z	C	N	O
0000	A	√	−	√	−
0001	A + B	√	√	√	√
0010	A + B + Carry	√	√	√	√
0011	A - B	√	√	√	√
0100	A AND B	√	−	√	−
0101	A OR B	√	−	√	−
0110	NOT A	√	−	√	−
0111	A XOR B	√	−	√	−
1000	LSL A	√	√	√	−
1001	LSR A	√	−	√	−
1010	ASL A	√	−	√	√
1011	ASR A	√	−	−	√
1100	CSL A	√	√	√	√
1101	CSR A	√	√	√	√
1110	B	√	−	√	−
1111	NOT B	√	−	√	−

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

(Circular | **A**rithmetic | **L**ogical) **S**hift (**L**eft | **R**ight) operations are depicted in Figure 2, *Figure 3, and Figure 4*.

Carry Flag

C

A₇ | A₆ | A₅ | A₄ | A₃ | A₂ | A₁ | A₀
A Input

CSL

Carry Flag

A₇

OutALU₇ | OutALU₆ | OutALU₅ | OutALU₄ | OutALU₃ | OutALU₂ | OutALU₁ | OutALU₀

A₆ | A₅ | A₄ | A₃ | A₂ | A₁ | A₀ | C
OutALU Output

Carry Flag

C

A₇ | A₆ | A₅ | A₄ | A₃ | A₂ | A₁ | A₀
A Input

CSR

Carry Flag

A₀

OutALU₇ | OutALU₆ | OutALU₅ | OutALU₄ | OutALU₃ | OutALU₂ | OutALU₁ | OutALU₀

C | A₇ | A₆ | A₅ | A₄ | A₃ | A₂ | A₁
OutALU Output
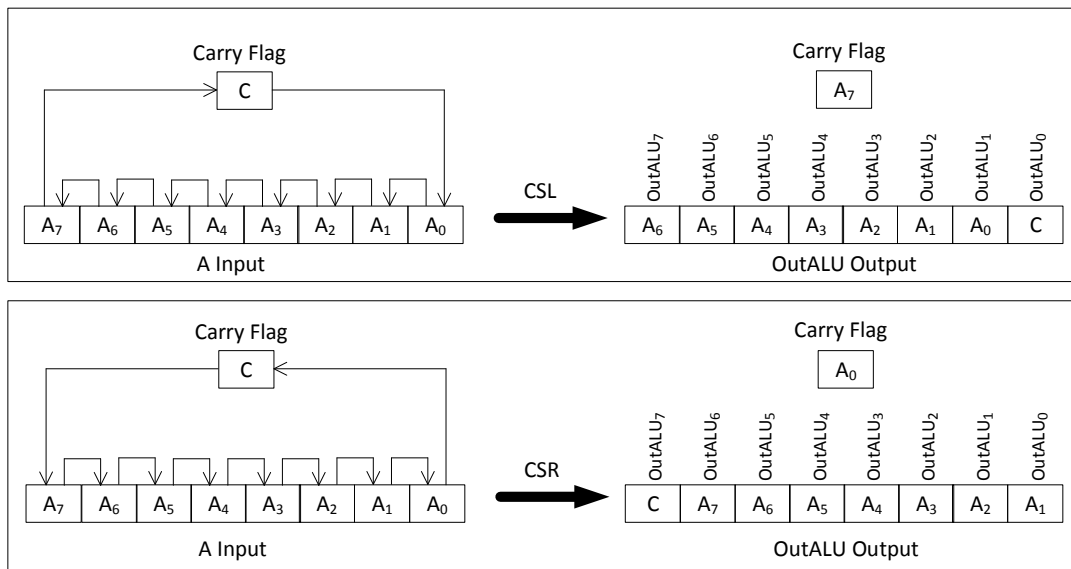
Figure 2: Circular Shift Operations

Carry Flag

C

0

A₇ | A₆ | A₅ | A₄ | A₃ | A₂ | A₁ | A₀
A Input

LSL

Carry Flag

A₇

OutALU₇ | OutALU₆ | OutALU₅ | OutALU₄ | OutALU₃ | OutALU₂ | OutALU₁ | OutALU₀

A₆ | A₅ | A₄ | A₃ | A₂ | A₁ | A₀ | 0
OutALU Output

Carry Flag

C

0

A₇ | A₆ | A₅ | A₄ | A₃ | A₂ | A₁ | A₀
A Input

LSR

Carry Flag

A₀

OutALU₇ | OutALU₆ | OutALU₅ | OutALU₄ | OutALU₃ | OutALU₂ | OutALU₁ | OutALU₀

0 | A₇ | A₆ | A₅ | A₄ | A₃ | A₂ | A₁
OutALU Output
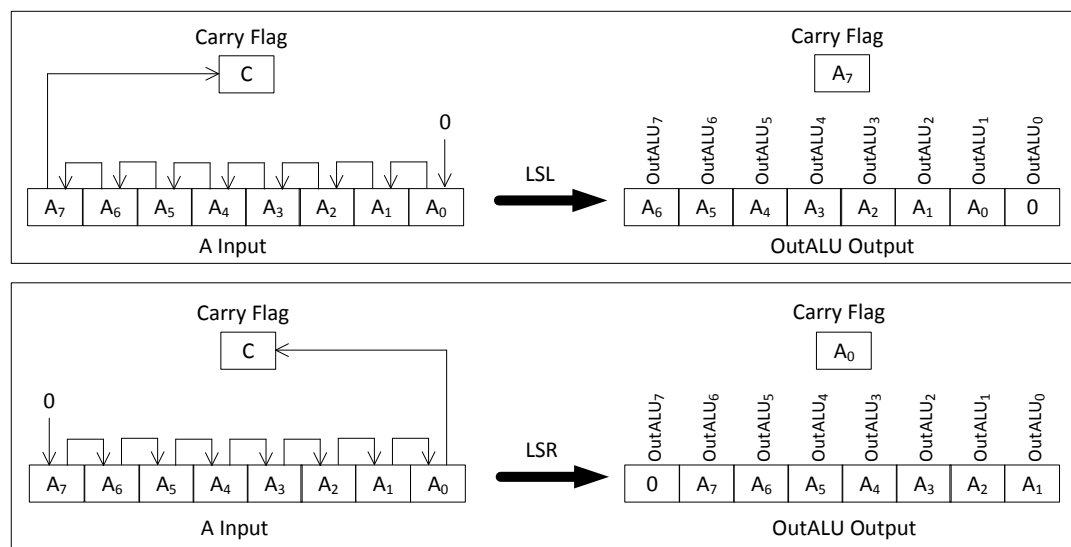
Figure 3: Logical Shift Operations

0

A₇ | A₆ | A₅ | A₄ | A₃ | A₂ | A₁ | A₀
A Input

ASL

OutALU₇ | OutALU₆ | OutALU₅ | OutALU₄ | OutALU₃ | OutALU₂ | OutALU₁ | OutALU₀

A₆ | A₅ | A₄ | A₃ | A₂ | A₁ | A₀ | 0
OutALU Output

A₇ | A₆ | A₅ | A₄ | A₃ | A₂ | A₁ | A₀
A Input

ASR

OutALU₇ | OutALU₆ | OutALU₅ | OutALU₄ | OutALU₃ | OutALU₂ | OutALU₁ | OutALU₀
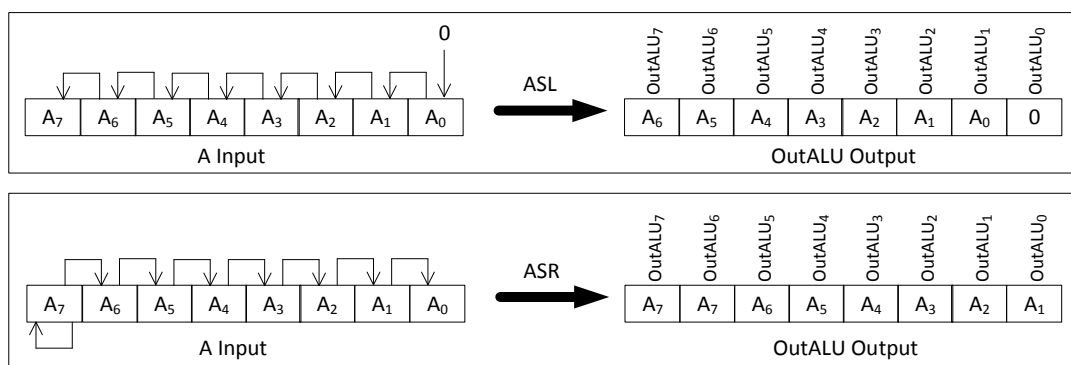
A₇ | A₇ | A₆ | A₅ | A₄ | A₃ | A₂ | A₁
OutALU Output

Figure 4: Arithmetic Shift Operations

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

MuxASel	MuxAOut		MuxBSel	MuxBOut
00	OutALU		00	OutALU
01	Address		01	ϕ
10	Memory Output		10	Memory Output
11	IROut (0-7)		11	IROut (0-7)

Figure 5: ALU System