CSE - 306

Computer Architecture Sessional

# 4-bit ALU Design

Group - 02

Group members -

1905033
1905049
1905053
1905055
1905056

## Introduction:

Arithmetic Logic Unit, which is known as 'ALU', is the part of a computer that performs all arithmetic computations, such as addition, multiplication and all comparison operations. The arithmetic logic unit (ALU) represents the fundamental building block of the CPU (Central Processing Unit) of a computer. Examples of logic operations are NOT, AND, OR which are done by ALU. The arithmetic logic unit has two 4-bit input and three selection variables. The selection bits enable different types of operation. $2^3 = 8$ operations are possible in our experiment. using the 4-bit inputs. In order to handle 4-bit inpats, 4-bit full adder (74LS83) is used.

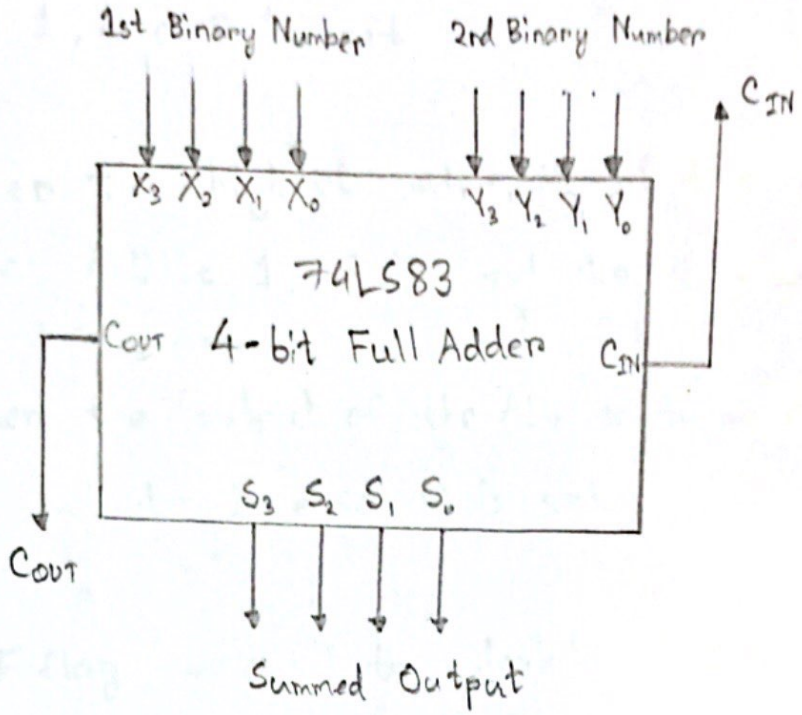


Fig: A 4-bit Full Adder

Different types of status arise while implementing arithmetic logic unit which are carry (C), sign (S), overflow (⁂V) and zero (Z). Different values appear in them in different operations of arithmetic logic unit.

CF: when the output carry of the 'ALU' becomes 1, CF is set to 1, else 0 is set.

OF: If the XOR of carries $C_4$ and $C_3$ is, V is set to 1, else 0 is set.

SF: When the highest order bit of the output of the ALU is 1, S is set to 1 else 0 is set.

ZF: When the output of the ALU contains all 0's, Z is set to 1 else 0 is set.

For OF flag, we need to calculate $C_3 \oplus C_4$ but we can't get directly $C_3$ from adder. So the value of OF is calculated in the following way,

$V = C_3 \oplus C_4$

$\Rightarrow V = (X_3 \oplus Y_3 \oplus S_3) \oplus C_4$

$S_3 = X_3 \oplus Y_3 \oplus C_3$

Here,
X and Y are the 4-bits inputs provided to the adder and $S_3$ is the MSB of summed output.

## Problem Specification:

Design a 4-bit ALU with three selection bits $cs_2$(s) $cs_1$(s) $cs_0$(s) for performing the following operations,

| $C_{IN}$ | | | Operation | Output |
|---|---|---|---|---|
| $S_2$ | $S_1$ | $S_0$ | | |
| 0 | 0 | 0 | Transfer A | F = A |
| 0 | 0 | 1 | AND | F = A & B |
| 0 | 1 | X | ADD | F = A + B |
| 1 | 0 | 0 | ADD with carry | F = A + B + 1 |
| 1 | 0 | 1 | Complement A | F = A' |
| 1 | 1 | X | Decrement A | F = A - 1 |

## Truth Table:

| $cs_2$ | $cs_1$ | $cs_0$ | $X_i$ | $Y_i$ | $C_{in}$ | Output (F) |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | $A_i$ | 0 | 0 | F = A |
| 0 | 0 | 1 | $A_i \oplus B_i$ | 0 | 0 | F = A ⊕ B |
| 0 | 1 | X | $A_i$ | $B_i$ | 0 | F = A + B |
| 1 | 0 | 0 | $A_i$ | $B_i$ | 1 | F = A + B + 1 |
| 1 | 0 | 1 | $\overline{A_i}$ | 0 | 0 | F = A' |
| 1 | 1 | X | $A_i$ | 1 | 0 | F = A - 1 |

All 1's

## Equations:

$$X_i = \bar{S}_0 A_i + S_1 A_i + \bar{S}_2 A_i B_i + S_2 \bar{S}_1 S_0 \bar{A}_i$$

$$Y_i = S_1 B_i + S_2 S_1 + S_2 \bar{S}_0 B_i$$

$$C_{in} = S_2 \bar{S}_1 \bar{S}_0$$

| $A_i$ | $\bar{A}_i$ | $A_i \& B_i$ | $t_1$ | $t_0$ |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |

Truth table for 4x1 MUX selection bit of $X_i$

$t_0 = \bar{A}_i$ (Decoder output $Y_5$)

$t_1 = A_i \& B_i$ (Decoder output $Y_1$)

| $B_i$ | 0 | 1 | $t_1$ | $t_0$ |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |

Truth table for 4x1 MUX selection bit of $Y_i$

$t_1$ = OR of decoder output lines that give 1

$t_0$ = OR of decoder output lines that give 0

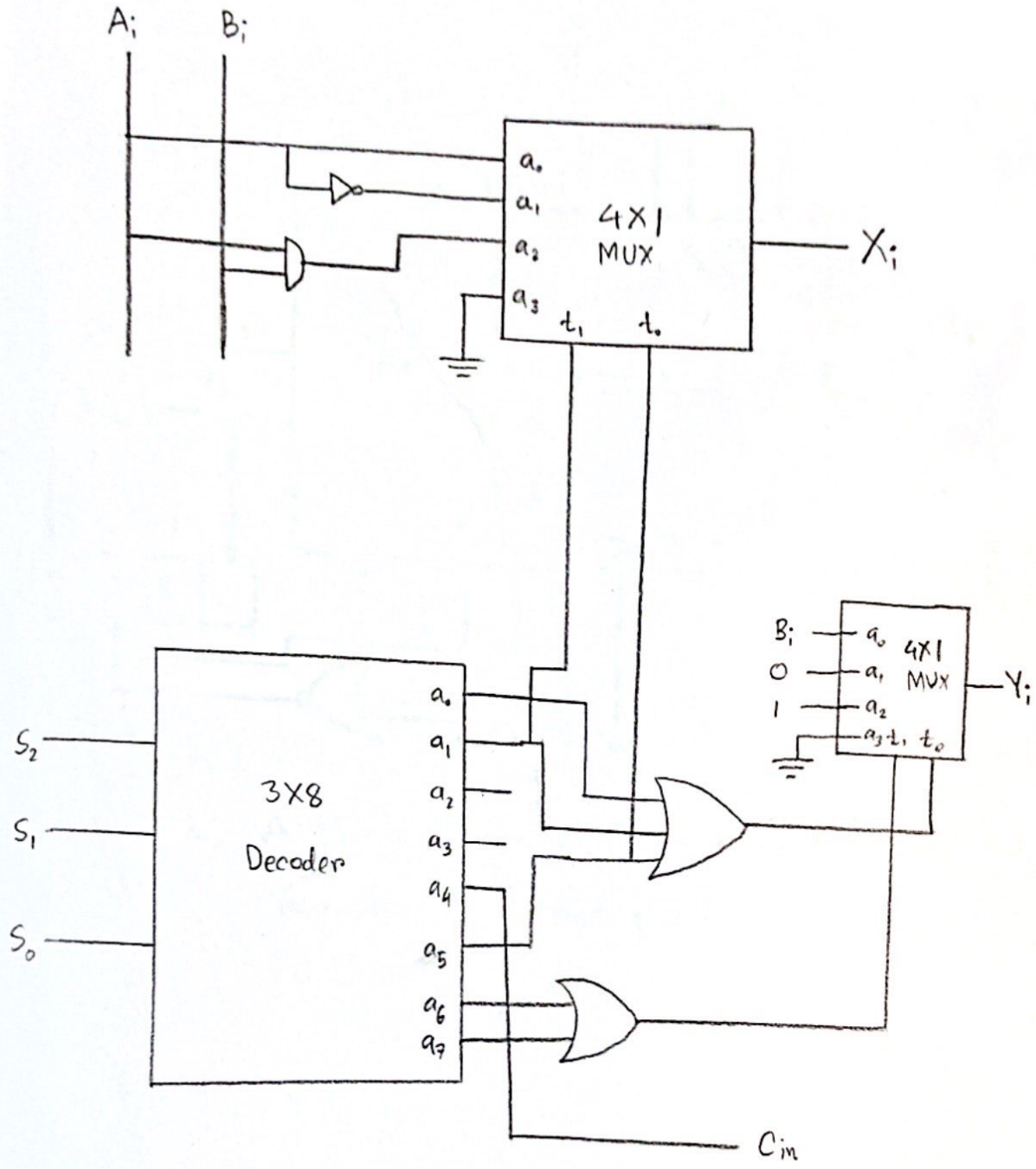


Fig 3 : A block diagram to produce inputs of 4 bit adder.

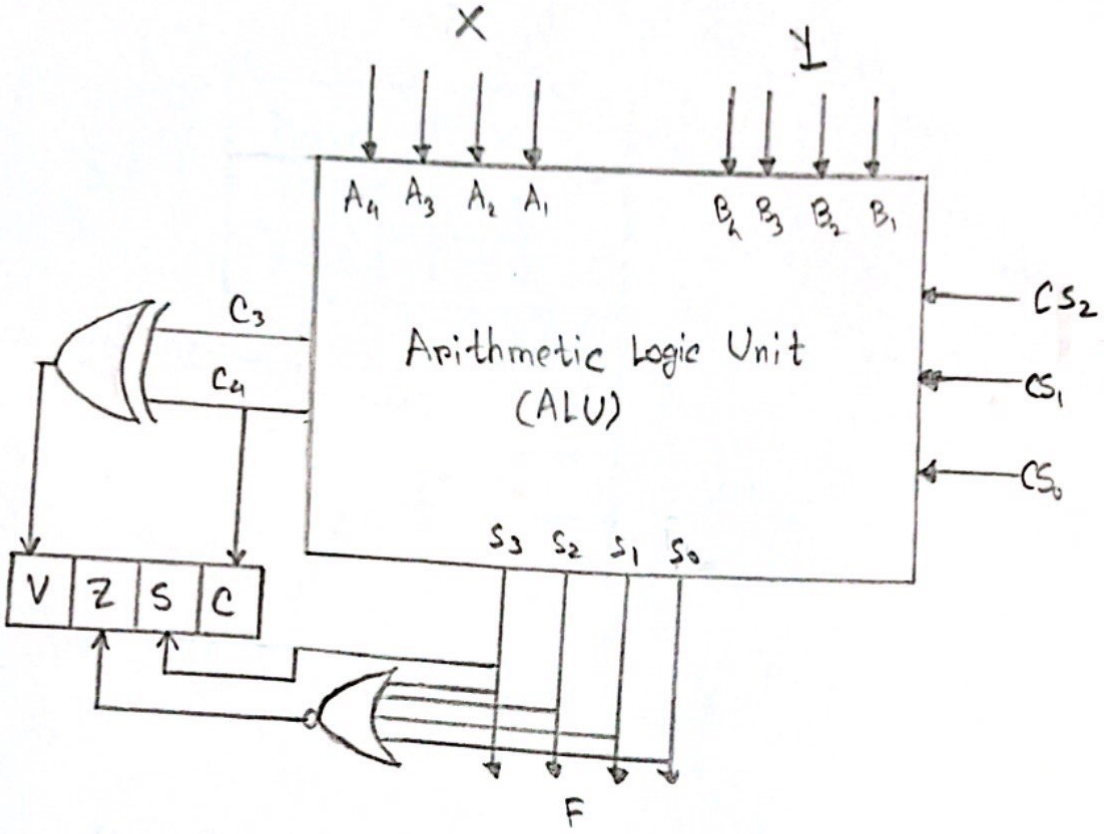


Fig 2: A block diagram of an ALU with status registers. (External View)

Required ICs :

| Name | Count |
|---|---|
| 74238 | 1 |
| 74153 | 4 |
| 7404 | 1 |
| 7408 | 1 |
| 7432 | 2 |
| 7483 | 1 |
| 7486 | 1 |

Simulator : Logisim