CSE306

Computer Architecture Sessional

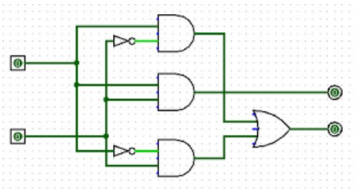
Assignment-1: 4-bit ALU Simulation

Section-A2

GROUP-03

Group members student_ID:

1. 1605051
2. 1705040
3. 1705042
4. 1705048
5. 1705054



Submission date: March 28,2021

Assignment-1:4-bit ALU simulation

Introduction

An Arithmetic Logic Unit (ALU) is a multioperation, combinational-logic digital function. It can perform a set of basic arithmetic and logical operation. The ALU has a number of selection lines to select a particular operation. In our assignment , we have cs1,cs0 as selection bits and cs2 as input carry. The four data inputs from A are combined with the four inputs from B to generate an operation at the F output. We will also implement status register along with the ALU. The flags in the status registers are modified as effects of arithmetic and bit modulation operation. The four status bits C,S,Z and V are set or cleared as follows:

1. Carry(C): C is set to 1 if output carry is 1 and cleared if output carry is 0.
2. Sign(S): S is set if the highest order bit of the result in the output of ALU (the sign bit) is 1. It is cleared otherwise.
3. Zero(Z): Z is set to 1 if the output of ALU contains all 0's. Z=0 when result is non-zero.
4. Overflow(V): If the input carry and the output carry of last (msb) operation is not same then V is set to 1 . Otherwise V=0.

Functional Design specification

cs2(cin)	cs1	cs0	function
0	0	0	subtract with borrow
0	0	1	transfer A
1	0	0	subtract
1	0	1	Increment A
X	1	0	AND
X	1	1	OR

Mode Selection

cs1	operation
0	Arithmetic
1	Logical

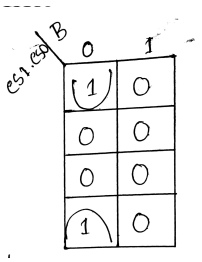
Truth Table

cs2(cin)	cs1	cs0	X	Y	$F = X \oplus Y$	required
0	0	0	A	$\neg B$	$F = A + \neg B$	$F = A + \neg B$
0	0	1	A	0	$F = A$	$F = A$
1	0	0	A	$\neg B$	$F = A + \neg B + 1$	$F = A + \neg B + 1$
1	0	1	A	0	$F = A + 1$	$F = A + 1$
X	1	0	A	$\neg B$	$F = A \oplus \neg B$	$F = AB$
X	1	1	A	0	$F = A$	$F = A \vee B$

Truth table for modified B, Y

cs1	cs0	B	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

k map for Y_i



boolean function for Y_i : $Y_i = \neg cs0 \cdot \neg B_i$

Truth table for input carry ,Z_i

Input carry to each adder depends on the operation mode. For logical operation input carry= 0 . For arithmetic operation ,input carry is required.

cs1	c_i
0	c_i
1	0

Truth table

cs1	c_i	Z_i
0	0	0
0	1	1
1	0	0
1	1	0

boolean function for Z_i : $Z_i = \neg cs1 \cdot c_i$

The values of Y input to the full-adder circuits depends on the selection bit cs0.

cs0	Y_i
0	$\neg B$
1	0

Logical operations (when $cs1 = 1$)

cs1	cs0	X	Y	F	required
1	0	A	$\neg B$	$A \oplus \neg B$	AB
1	1	A	0	A	$A \vee B$

For the 2^{nd} row , we need ,$F = A \vee B$
so ,we need , $X = A + B$, since $Y = 0$
so $X_i = A_i + cs1 \cdot cs0 \cdot B_i$
For 1^{st} row ,$F = A \oplus \neg B$
let, $A = A + K$
then ,$F = (A + K) \oplus \neg B = AB + KB + \neg A \neg K \neg B$
when $K = \neg B$ then $F = AB$
hence , the boolean function for input X_i,

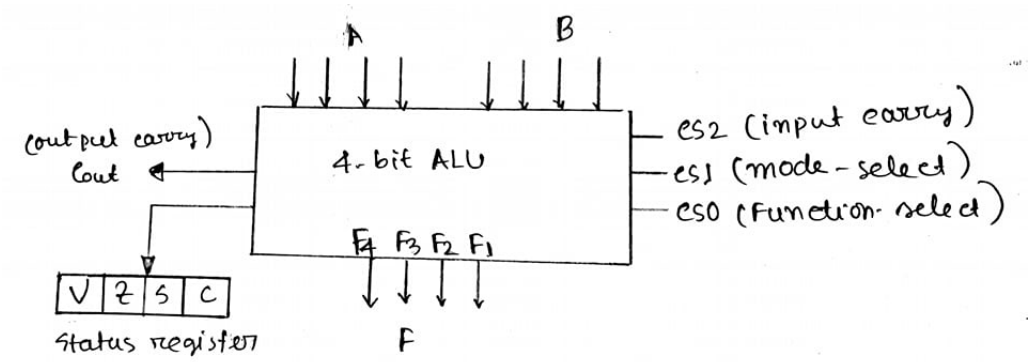
$$X_i = A_i + cs1 \cdot cs0 \cdot B_i + cs1 \cdot \neg cs0 \cdot \neg B_i$$

Boolean functions for inputs

The Inputs to each full-adder circuit are specified by the following boolean functions:

$$X_i = A_i + cs1 \cdot cs0 \cdot B_i + cs1 \cdot \neg cs0 \cdot \neg B_i$$
$$Y_i = \neg cs0 \cdot \neg B_i$$
$$Z_i = \neg cs1 \cdot c_i$$

Block diagram



Simulator used with version

simulator : Logisim-generic-2.7.1

IC used with number

IC number	IC type	count
7408	quad 2 input AND gate	6
7432	quad 2 input OR gate	2
7404	Hex NOT gate	1
7486	quad 2 input XOR gate	1
7425	dual 4 input NOR gate	1
7483	4 bit full-adder	4

Complete circuit diagram

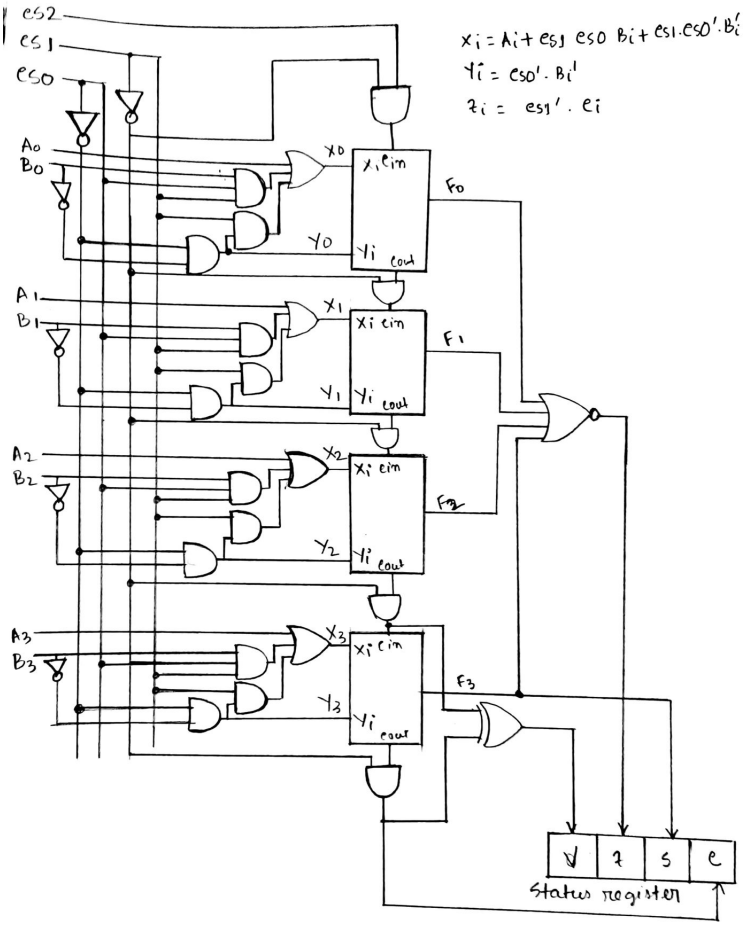


fig : complete circuit diagram

Discussion

In this assinment, we used basic logic gates (AND, OR , NOT) and universal NOR gate and XOR gate to simulate our specified arithmetic logic unit (ALU). We has cs2, cs1, cs0 as input selection bits . where

cs2 :=input carry bit
cs1:= mode selector bit
cs0 := function selection bit (as specified in assignment)

We also implemented status register to check status bits after an ALU operation. While simulating the circuit diagram in logisim simulator, we set the input number to any gate 2 or 3 for convenience . But we counted IC number considering 2 input gate(except NOT gate). We needed 1 bit full-adder for add operation , since input carry to each adder depended on the operation mode(arithmetic or logical). So , we needed to set input carry(Z_i) for every adder . That's why we counted 4 IC-7483(4 bit full-adder).