Project Report (Part-01)

CSE332

SECTION: 9

Submitted to:

Syed Mahamud Husain (SMH2)

Department of Electrical & Computer Engineering

Submitted by:

Muhammad Zawad Mahmud

ID: 1931401642

Design Process (QM method):

Truth Table:

$Alu\,op_0$	$Alu\,op_1$	$Alu\,op_2$	F_1	F_0	X	Y	Z
0	0	0	0	0	0	1	0
0	1	0	x	0	1	1	0
0	1	1	x	0	0	1	1
0	1	1	x	1	1	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1
0	0	0	1	1	1	1	1

For this problem I am considering all the don't cares as 1's as I am going to use SOP.

with the help of SOP,

$X = \overline{Alu\,op_0}\; Alu\,op_1\; \overline{Alu\,op_2}\; F_1 \bar{F}_0 + \overline{Alu\,op_0}\; Alu\,op_1\; Alu\,op_2\; F_1 F_0 + \overline{Alu\,op_0}\; \overline{Alu\,op_1}\; \overline{Alu\,op_2}\; F_1 F_0$

$= \Sigma m(3, 10, 15)$

$Y = \overline{Alu\,op_0}\; \overline{Alu\,op_1}\; \overline{Alu\,op_2}\; \bar{F}_1 \bar{F}_0 + \overline{Alu\,op_0}\; Alu\,op_1\; \overline{Alu\,op_2}\; F_1 \bar{F}_0 + \overline{Alu\,op_0}\; Alu\,op_1\; Alu\,op_2\; F_1 \bar{F}_0 + \overline{Alu\,op_0}\; \overline{Alu\,op_1}\; \overline{Alu\,op_2}\; F_1 F_0$

$= \Sigma m(0, 3, 10, 14)$

$Z = \overline{AluOp_0}\ AluOp_1\ AluOp_2\ F_1\ \bar{F}_0 + \overline{AluOp_0}\ \overline{AluOp_1}\ \overline{AluOp_2}\ F_1\ \bar{F}_0 + \overline{AluOp_0}\ \overline{AluOp_1}\ \overline{AluOp_2}\ F_1\ F_0$

$= \Sigma m\ (2, 3, 14)$

am Method for simplifying X,

Step - 01:

Group	Term	$AluOp_0$	$AluOp_1$	$AluOp_2$	F_1	F_0
group-0	3	0	0	0	1	1
(two number of 1's)	10 ✓	0	1	0	1	0
group-1	15	0	1	1	1	1

(four number of 1's)

Step - 02: No matched pairs found.

Step - 03: P.I Table

P.I	Minterm 3	10	15
$\overline{AluOp_0}\ \overline{AluOp_1}\ \overline{AluOp_2}\ F_1\ F_0$	⊗		
$\overline{AluOp_0}\ AluOp_1\ \overline{AluOp_2}\ F_1\ \bar{F}_0$		⊗	
$\overline{AluOp_0}\ AluOp_1\ AluOp_2\ F_1\ F_0$			⊗

$\therefore X = \overline{AluOp_0}\ \overline{AluOp_1}\ \overline{AluOp_2}\ F_1\ F_0 + \overline{AluOp_0}\ AluOp_1\ \overline{AluOp_2}\ F_1\ \bar{F}_0 + \overline{AluOp_0}\ AluOp_1\ AluOp_2\ F_1\ F_0$

Simplifying Y using Q.M:

Step-01:

Group	Term	$Alu op_0$	$Alu op_1$	$Alu op_2$	F_1	F_0
group-0 (0 number of 1's)	0	0	0	0	0	0
group-1 (two number of 1's)	3	0	0	0	1	1
group-2 (three number of 1's)	10	0	1	0	1	0
	14	0	1	1	1	0

$\therefore$ P.I's: $\overline{Alu op_0}\ \overline{Alu op_1}\ \overline{Alu op_2}\ \bar{F_1}\ \bar{F_0} + \overline{Alu op_0}\ \overline{Alu op_1}\ Alu op_2\ F_1\ F_0$

Step-02:

Group	Matched pairs	$Alu op_0$	$Alu op_1$	$Alu op_2$	F_1	F_0
group-1	10 - 14	0	1	-	1	0

$\therefore$ P.I's: $\overline{Alu op_0}\ Alu op_1\ F_1\ \bar{F_0}$

Step-03: P.I Table

P.I	minterms	0	3	10	14
$\overline{Alu op_0}\ \overline{Alu op_1}\ \overline{Alu op_2}\ \bar{F_1}\ \bar{F_0}$	0	⊗			
$\overline{Alu op_0}\ \overline{Alu op_1}\ Alu op_2\ F_1\ F_0$	3		⊗		
$\overline{Alu op_0}\ Alu op_1\ F_1\ \bar{F_0}$	10-14			⊗	⊗

$\therefore Y = \overline{Alu op_0}\ \overline{Alu op_1}\ \overline{Alu op_2}\ \bar{F_1}\ \bar{F_0} + \overline{Alu op_0}\ \overline{Alu op_1}\ Alu op_2\ F_1\ F_0 + \overline{Alu op_0}\ Alu op_1\ F_1\ \bar{F_0}$

simplifying z using -QM:

Step-01:

group	Term	$AluOp_0$	$AluOp_1$	$AluOp_2$	F_1	F_0
group-0 (one number of 1's)	2	0	0	0	1	0 ✓
group-1 (two number of 1's)	3	0	0	0	1	1 ✓
group-2 (three number of 1's)	14	0	1	1	1	0

∴ P.I's: $\overline{AluOp_0}\ AluOp_1\ AluOp_2\ F_1\ \overline{F_0}$

Step-02:

group	matched Pairs	$AluOp_0$	$AluOp_1$	$AluOp_2$	F_1	F_0
group-0	2-3	0	0	0	1	-

∴ P1: $\overline{AluOp_0}\ \overline{AluOp_1}\ \overline{AluOp_2}\ F_1$

Step-04: P.I Table

P-I	minterm	2	3	14
$\overline{AluOp_0}\ AluOp_1\ AluOp_2\ F_1\ \overline{F_0}$	14			⊗
$\overline{AluOp_0}\ \overline{AluOp_1}\ \overline{AluOp_2}\ F_1$	2-3	⊗	⊗	

∴ $Z = \overline{AluOp_0}\ AluOp_1\ AluOp_2\ F_1\ \overline{F_0} + \overline{AluOp_0}\ \overline{AluOp_1}\ \overline{AluOp_2}\ F_1$

Result of simulation:

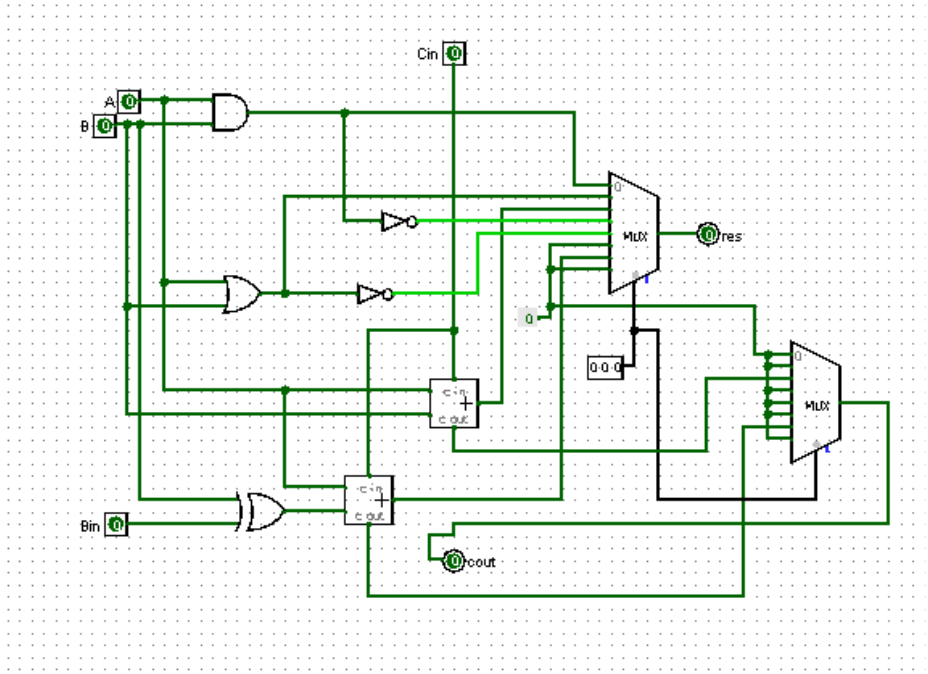

Figure-1: Design of 1-bit ALU

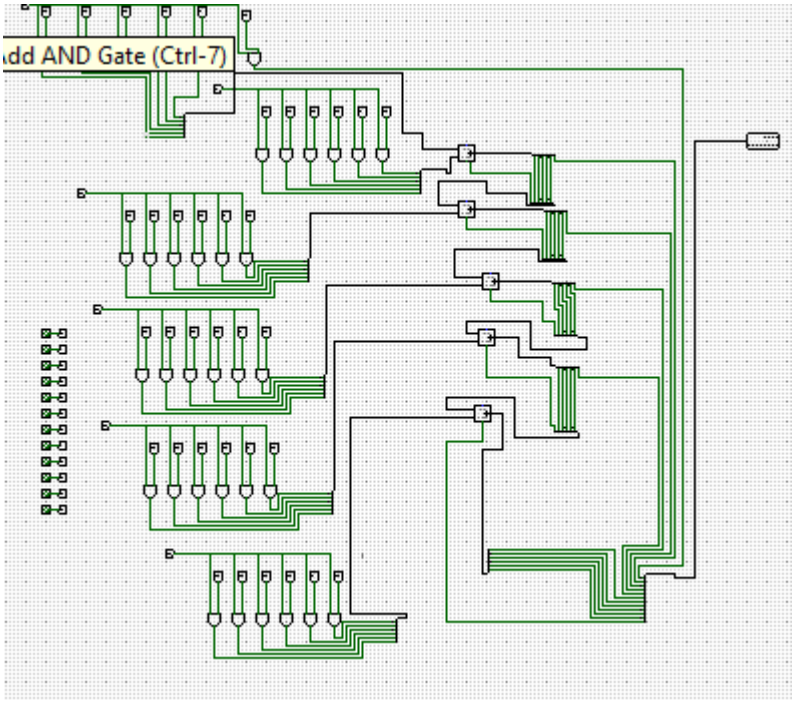

Figure-2: 6-bit by 6 bit (12 bit) multiplier circuit

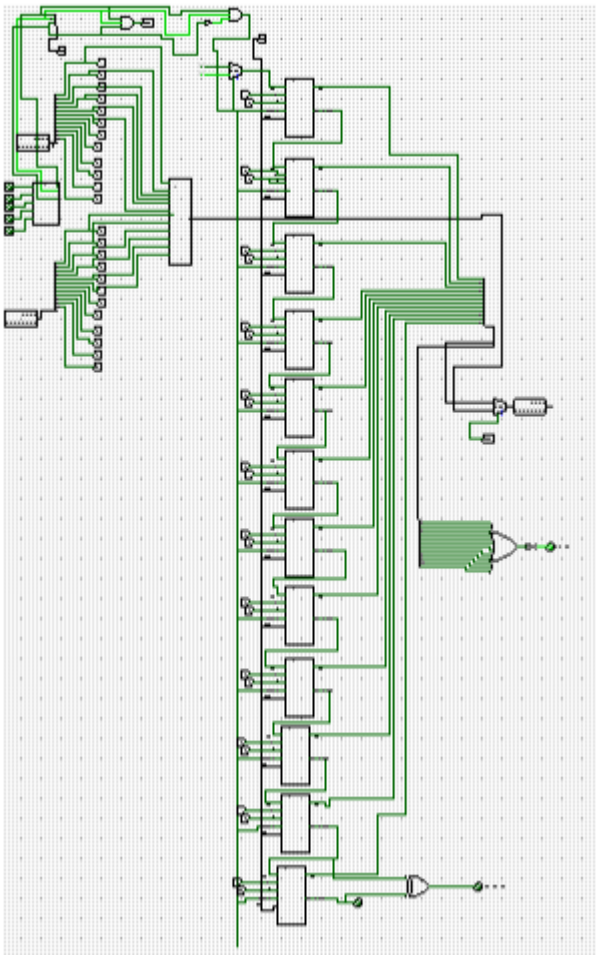

Figure-3: 12-bit ALU

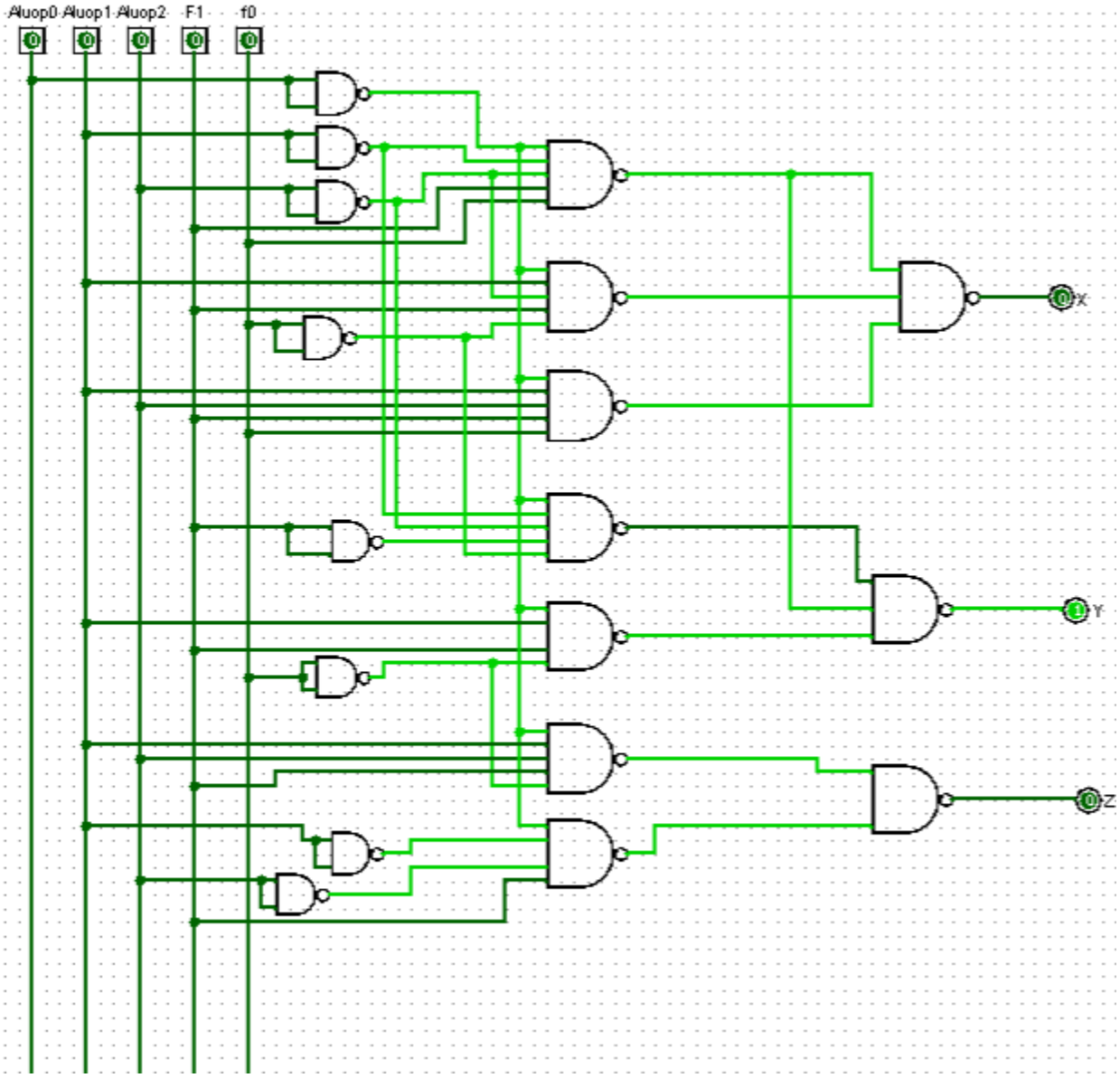

Figure-4: ALU Control Circuit (Nand-Nand)

Discussion:

In this part of my project, the objective was to design a 12-bit Alu, 12-bit multiplier and a control which can operate the Alu. Here the Alu may not perform all the possible operations but it can perform and, or, add, subtract, nand, nor & multiplication.

Firstly, I designed a 1-bit Alu which has two inputs and a selector which can determine what to perform.

For multiplication, I had to design a 12-bit multiplier which can multiply two 6-bit binary numbers. To implement this multiplication unit in logisim at first, I created 2 six-bit input A and B. Then I used all six bit of B input, anded them with the least significant bit of input A. The first output directly goes into the result. Then I used six and gate with all six bits of B and second least significant bit of A. Then I used a six-bit full adder to add the first five and gate result and 0 input along with the last six and gate output result. The first bit of the summation goes in output bits. Then I continued this process for four more times, each time the first bit of summation will go to output and for the last addition all six bits and carry output will go to the output. As the design was a bit long so I used tunnel for less wiring by which I was able to design it quickly and easily,

Now to construct my 12-bit Alu, I used 12 1-bit Alu's and I put them in such order that 1st Alu's carry out is connected with 2^{nd} Alu's carry in and so on. It can show 12-bit results of particular operations which are mentioned above. Here I have a zero detector as well. If the result is 0 than it will show 1 by which we will get to know that the result is 0. There is an over flow detector as well which works just like zero detector. If there is an over flow than it will show result 1.

Improvements:

There were some limitations in my project part 1. Those were: I didn't build my control circuit using nand-nand / nor-nor gates as in real life we can't find 5 input and gates. Also, I didn't connect my control circuit with my main design which is 12-bit alu. This time I have build my control circuit using nand-nand gates and also connected it with my 12-bit Alu.