Name-Md. Sayadul Haque

Id-1421353042

Course-Cse 332

Section-9

Faculty-SMH2

Project Part-1

Objective:

- Understanding the basic concept of logic functions (Nand, Nor, Xor), Arithmetic functions (Add with carry, subtract with borrow) and implement these functions in a single ALU design for 2 bits.
- Understanding the concept of designing 32-bit ALU from 2-bit ALU.
- Understanding the concept of designing Multiplication unit for 16 bits multiplicand and 16 bits multiplier.
- Understanding the concept of designing control unit for ALU and multiplication unit.

Operational Table:

B invert	A invert	Carry in	Function value, F1	Function Value, F0	ALU Operation	Functions
1	1	0	1	0	011	NAND
1	1	0	1	1	100	NOR
0	0	0	0	0	101	XOR
0	0	1	0	1	010	Add with carry
1	0	0	1	0	110	Subtract with borrow
0	0	0	1	0	111	Multiplication

Process of designing control unit:

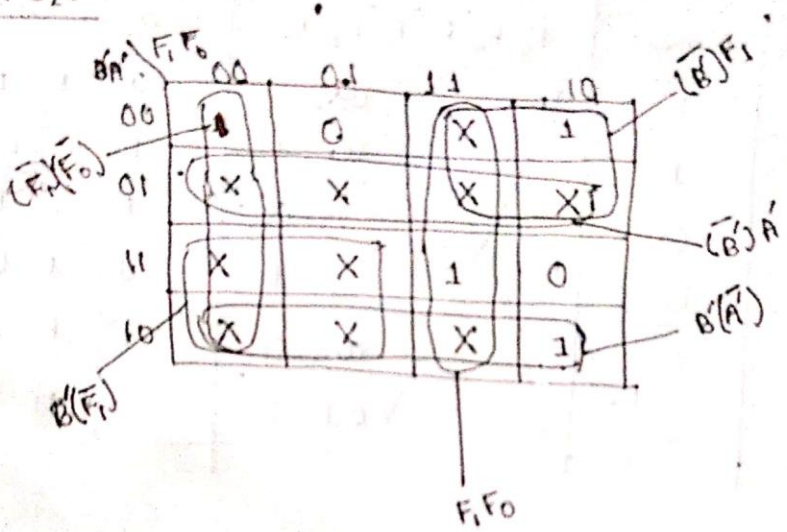
Alu operation		Function field, Fx						3 bits operation			
B invert	A invert	F_5	F_4	F_3	F_2	F_1	F_0	O_2	O_1	O_0	
1	1					(1) X	0	0	1	1	(nand)
1	1					(1) X	1	1	0	0	(nor)
0	0					0	0	1	0	1	(XOR)
0	0					0	1	0	1	0	(Add with carry)
1	0					(1) X	0	1	1	0	(Subtract with borrow)
0	0					1	0	1	1	1	(Multiplication)

Now,

	O_2	O_1	O_0
✓0000	1	0	1
✓0001	0	1	0
✓0010	1	1	1
0011	X	X	X
0100	X	X	X
0101	X	X	X
0110	X	X	X
0111	X	X	X
1000	X	X	X
1001	X	X	X
✓1010	1	1	0
1011	X	X	X
1100	X	X	X
1101	X	X	X
✓1110	0	1	1
✓1111	1	0	0

So,

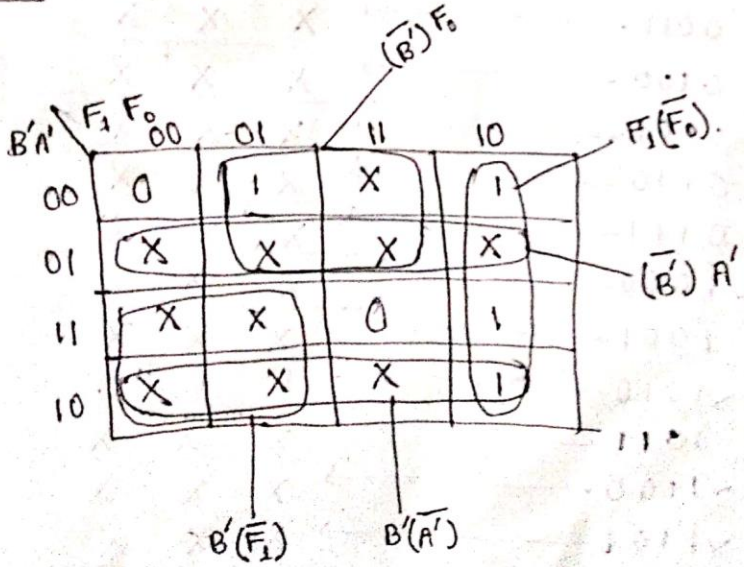
For O_2:



So, $O_2 = B'(\overline{F_1}) + (\overline{B'})F_1 + (\overline{B'})A' + B'(\overline{A'}) + (\overline{F_1})(\overline{F_0})$

$= (B' \oplus F_1) + (B' \oplus A') + (\overline{F_1})(\overline{F_0})$

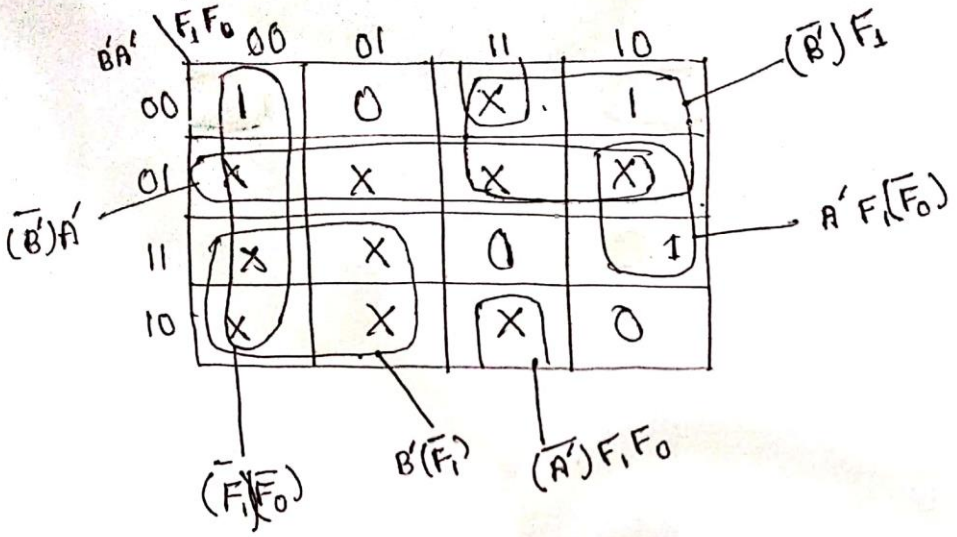
For O_1:



So, $O_1 = (\overline{B'})A' + B'(\overline{A'}) + B'(\overline{F_1}) + (\overline{B'})F_0 + F_1(\overline{F_0})$

$= (B' \oplus A') + B'(\overline{F_1}) + (\overline{B'})F_0 + (\overline{F_0})F_1$

For O_0:



So, $O_0 = B'(\bar{F}_1) + (\bar{B'})F_1 + (\bar{B'})A' + (\bar{F}_1)(\bar{F}_0) + A'F_1(\bar{F}_0)$

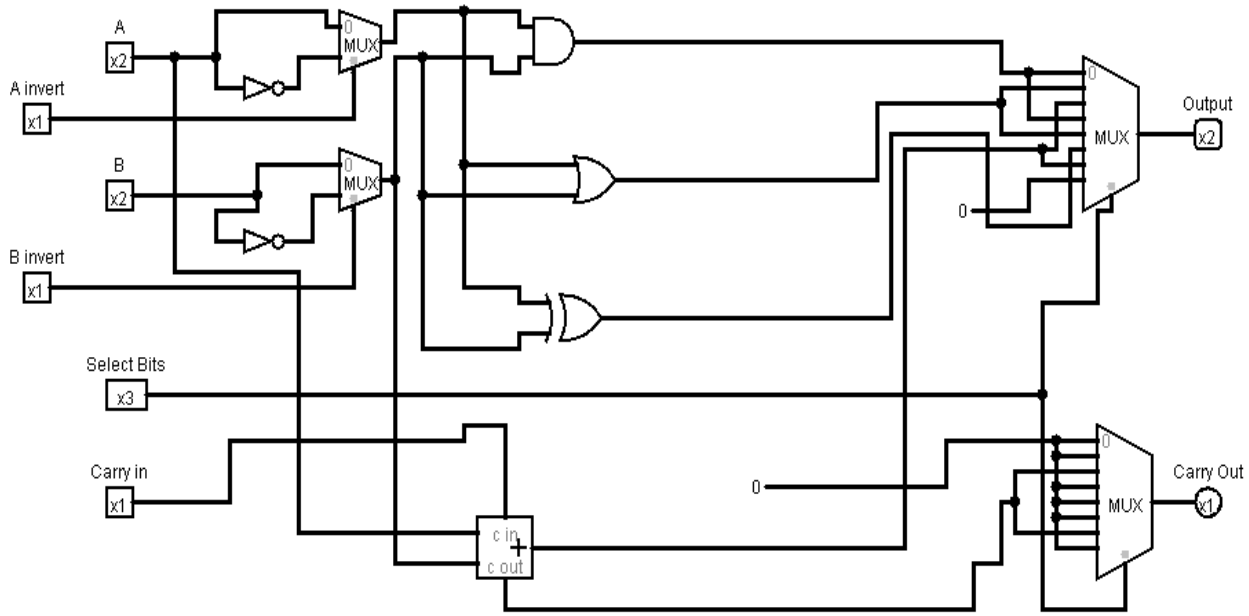
$+ (\bar{A'})F_1F_0$

$= (B' \oplus F_1) + (\bar{B'})A' + (\bar{F}_1)(\bar{F}_0) + A'F_1(\bar{F}_0)$

$+ (\bar{A'})F_1F_0$

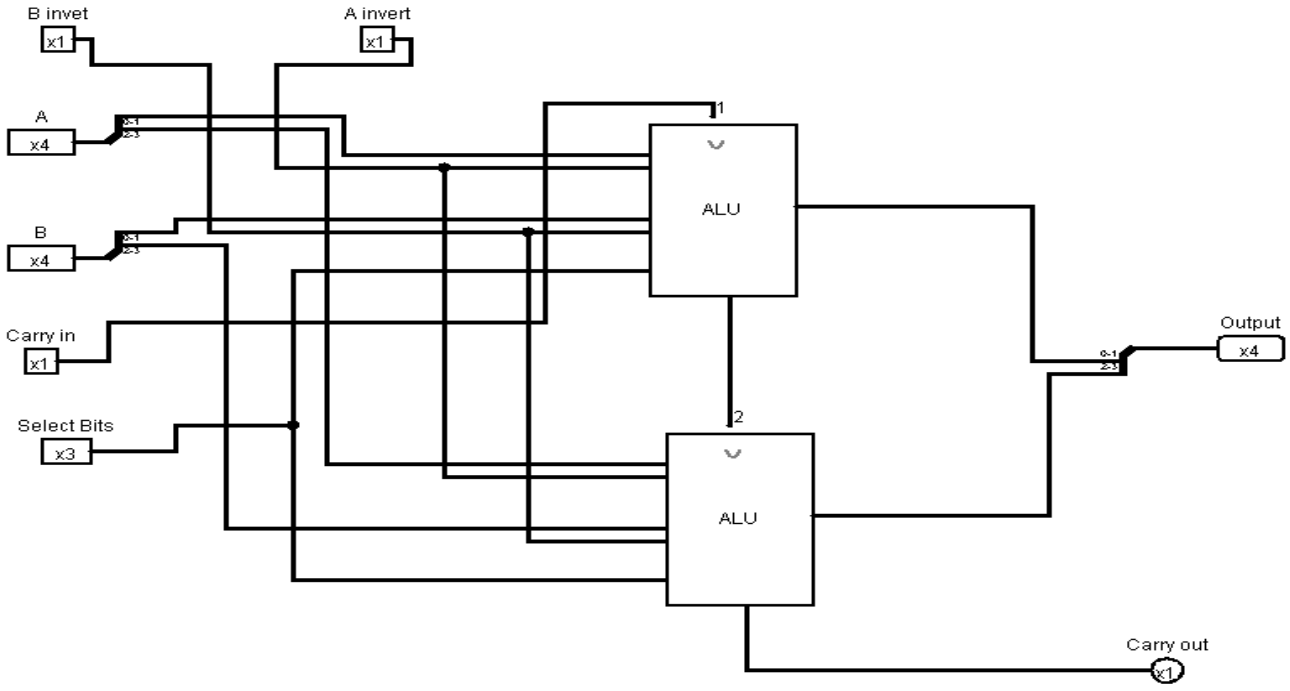
Circuit Design:

2-bits ALU for Nand, Nor, XOR, add with carry, subtract with Borrow functions:

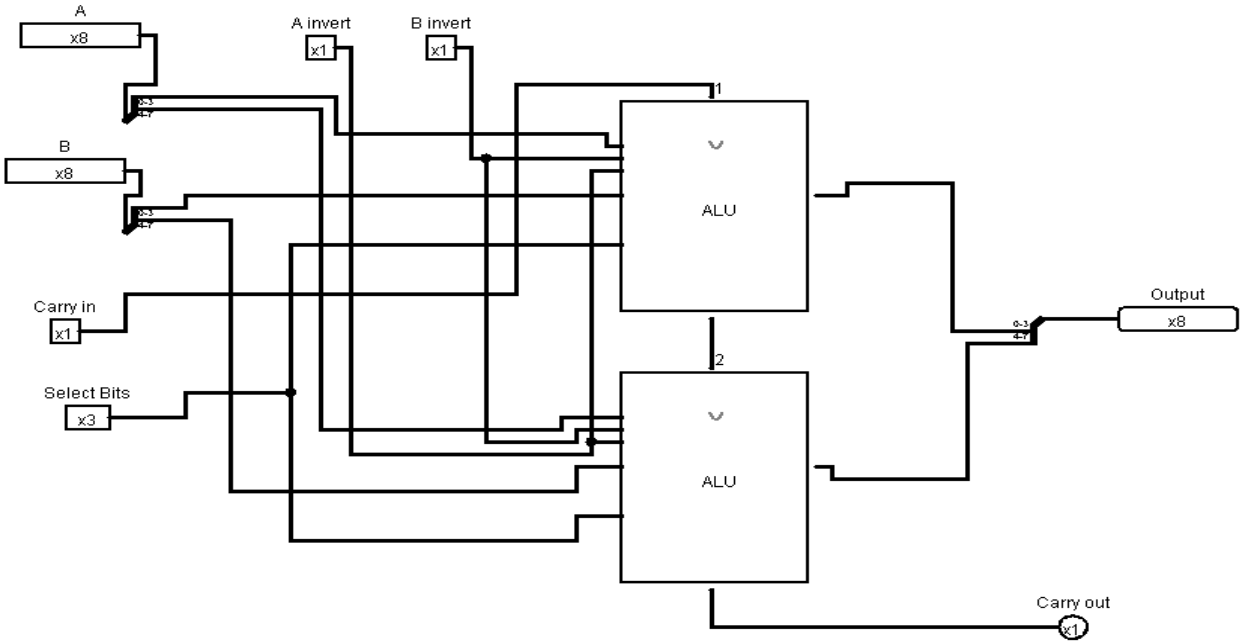


Cascading the above 2-bit ALU for to implement a 32-bits MIPS ALU:

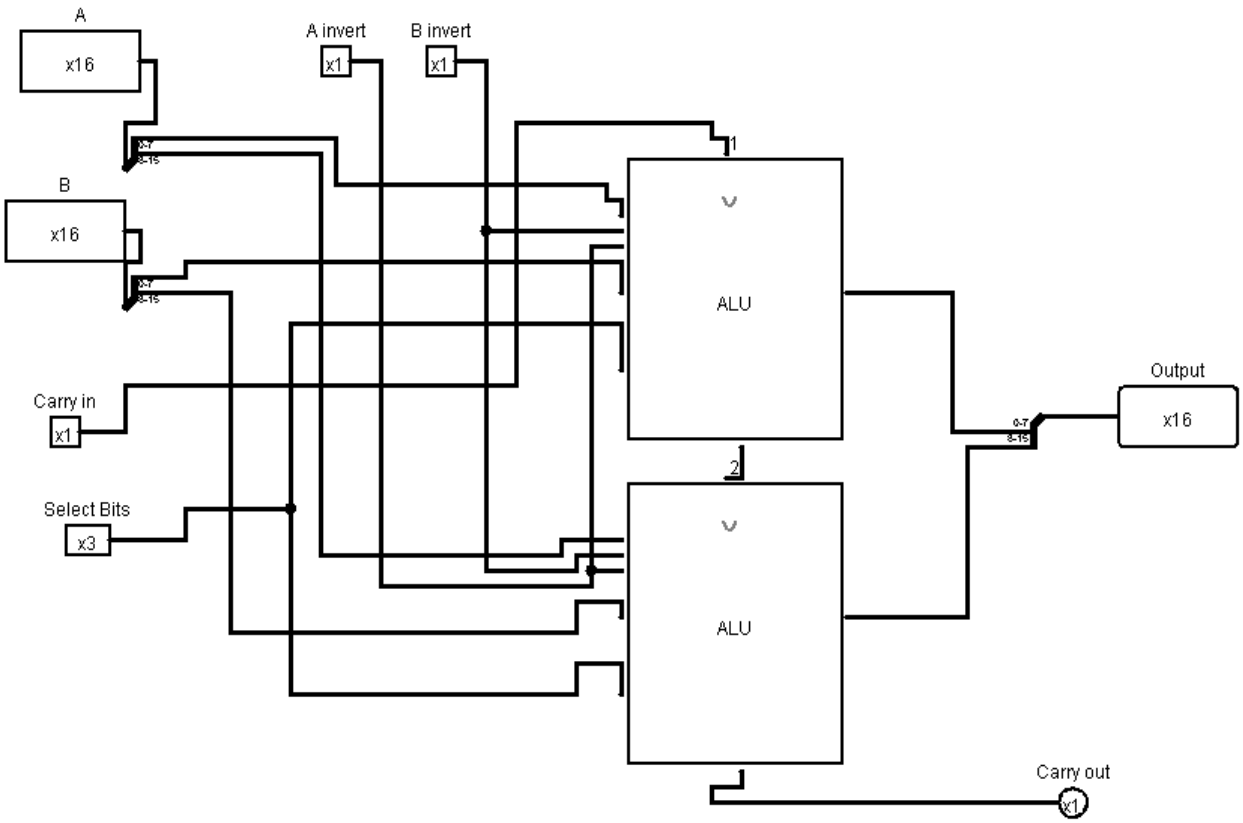
2 to 4-bits ALU:



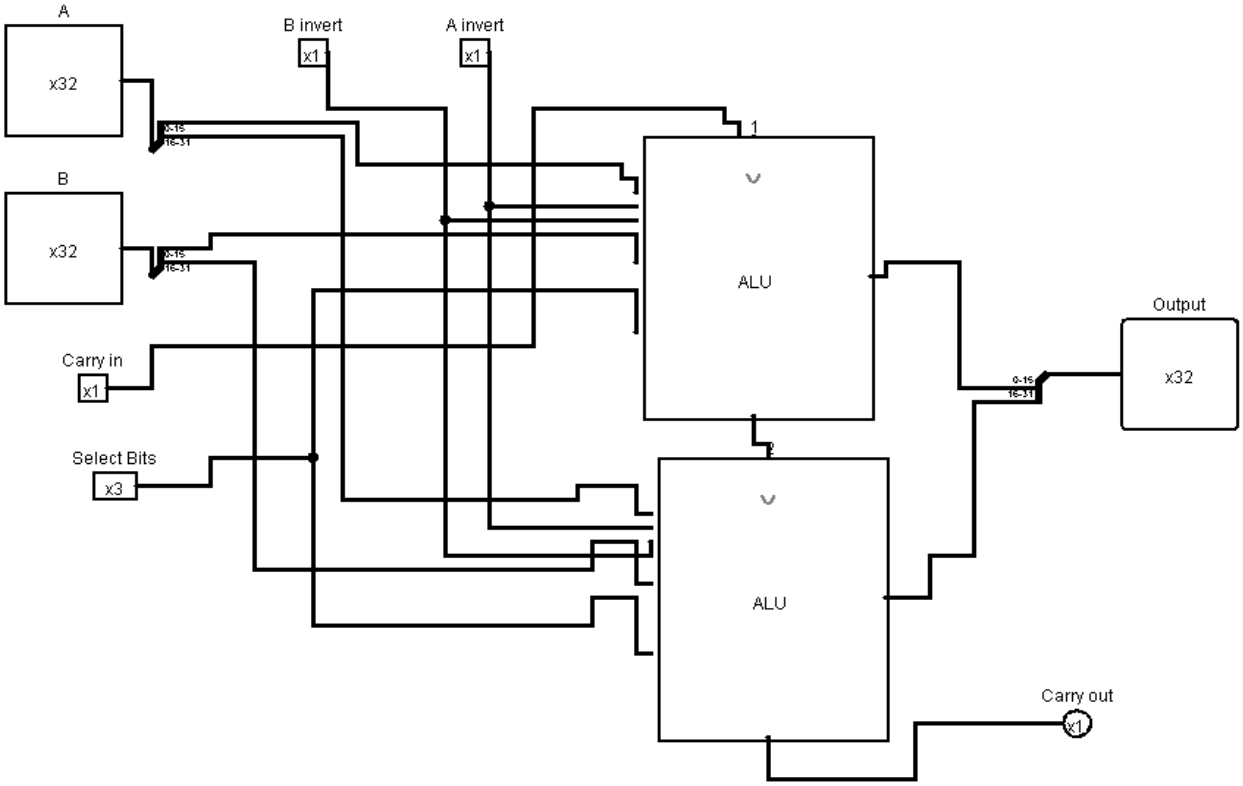
4 to 8-bits ALU:



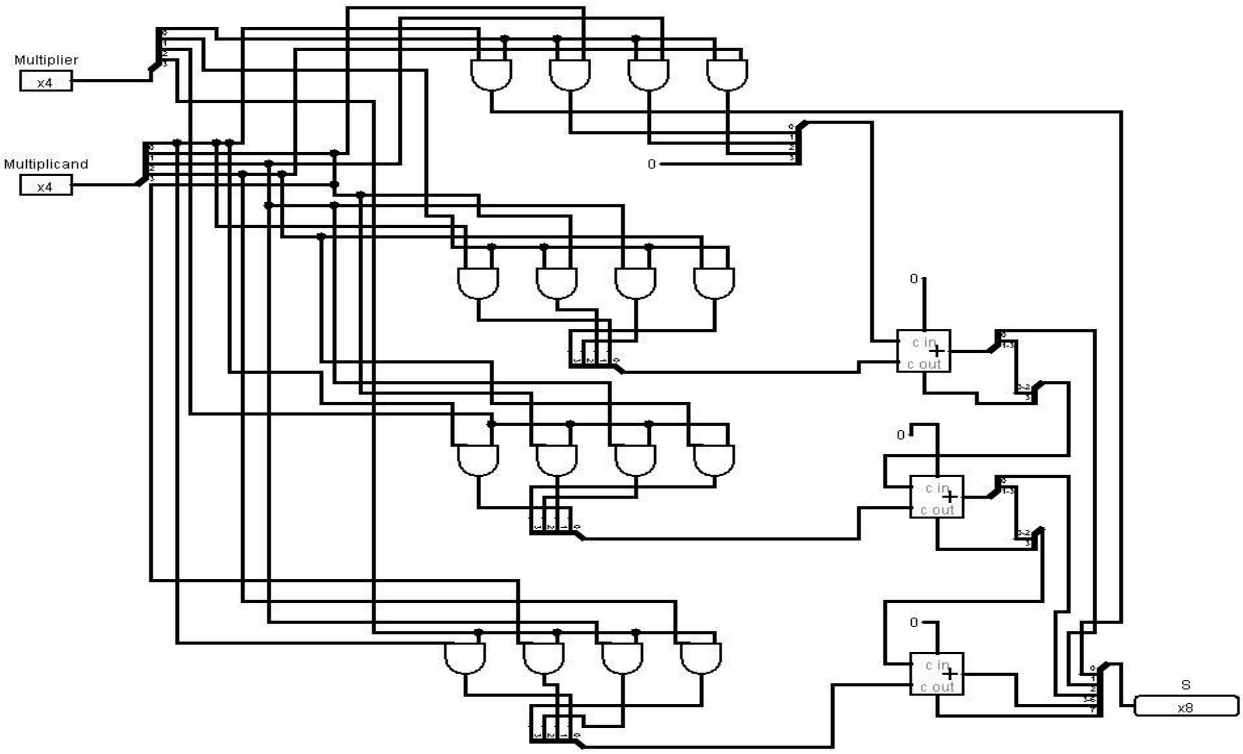
8 to 16-bits ALU:



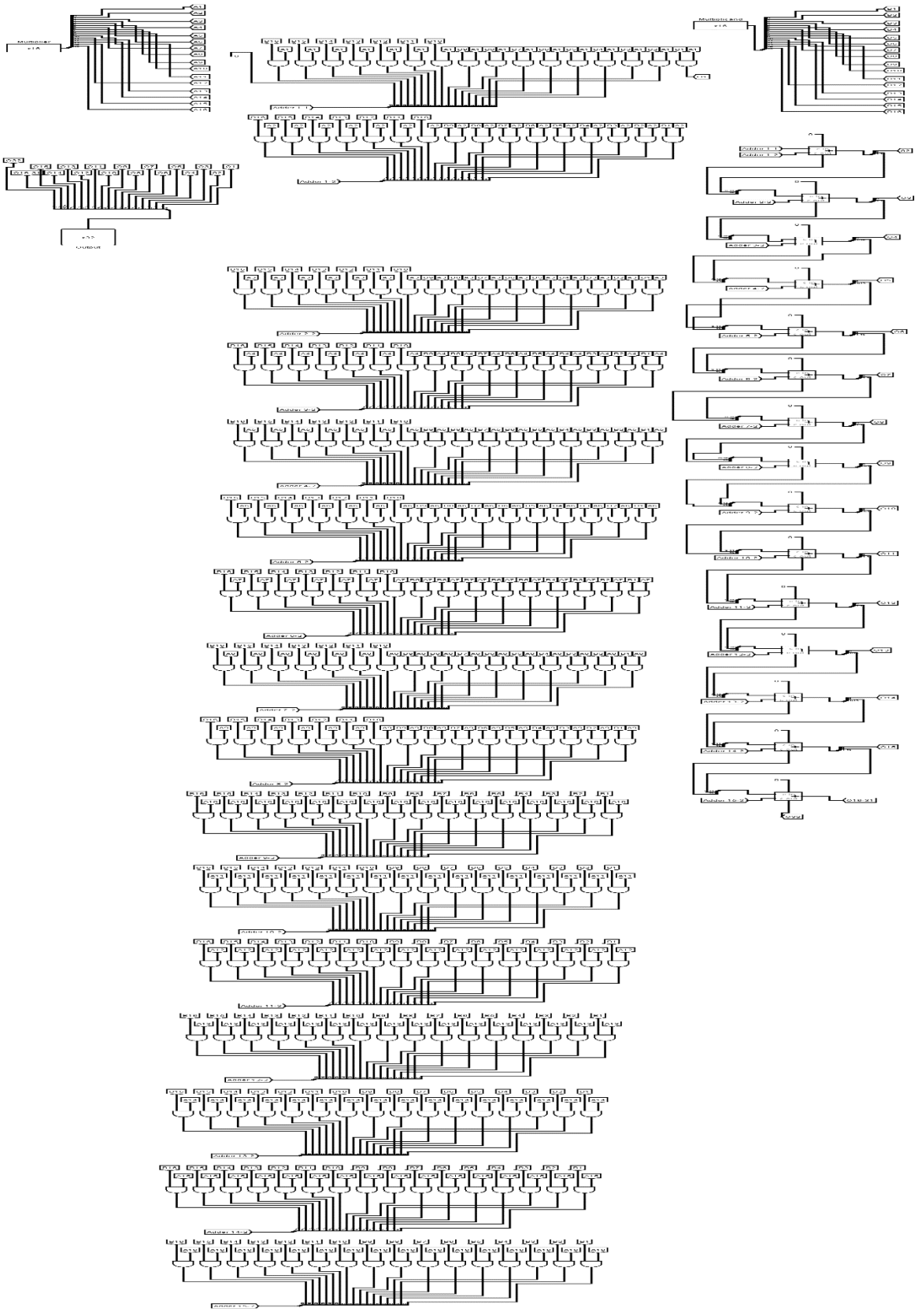
16 to 32-bits ALU:



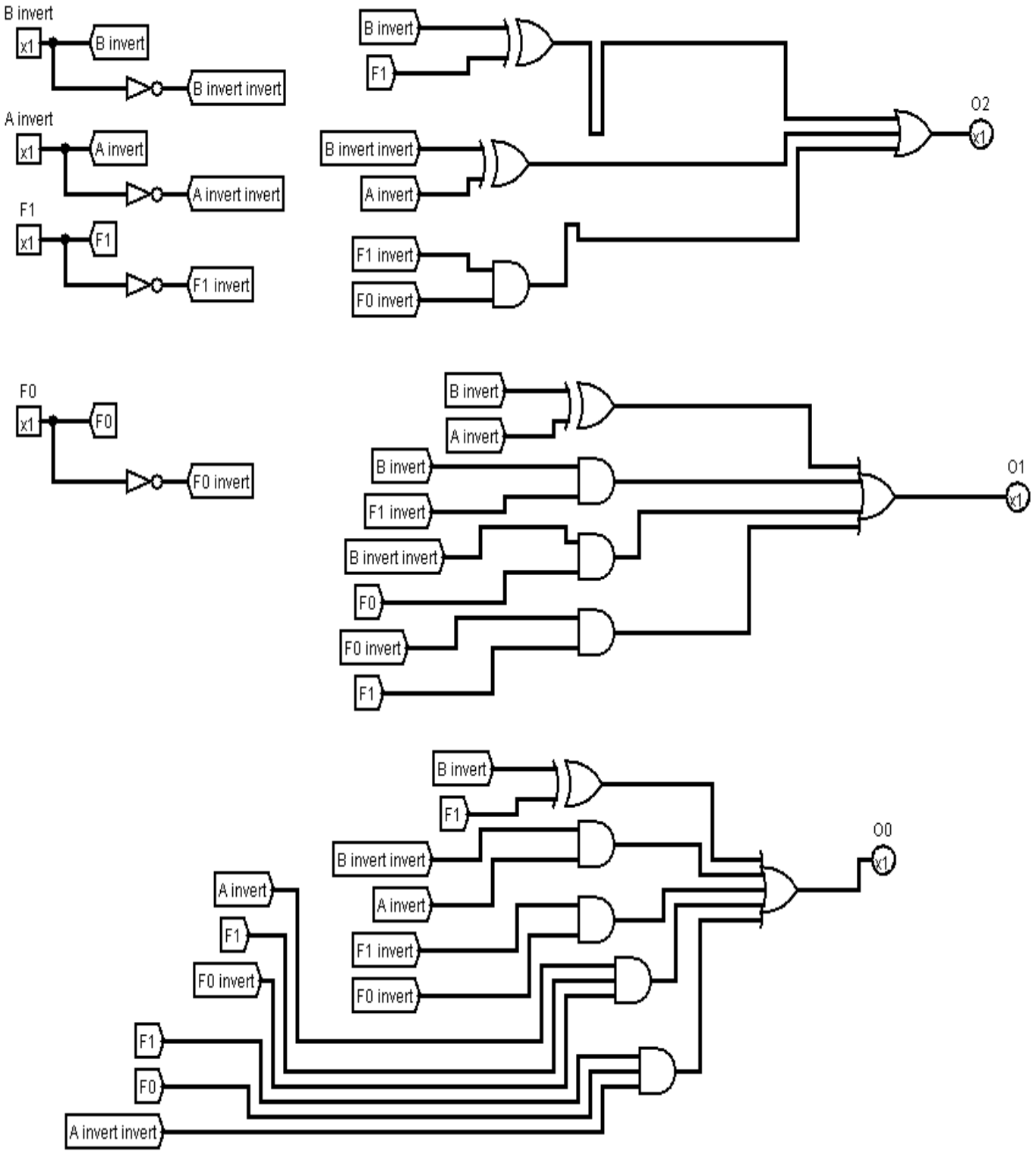
4-bits by 4-bits Multiplication Unit:



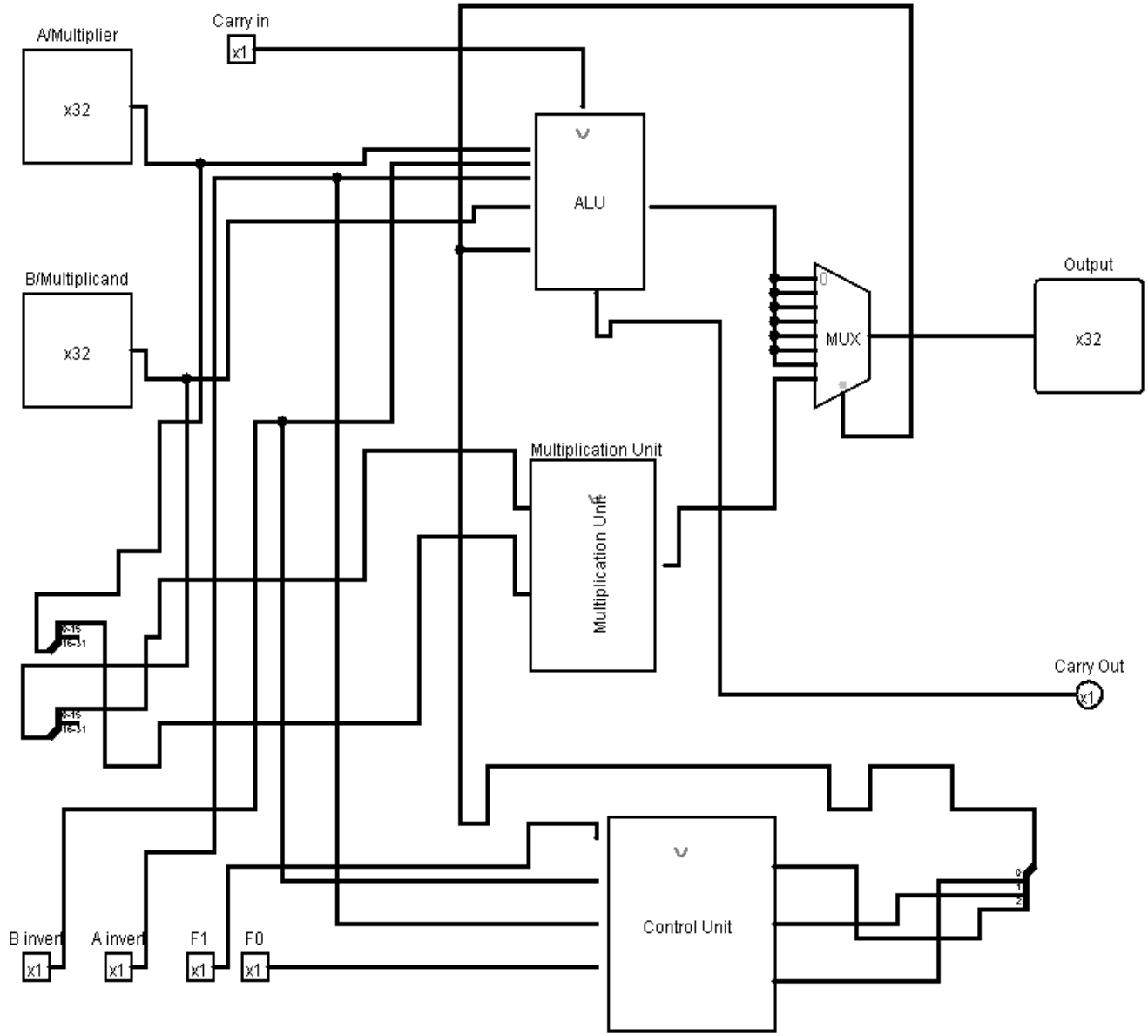
16-bits by 16-bits Multiplication Unit:



Control Unit:



Master Unit:



Discussion:

It should be understood that designing a 2-bit ALU for different types of logic and arithmetic function is a creative process Ma-sha-Allah. We used NAND, NOR, XOR, ADD WITH CARRY, SUBTRACT WITH BORROW functions for designing our ALU. Our used functions are basic operations of our ALU that can be performed by our designed ALU. We also designed a 32-bits ALU from our basic 2-bits ALU using Logisim. We used some steps for designing our 32-bit ALU (2-bits ALU=> 4-bits

ALU =>8-bits ALU=>16-bits ALU => 32-bits ALU). We also designed 4-bits by 4-bits Multiplication unit for showing the basic design of multiplication unit. Then we designed 16-bits by 16-bits multiplication unit for using it in our master unit design. We showed our process of designing control unit for our master unit in this report. After seeing the operational table, it should be understood what you need to input in master unit with you inputs A & B for finding the output S of your desired functions. Thank you.
