Name-Md. Sayadul Hoque

Id-1421353042

Course-Cse 332

Section-9

Faculty-SMH2

Project Part-2

Objective:

- Design a 32-bit Instruction Fetch Unit of the Datapath.
- Design a R-format and I-format Datapath.
- Compose the designed Datapath segments to yield a complete single cycle Datapath.
- Design a 32-bit ALU with control for a single cycle Datapath.

Experiment tool:

Logisim tool.

Experiment details:

R -format:

Op (31-26)	Rs (25-21)	Rt (20-16)	Rd (15-11)	Shamt (10-6)	Function (5-0)

I-format (Load/Store & Branch Operation):

Op (31-26)	Rs (25-21)	Rt (20-16)	Address (15-0)

Operational Table:

R-format:
Opcode- 000000
I-format:
Load opcode-100011
Store opcode-101011
Branch (beq) -000100

Alu Operation:

Operation	Bin-Ain-F1-F0
Add (With Carry)	0001
Sub (With borrow)	1010
Nand	1110
Nor	1111
Xor	0000
Multiplication	0010

32- bit Register:

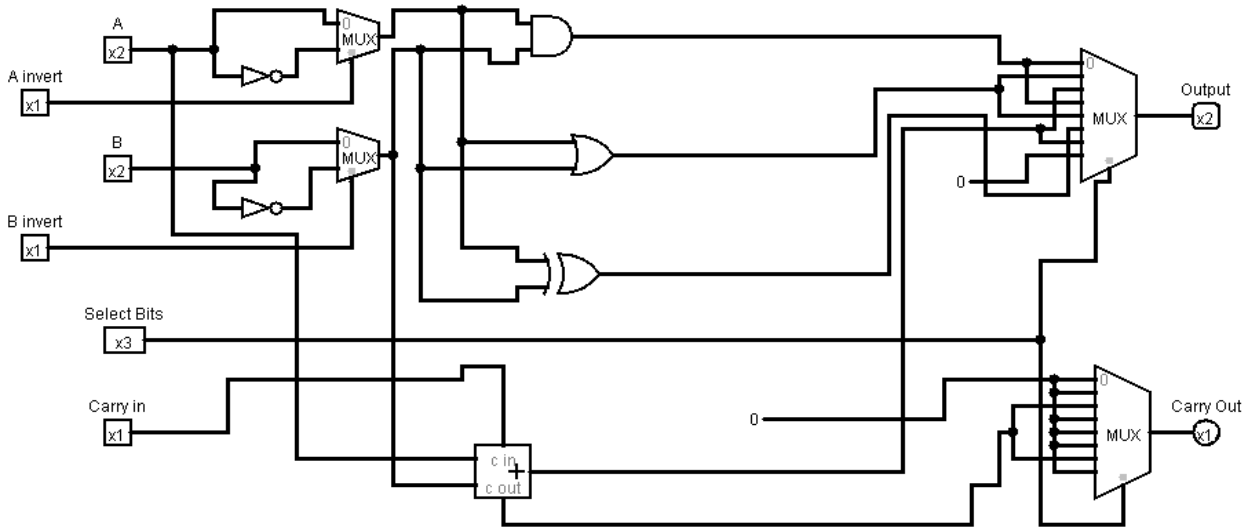
Register name	Register number
$Zero	0
$at	1
$v0	2
$v1	3
$a0	4
$a1	5
$a2	6
$a3	7
$t0	8
$t1	9
$t2	10
$t3	11
$t4	12
$t5	13
$t6	14
$t7	15
$s0	16
$s1	17
$s2	18
$s3	19
$s4	20
$s5	21
$s6	22
$s7	23
$t8	24
$t9	25
$k0	26
$k1	27
$gp	28
$sp	29
$fp	30
$ra	31

Instruction:

Instruction number	MIPS Code	Machine Code	Hexa Value
Add (00)	Add $t3, $t1, $t2	0000000 10010 10100 10110 00001 00000	012A5820
Sw (04)	Sw $t0,0(t1)	101011 01001 01000 00000 00000 000000	AD280000

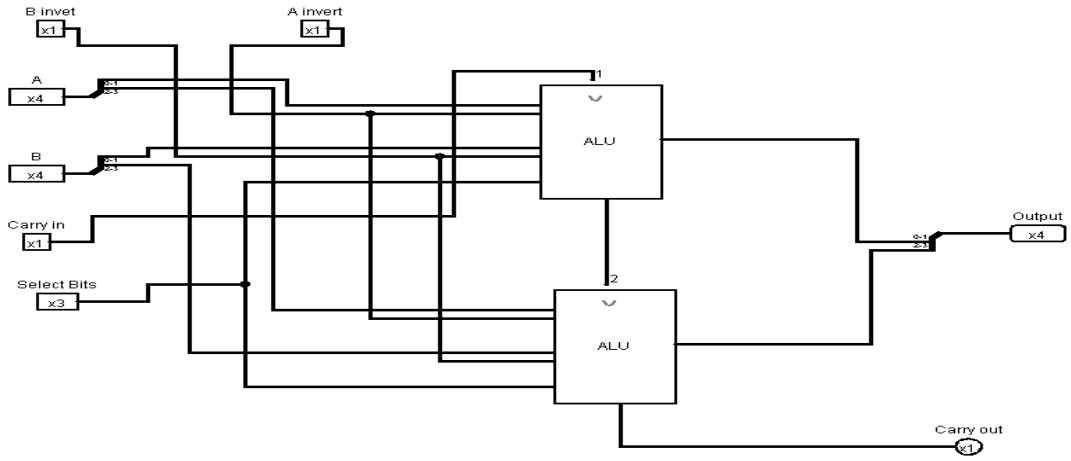
Circuit Design:

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

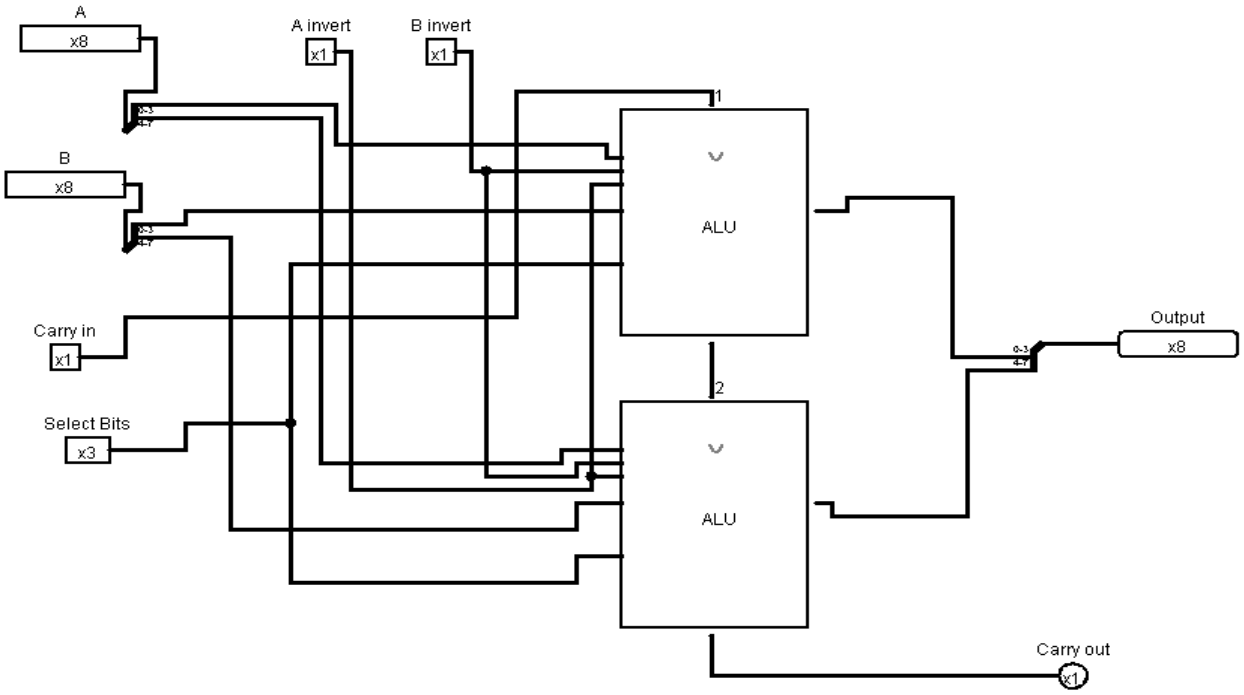


Cascading the above 2-bit ALU for to implement a 32-bits MIPS ALU:

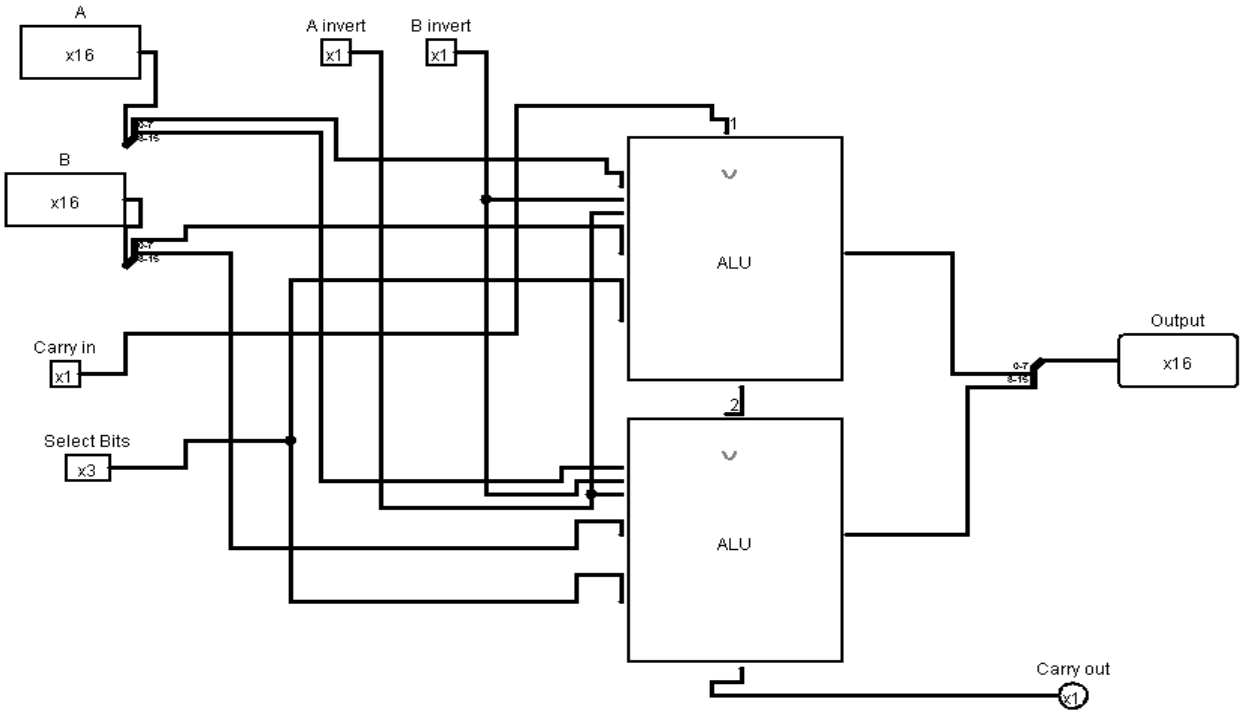
2 to 4-bits ALU:



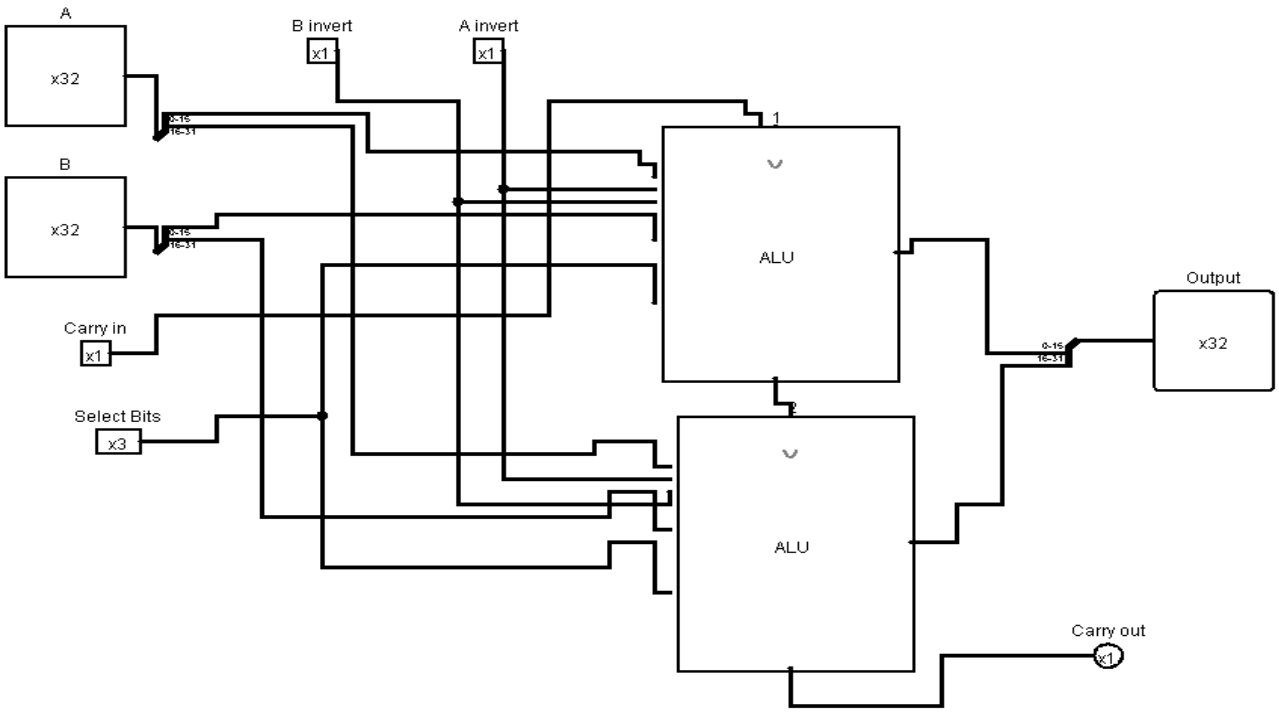
4 to 8-bits ALU:



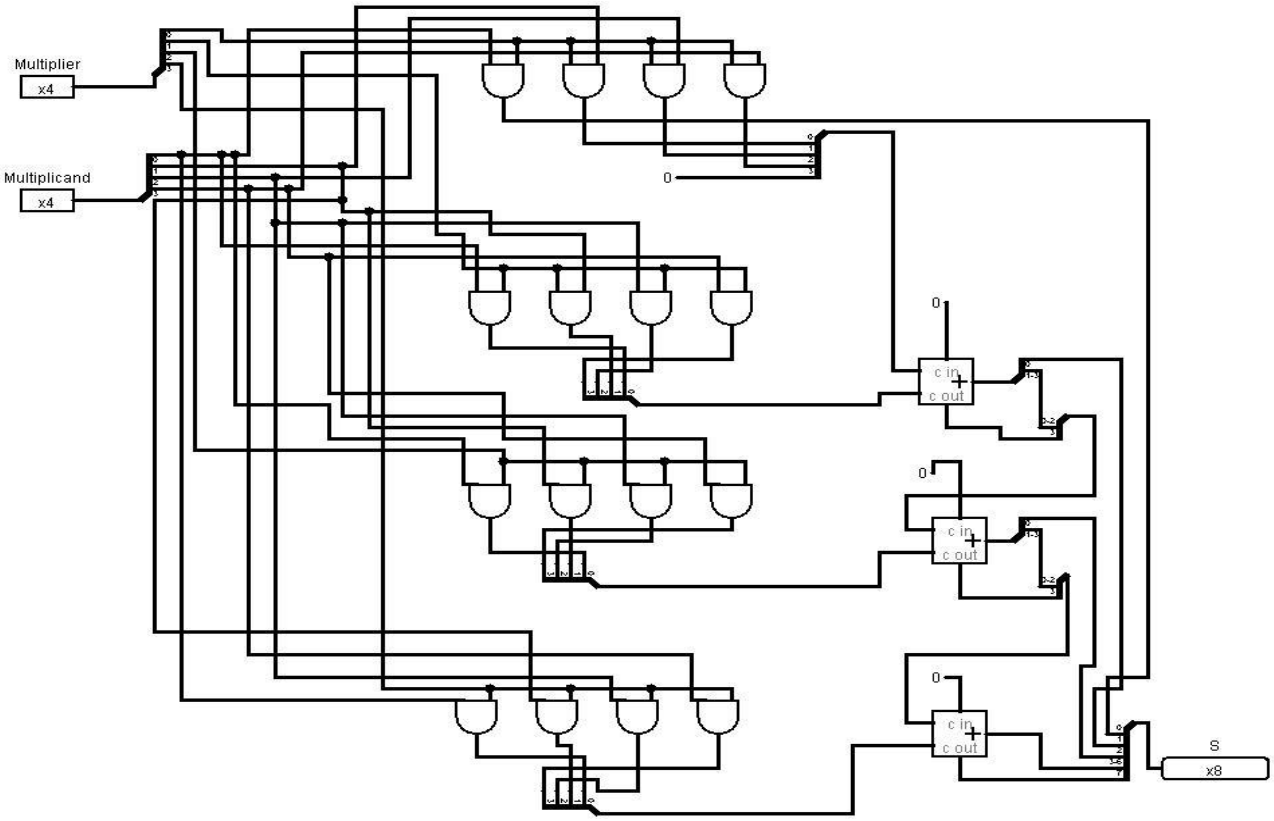
8 to 16-bits ALU:



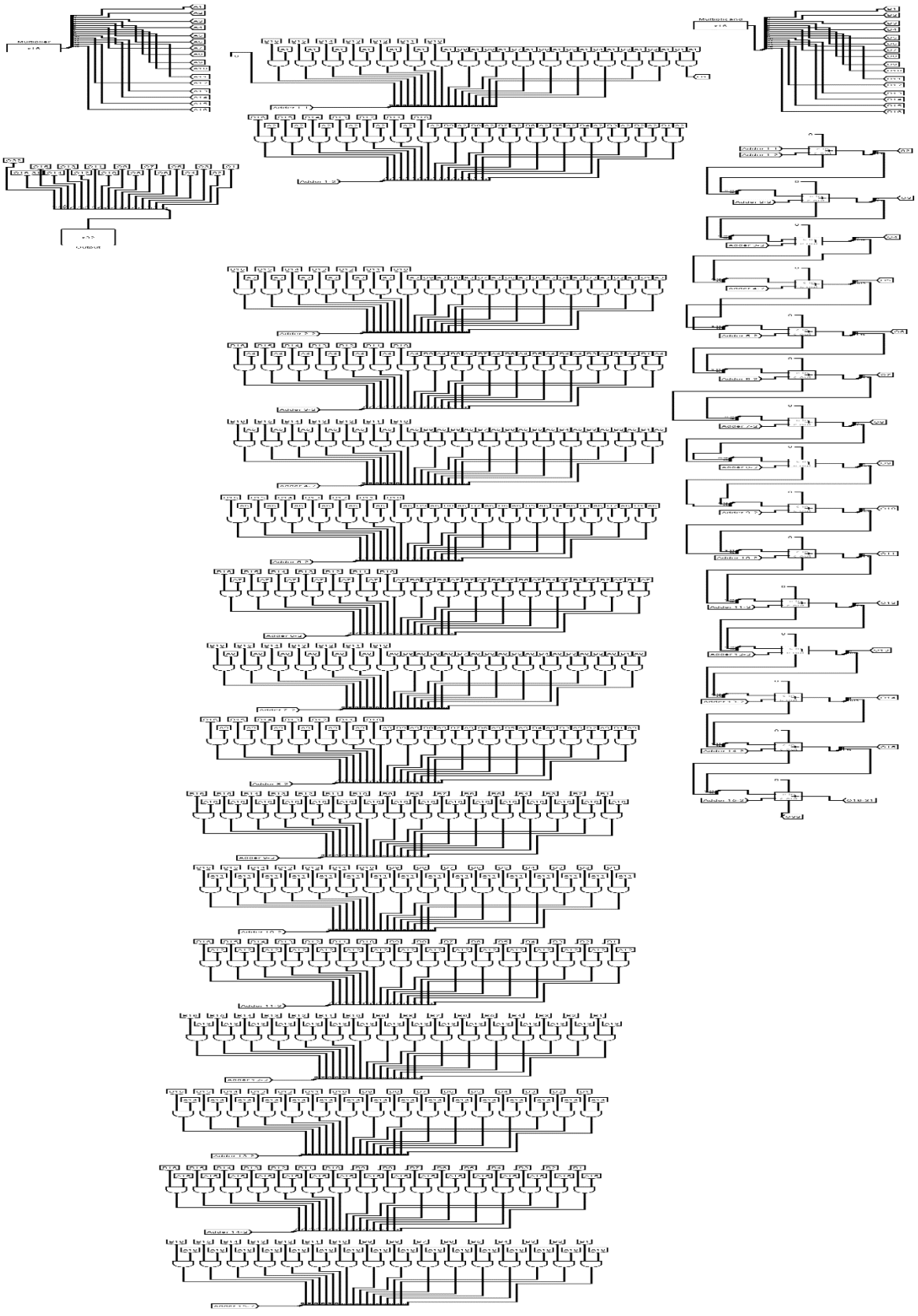
16 to 32-bits ALU:



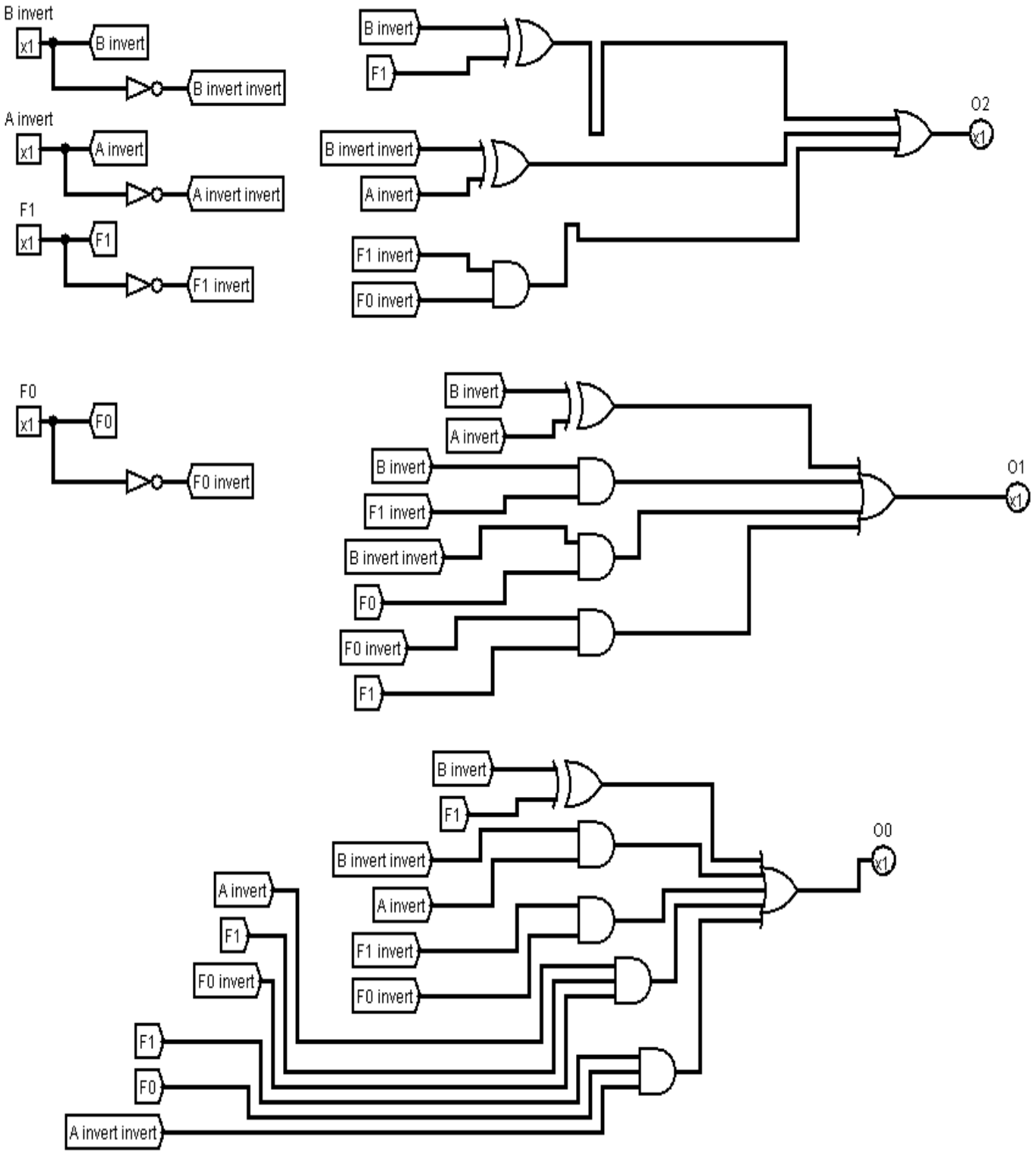
4-bits by 4-bits Multiplication Unit:



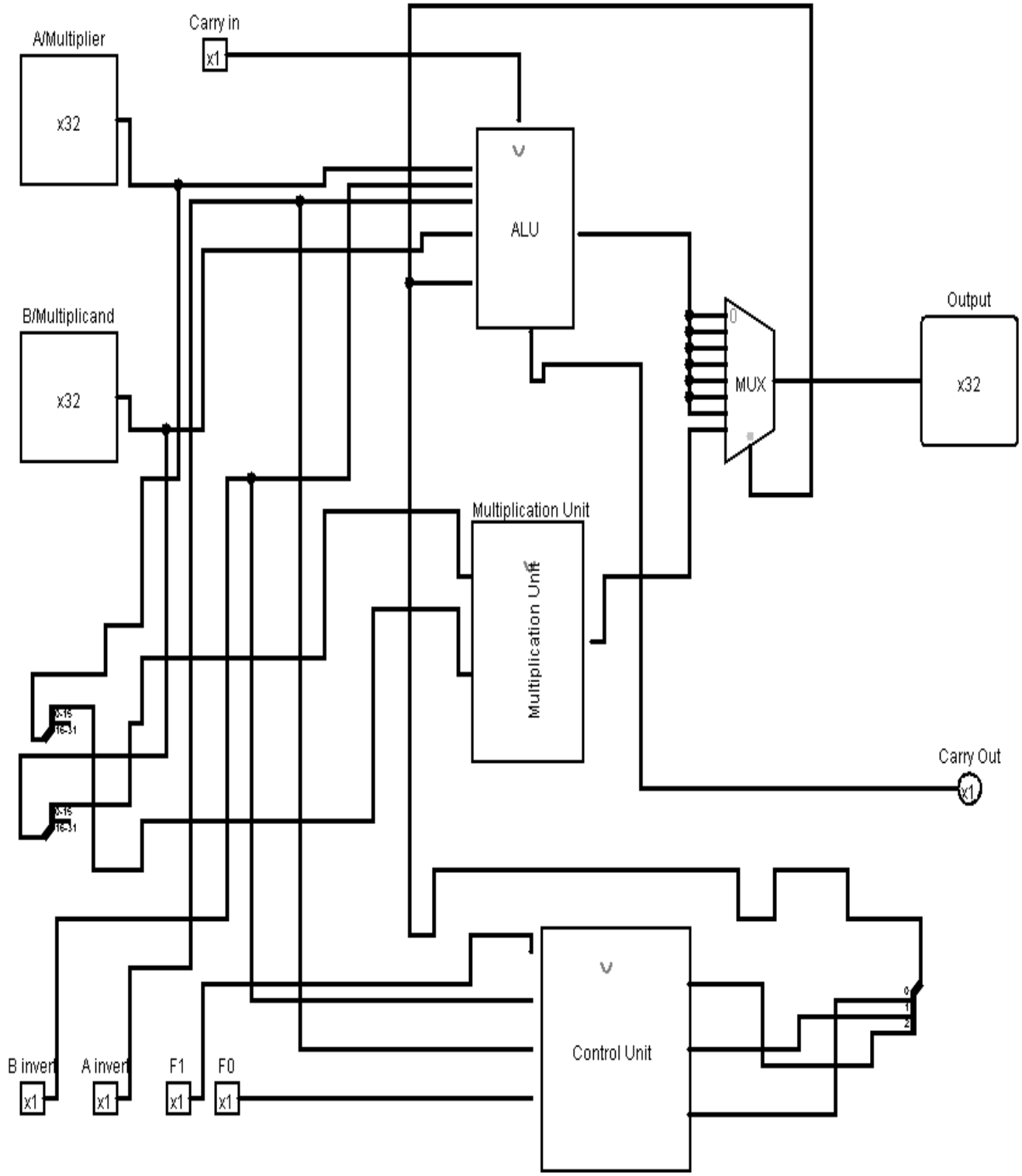
16-bits by 16-bits Multiplication Unit:



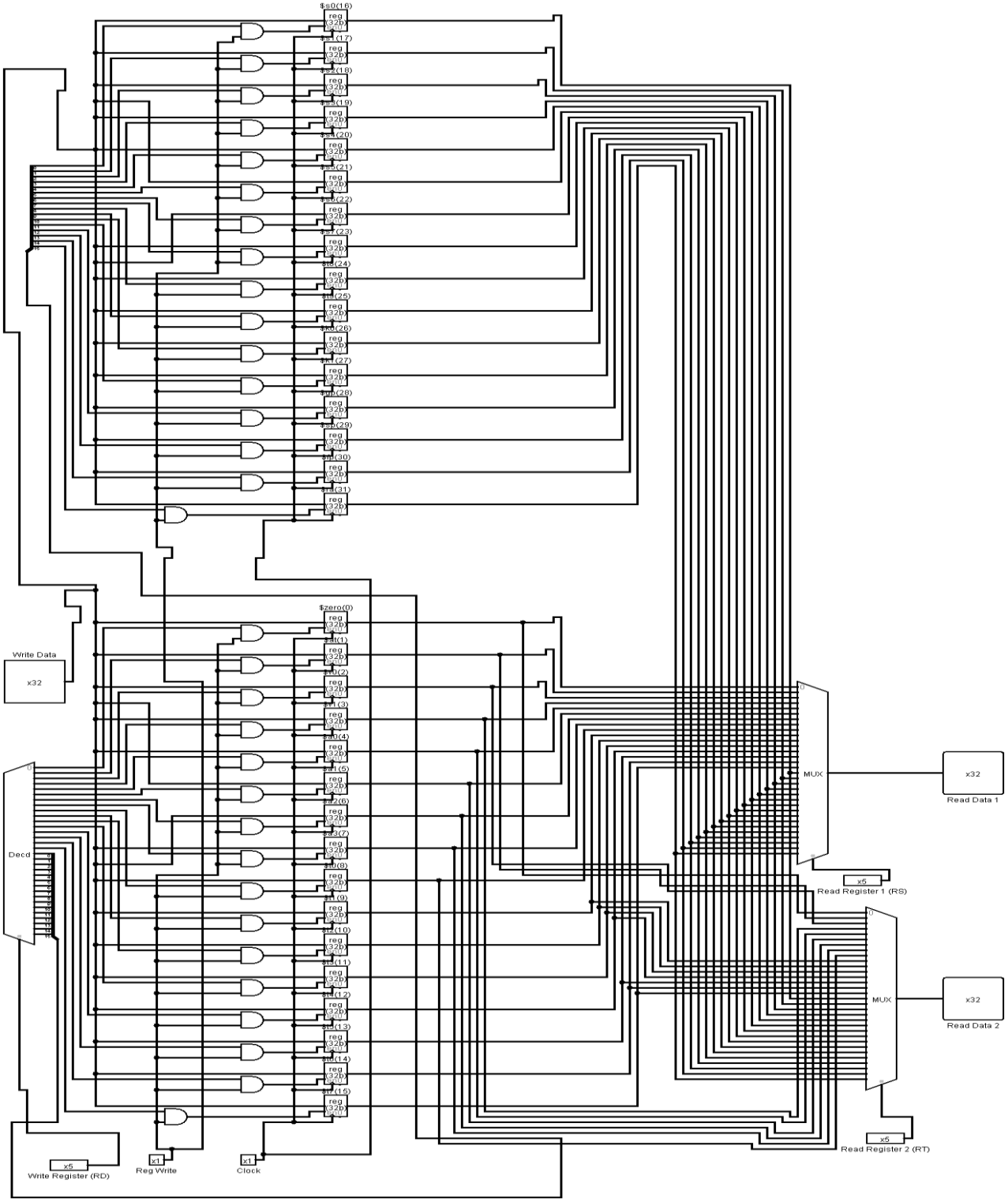
Control Unit:



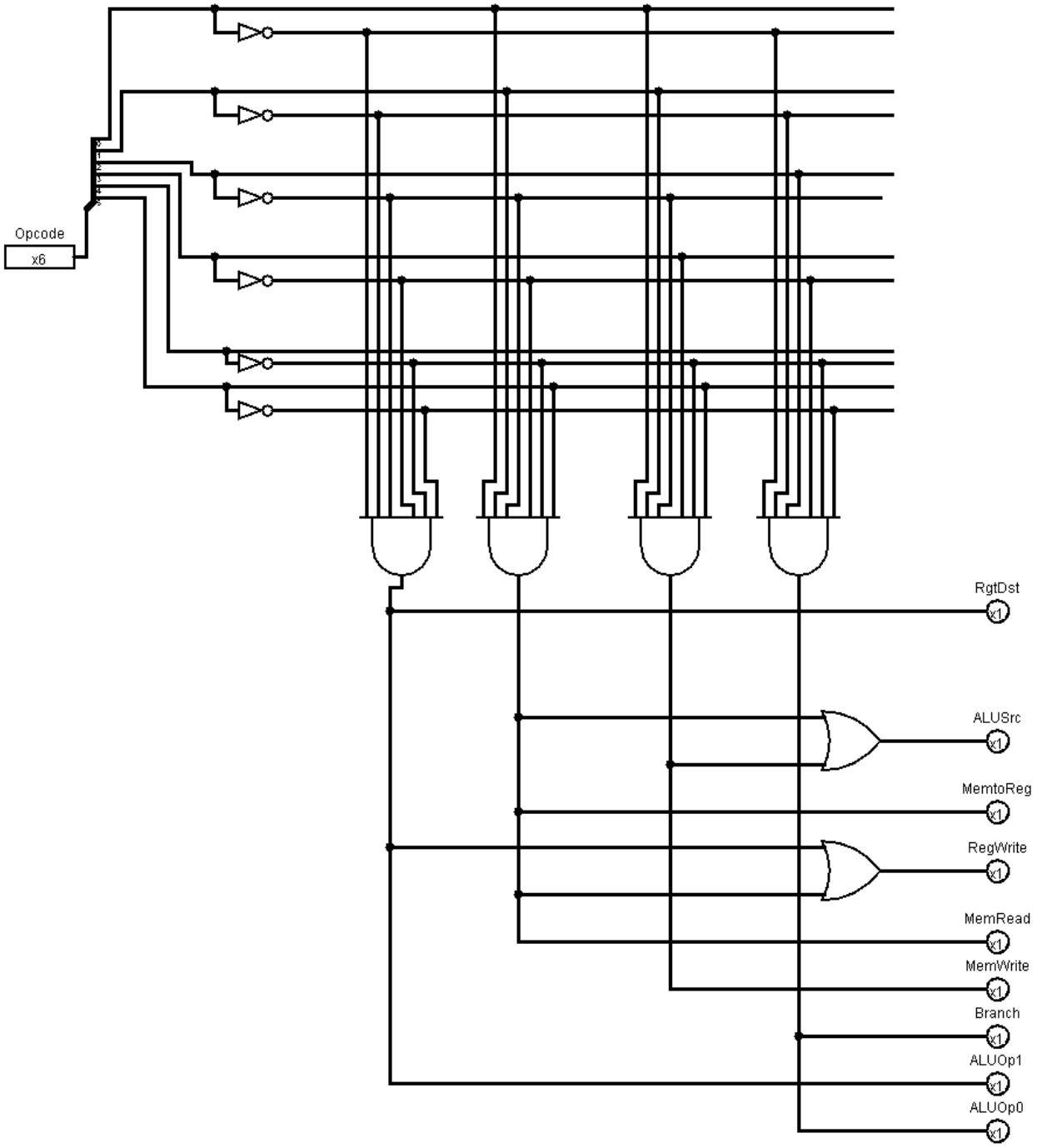
32-bit 6 functions ALU (Part 1):



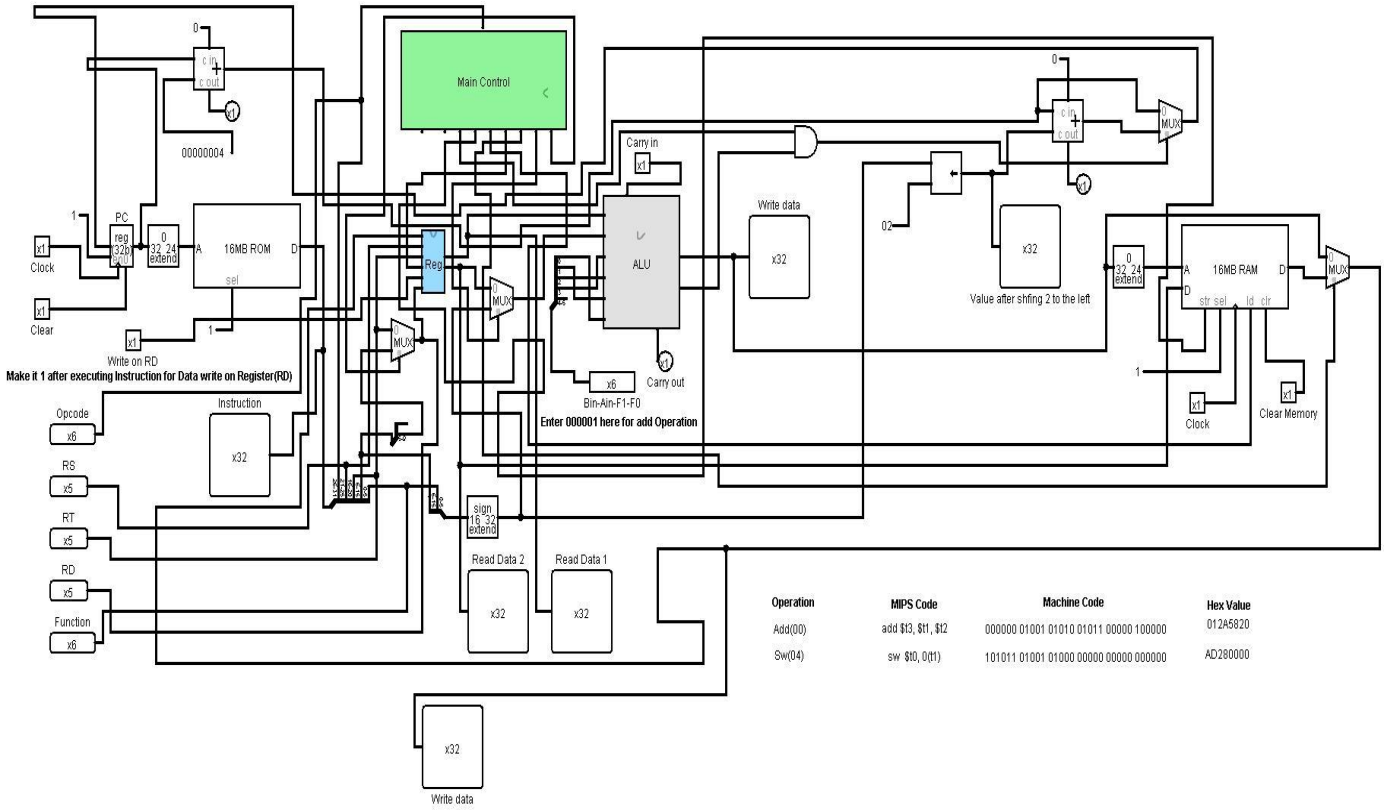
32-bit Register:



Main Control:



Single Cycle Datapath (Part-2):



Discussion:

It should be understood that designing a single cycle Datapath for different types of instructions is a pretty beautiful process Ma-sha-Allah. We used ROM for designing our Datapath design. We also designed 32-bit register and PC for our single cycle Datapath design. We implemented add operation with a R-format instruction and store word with I-format instruction in our Datapath and that worked Alhamdulillah. We also designed a 32-bits ALU from our basic 2-bits ALU using Logisim in part 1 and we used that ALU here where we have basic operations like Add with carry, Subtract with borrow, Nand, Nor, Xor and Multiplication. After seeing the operational table, it should be understood what you need to input in single cycle datapath unit for getting your desired output. Thank you.
