COA Lab Assignment

Assignment:

C-RISC Architecture for a set of defined instructions.

Prepared by:

Methuku Preetham

(16CS01045)

mp19@iitbbs.ac.in

Rahul Kumar

(16CS01010)

rk36@iitbbs.ac.in

Instruction set

Here we present instruction set for our C-RISC Architecture.

Instruction name	Opcode	Instruction Length	meaning
ADD R_I R_J	0000	1	$[R_I] = [R_J] + [R_I]$
SUB R_I R_J	0001	1	$[R_I] = [R_J] - [R_I]$
MUL R_I R_J	0010	1	$[R_I] = [R_J] * [R_I]$
AND R_I R_J	0011	1	$[R_I] = [R_J] \& [R_I]$
0R R_I R_J	0100	1	$[R_I] = [R_J] \mid [R_I]$
XOR R_I R_J	0101	1	$[R_I] = [R_J] \oplus [R_I]$
MOV R_I R_J	0110	1	$[R_I] = [R_J]$
HLT	0111	1	STOP EXECUTION
LOAD R_I M	1000	2	$[R_I] = [\,[\,[PC] +1\,]\,]$
STORE M R_I	1001	2	$[\,[\,[PC]+1\,]\,] = [R_I]$
JUMP	1010	2	$[PC] = [\,[PC] +1]$

[X] = Contents of X (PC, Register, Memory)

Design overview

The design is made with the following considerations:

1. Each instruction takes 4 clock cycles
2. ADD, SUB, MUL, XOR, AND, OR, HLT and MOV are 1-byte instructions
3. JUMP, LOAD and STORE are 2-byte instruction.
4. Emphasis is given towards improving speed and simplicity of the circuit

Instruction execution

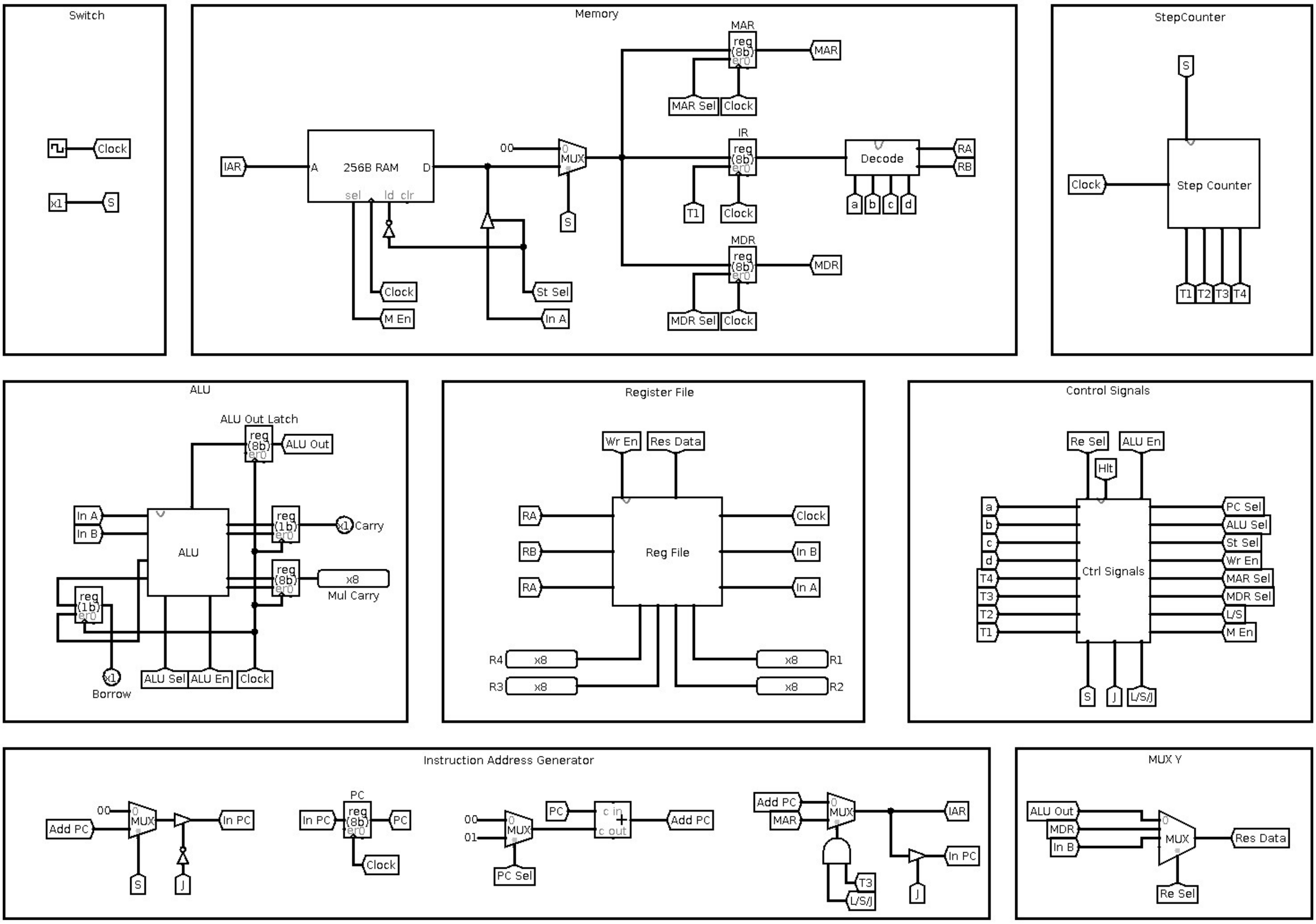
Total number of instructions: 11

Stage	Action
Arithmetic (OP Ri Rj)	
1	Fetch the instruction, Decode, Incrementing PC. ([PC]= [PC]+1)
2	ALU calculation. (ALU Output = [Ri] op [Rj])
3	Latching the ALU output. ([Temp] = ALU Output)
4	Restoring the output to destination register. ([Ri] = [Temp])
MOV Ri Rj	
1	Fetch the instruction, Decode, Incrementing PC. ([PC]= [PC]+1)
2	No Action
3	No Action
4	Restoring the value of Rj to Ri.

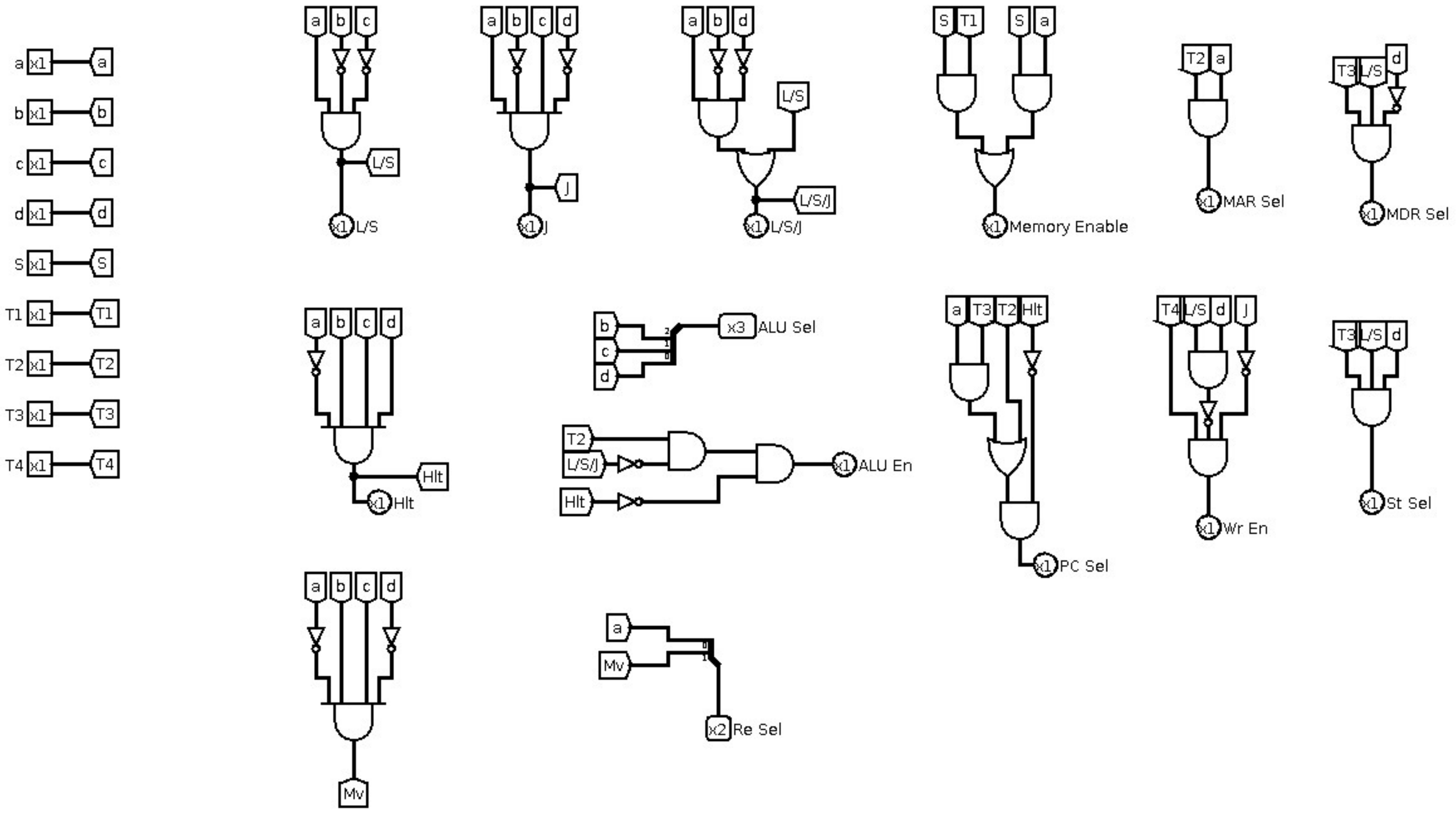
LOAD Ri M	
1	Fetch the instruction, Decode, Incrementing PC. ([PC]= [PC]+1)
2	Fetch the memory address to MAR, Incrementing PC. ([PC]= [PC]+1)
3	Fetch the data from memory address in MAR to MDR.
4	Restoring the value of MDR to Ri.
STORE M Ri	
1	Fetch the instruction, Decode, Incrementing PC. ([PC]= [PC]+1)
2	Fetch the memory address to MAR, Incrementing PC. ([PC]= [PC]+1)
3	Restoring the value of Ri to memory address in MAR.
4	No Action
HLT	
1	Fetch the instruction, Decode, Freezing PC ([PC] = [PC]+0)
2	No Action
3	No Action
4	No Action
JUMP M	
1	Fetch the instruction, Decode, Incrementing PC. ([PC]= [PC]+1)
2	Fetch the memory address to MAR, Incrementing PC. ([PC]= [PC]+1)
3	Update the contents of PC. ([PC] = [MAR])
4	No Action

Circuit diagrams

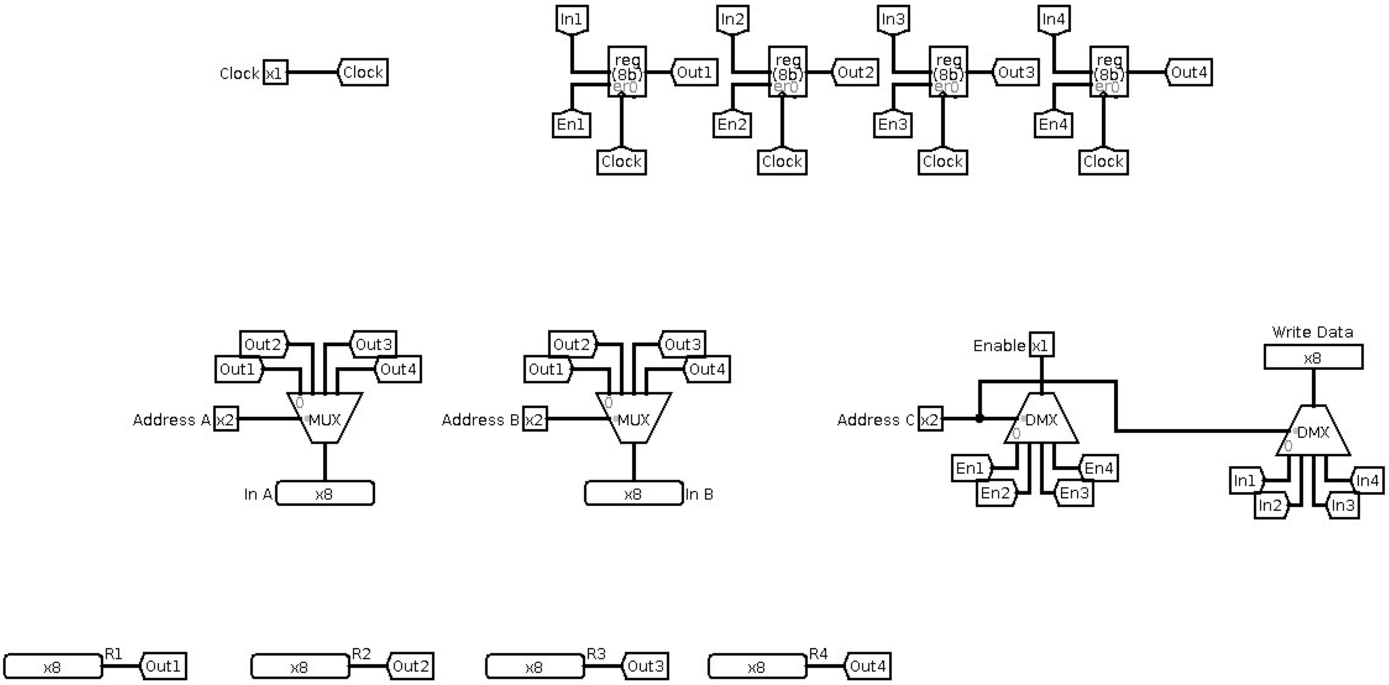
1. Main Circuit



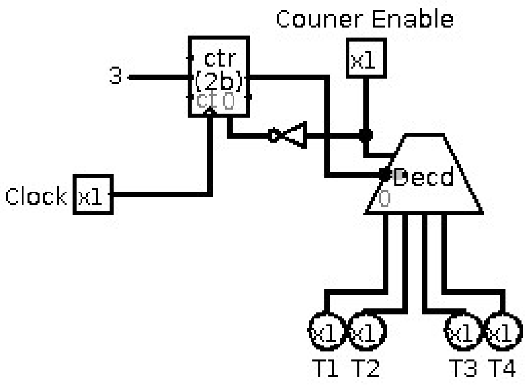
2. Signals


a x1 a
b x1 b
c x1 c
d x1 d
S x1 S
T1 x1 T1
T2 x1 T2
T3 x1 T3
T4 x1 T4
a b c
L/S
x1 L/S
a b c d
J
x1 J
a b d
L/S
L/S/J
x1 L/S/J
S T1 S a
x1 Memory Enable
T2 a
x1 MAR Sel
T3 L/S d
x1 MDR Sel
a b c d
Hlt
x1 Hlt
b c d
x3 ALU Sel
T2
L/S/J
Hlt
x1 ALU En
a T3 T2 Hlt
x1 PC Sel
T4 L/S d J
x1 Wr En
T3 L/S d
x1 St Sel
a b c d
Mv
a
Mv
x2 Re Sel

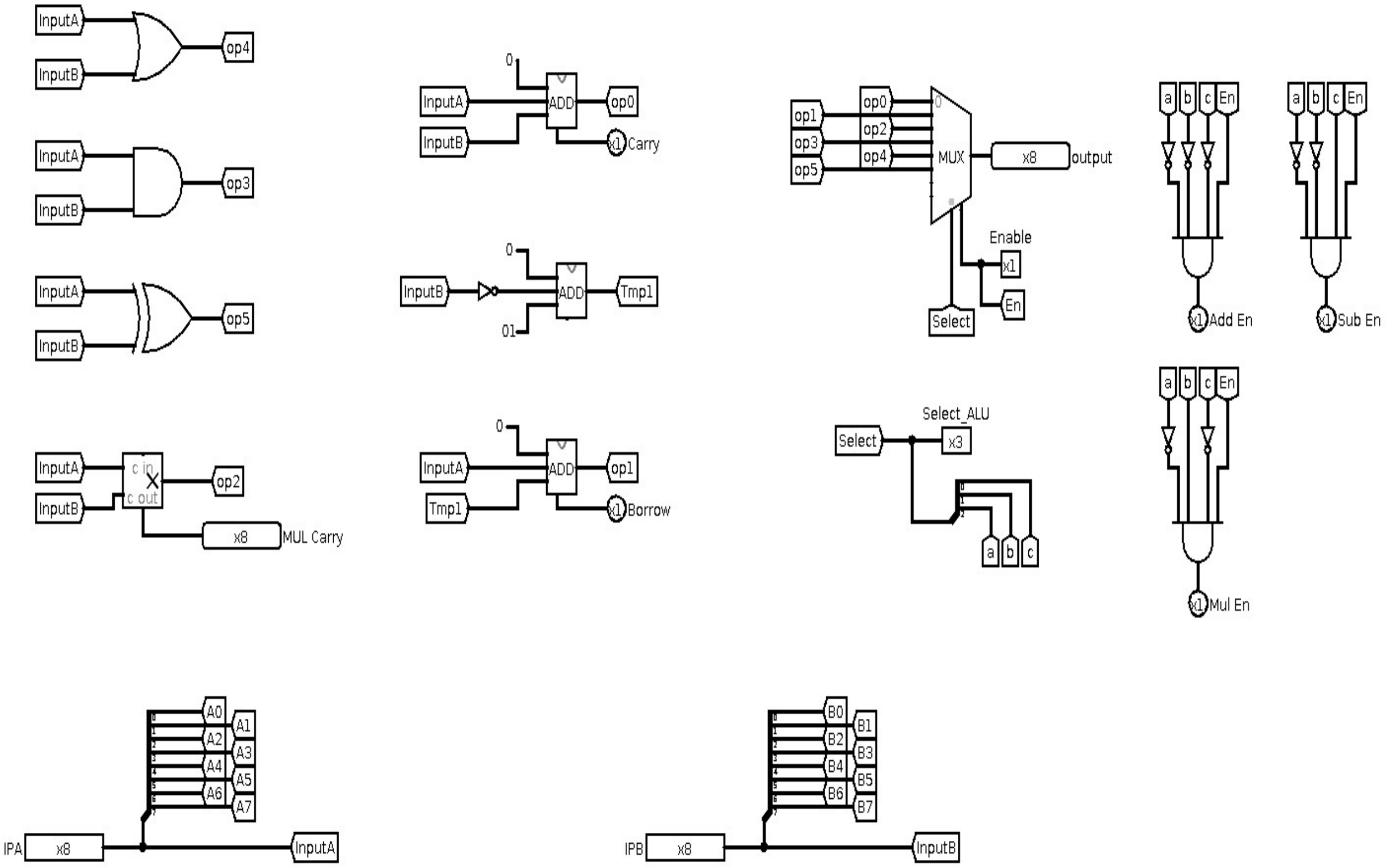
3. Register File



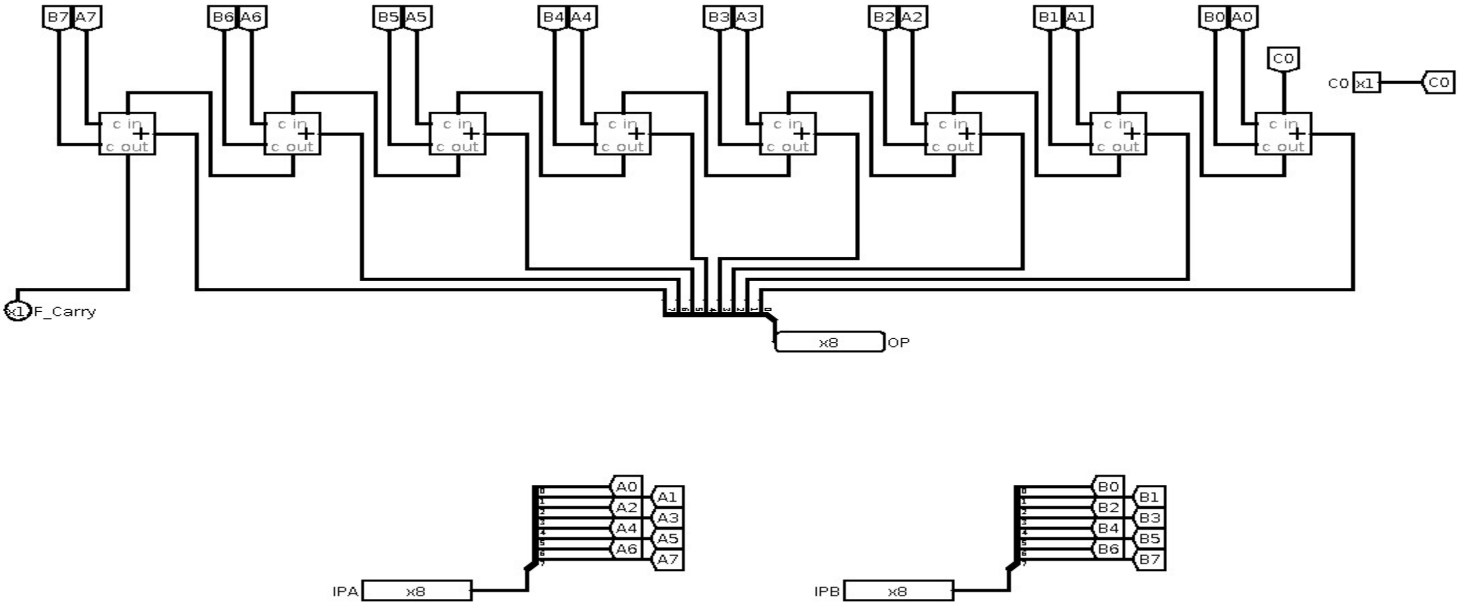
4. Step Counter



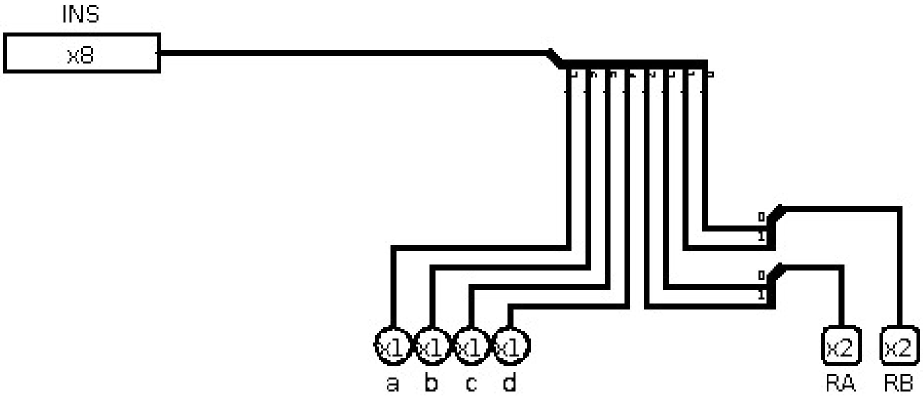
5. ALU


InputA
InputB
op4
op3
op5
c in
c out
op2
x8
MUL Carry
0
ADD
op0
x1
Carry
Tmp1
01
op1
Borrow
op1
op3
op5
op0
op2
op4
MUX
x8
output
Enable
x1
En
Select
Select_ALU
x3
a
b
c
En
Add En
Sub En
Mul En
A0
A1
A2
A3
A4
A5
A6
A7
IPA
B0
B1
B2
B3
B4
B5
B6
B7
IPB

6. ADD



7. Decode


INS
x8
x1 x1 x1 x1
a b c d
x2 x2
RA RB