(i

with done gone, exec doesn't
matter, Assume exec==1

And

Subi

Sub

$op==$ 3'b110

$op==$ 3'b010

$op==$ 3'b111

Read x

Add

$op==$ 3'b011

1

write_x

1   1   1

$op==$ 3'b011

Read y

$op==$ 3'b00)

MV

1

$op==3'b101$
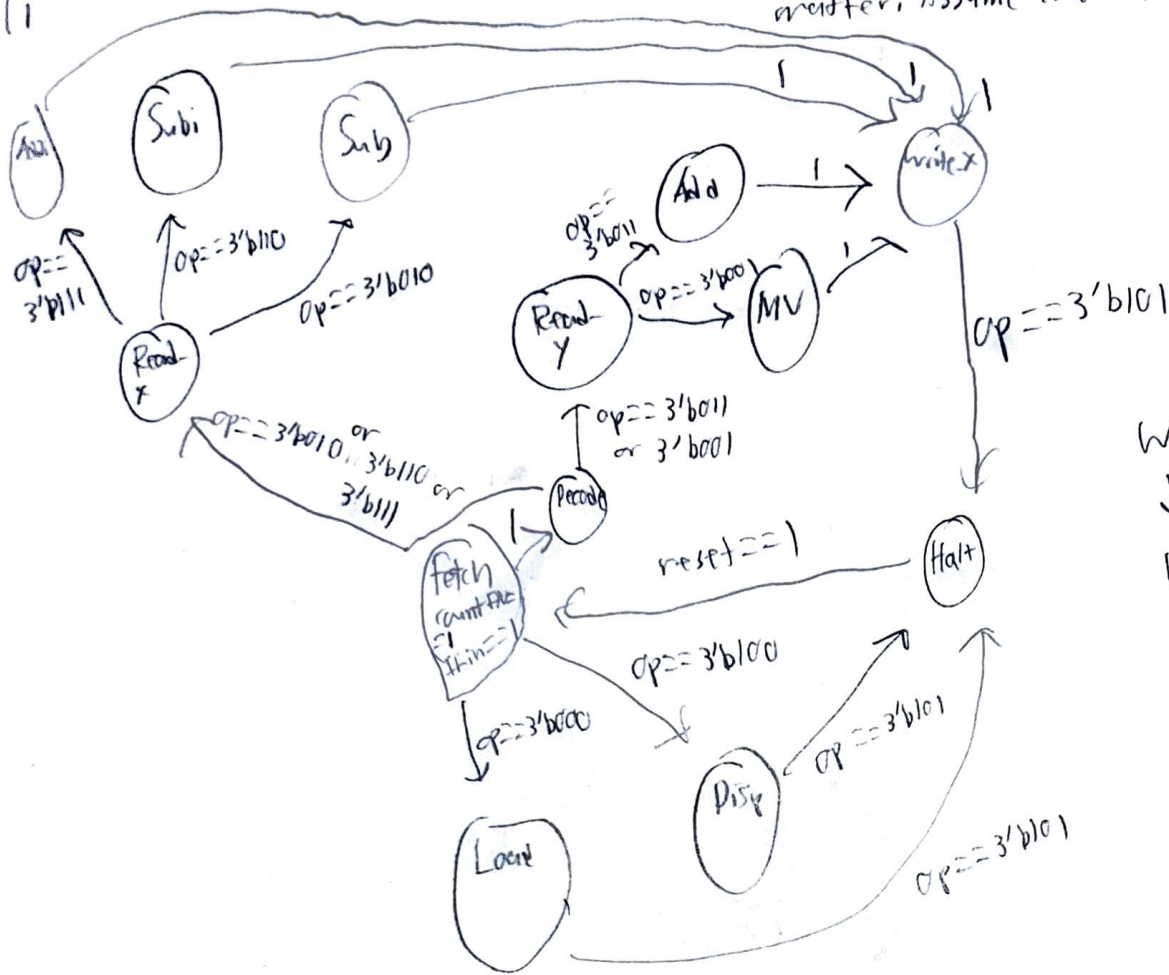
write_x
Jisp
load
all may
go straight
to fetch
(I'm not
sure)

$op==$ 3'b011
or 3'b001

Pdecode

$op==3'b010$ or
3'b010, 3'b110 or
3'b111

fetch
count PAC
=1
ithin==1

1

reset==1

Halt

$op==3'b100$

$op==3'b000$

$op==3'b101$

Disp

$op==3'b101$

Load

$op==3'b101$

i. The new encoding of Halt is 3'b101

Lab 4 script
Part iii.

Addi reg00 reg01 4 //reg00 =4
Subi reg01 reg00 2 //reg01 = 2
Load reg11 2 //reg11 = 2 loads immediate
Mv reg10 reg00 //reg10 = 4
Add reg00 reg00 //reg00=8
Sub reg01 reg01 //reg01 = 0
Disp reg00 //display 8
Halt

Op | reg_x | reg_y | imm | unused
111 | 00 | 01 | 0100 | 00000
110 | 01 | 00 | 0010 | 00000
000 | 11 | 00 | 0010 | 00000
001 | 10 | 00 | 0100 | 00000
011 | 00 | 00 | 0000 | 00000
010 | 01 | 01 | 0000 | 00000
100 | 00 | 00 | 0000 | 00000
101 | 00 | 00 | 0000 | 00000

1110001010000000
1100100001000000
0001100001000000
0011000010000000
0110000000000000
0100101000000000
1000000000000000
1010000000000000

iv. My program adds four to reg00, then subs 2 from reg00 and puts the difference in reg 01.
Next it loads 2 into reg11 and moves reg00 into reg10 (reg10 = 4). It then adds reg00 with reg00
(reg00=8) and subs reg01 from itself (reg01=0). Lastly it displays reg00 and halts.

At the end:
reg00=8
reg01=0
reg10=4
Reg11=2