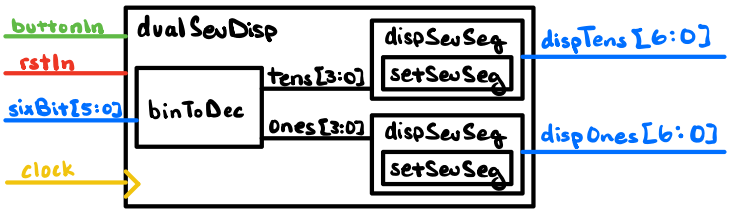
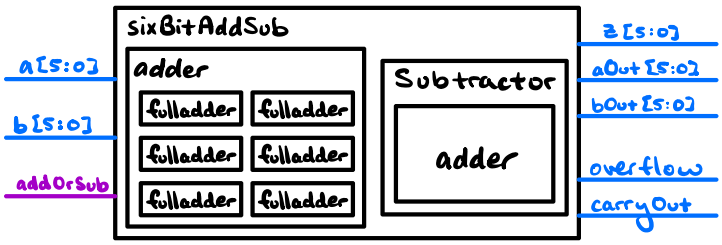
From lab 6

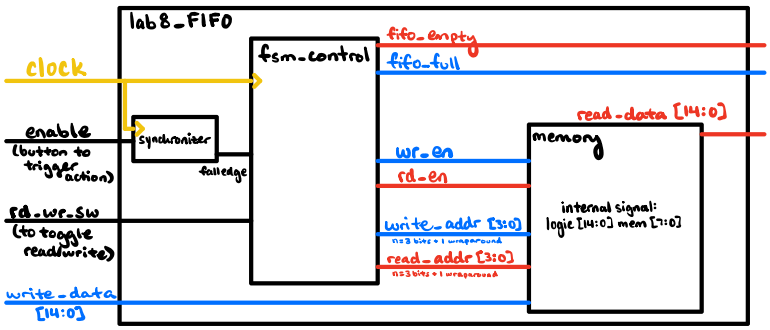
remove button input



From lab 4



From lab 8



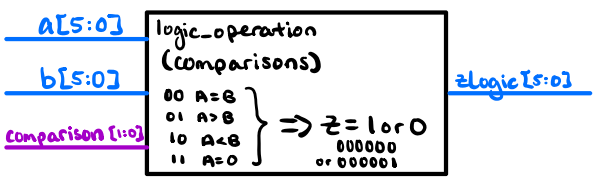
add reset to FIFO!

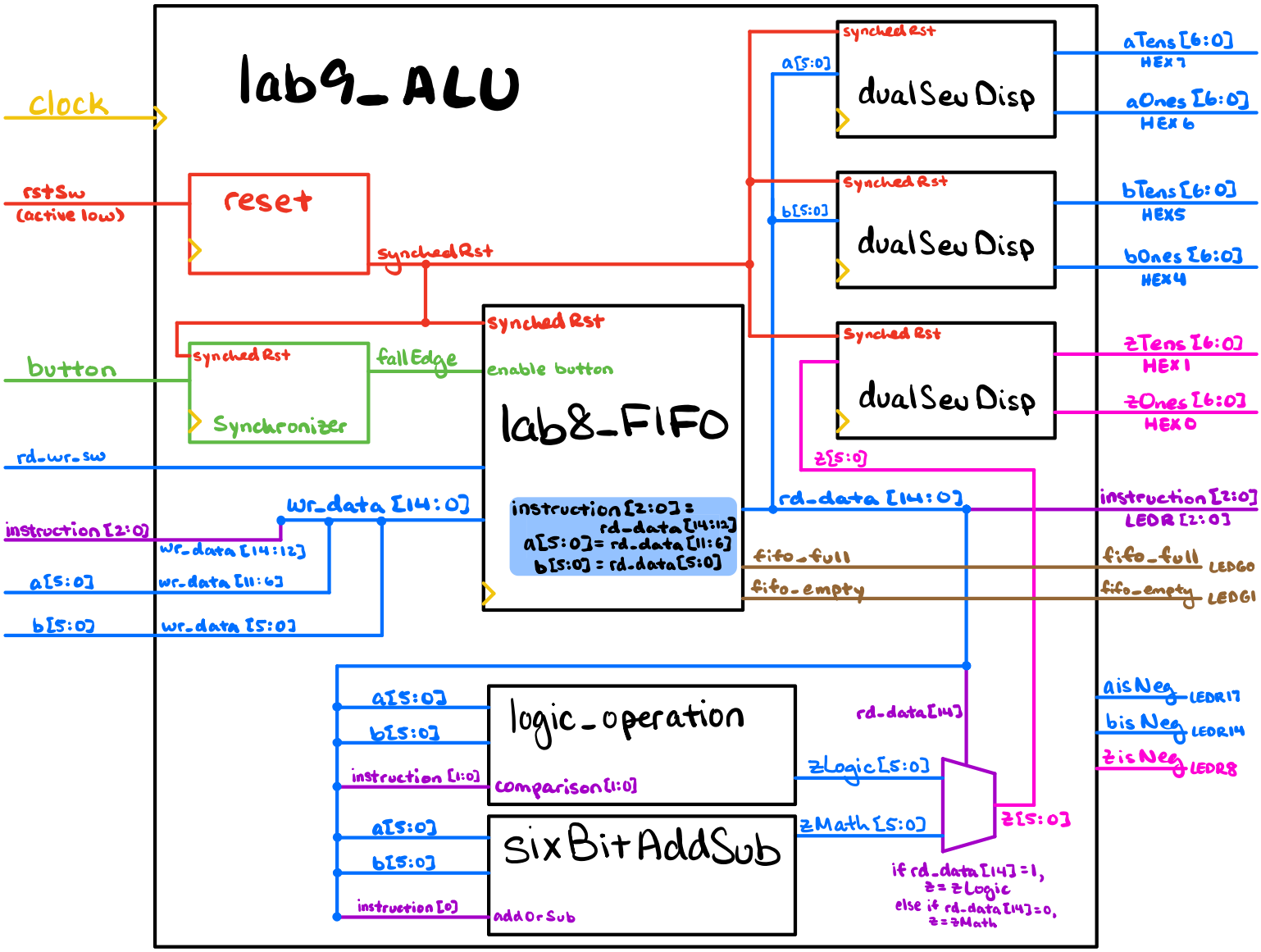
ALL FFs MUST INCORPORATE RESET!

→ reset FSM, addr, & en

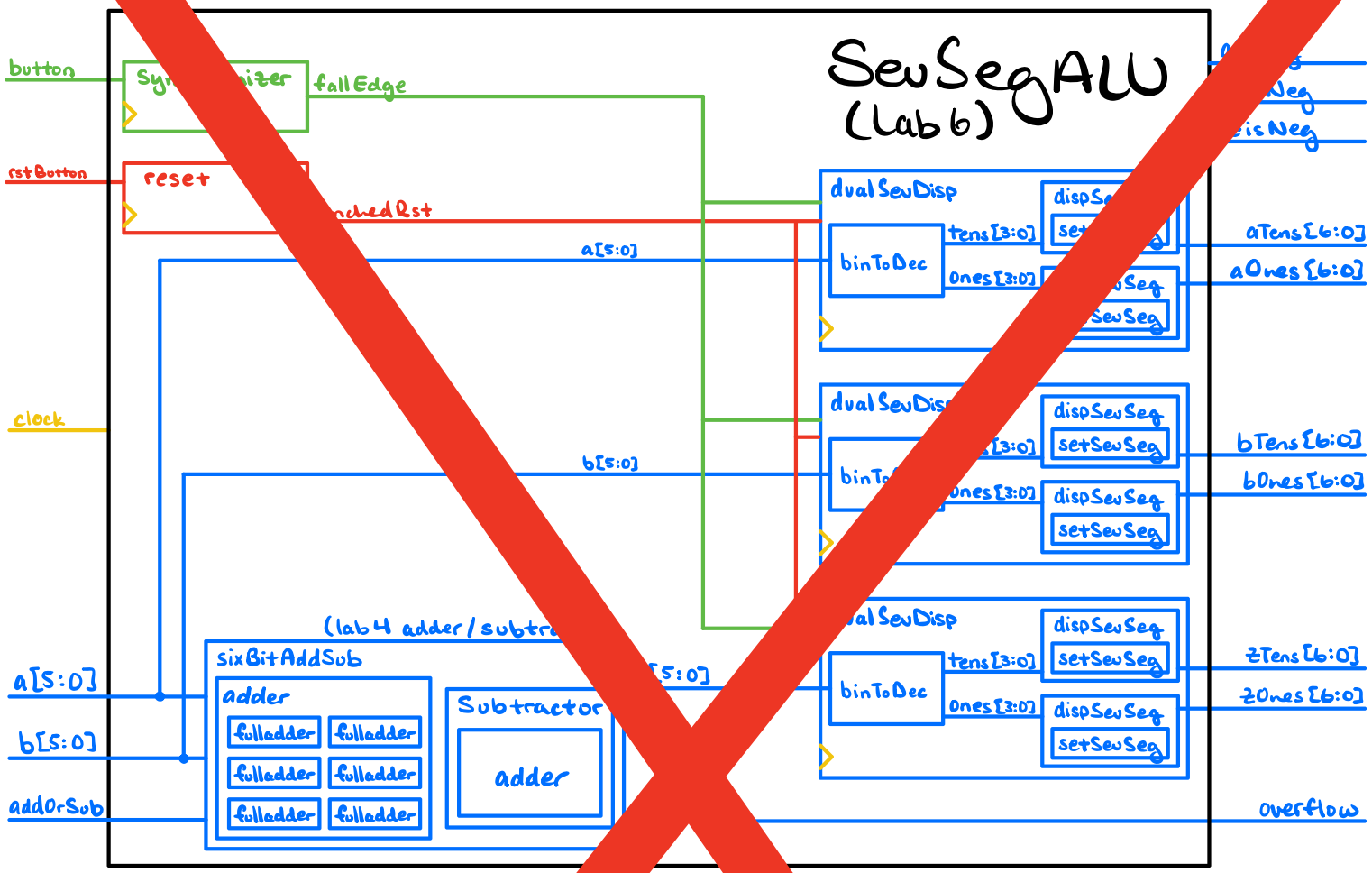
From lab 7

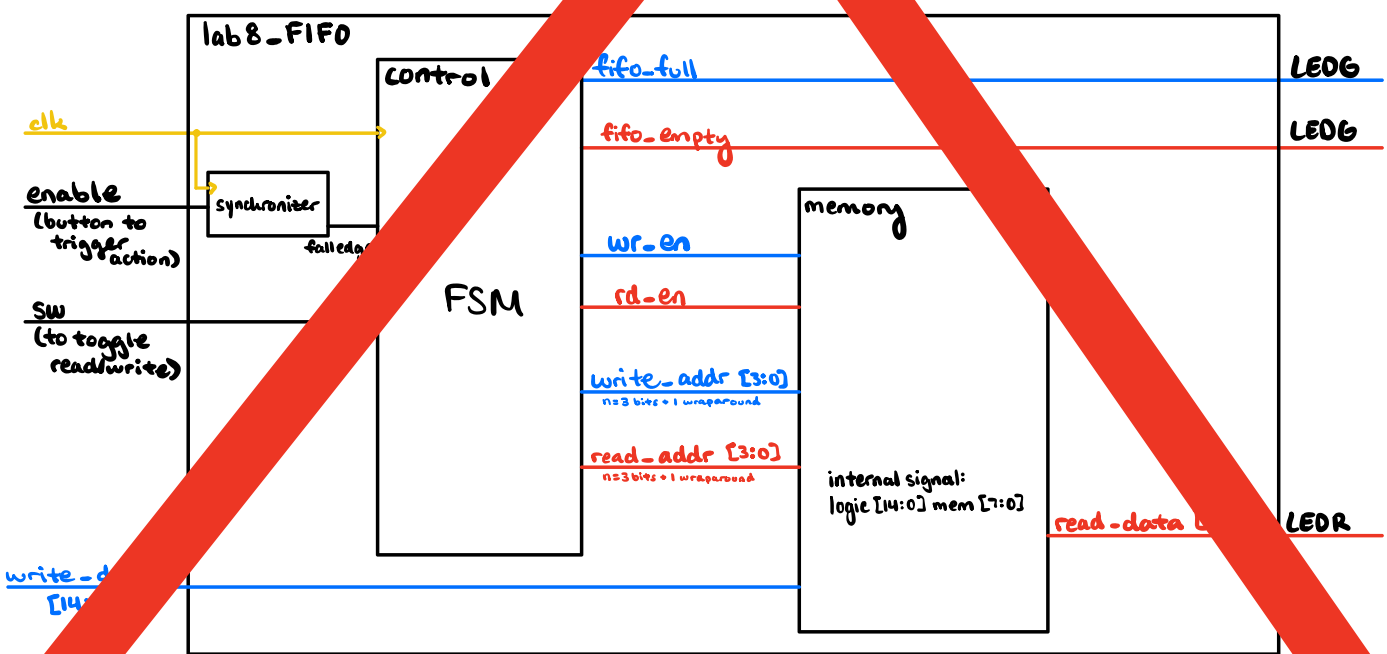
New Module



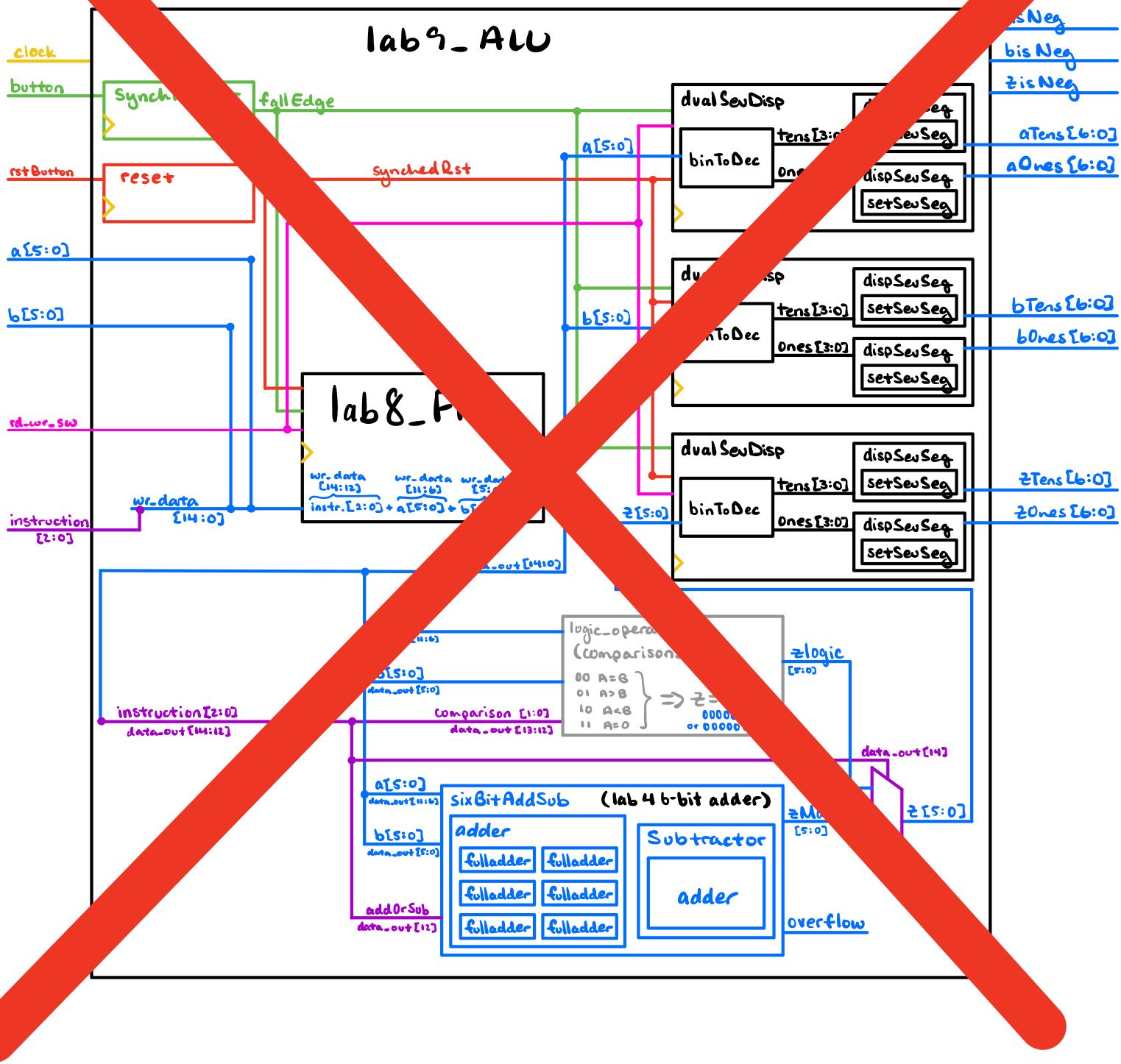

lab9_ALU
clock
rstSw
(active low)
reset
syncedRst
button
syncedRst
Synchronizer
fallEdge
syncedRst
enable button
lab8_FIFO
rd_wr_sw
instruction [2:0]
wr_data [14:0]
wr_data [14:12]
a[5:0]
wr_data [11:6]
b[5:0]
wr_data [5:0]
instruction [2:0] = rd_data [14:12]
a[5:0] = rd_data [11:6]
b[5:0] = rd_data [5:0]
fifo_full
fifo_empty
rd_data [14:0]
syncedRst
a[5:0]
dualSevDisp
aTens [6:0]
HEX7
aOnes [6:0]
HEX6
syncedRst
b[5:0]
dualSevDisp
bTens [6:0]
HEX5
bOnes [6:0]
HEX4
syncedRst
z[5:0]
dualSevDisp
zTens [6:0]
HEX1
zOnes [6:0]
HEX0
instruction [2:0]
LEDR [2:0]
fifo_full
LEDG0
fifo_empty
LEDG1
a[5:0]
b[5:0]
instruction [1:0]
logic_operation
comparison [1:0]
zLogic [5:0]
a[5:0]
b[5:0]
instruction [0]
SixBitAddSub
addOrSub
zMath [5:0]
rd_data[14]
z[5:0]
if rd_data[14] = 1, z = zLogic
else if rd_data[14] = 0, z = zMath
aisNeg
LEDR17
bisNeg
LEDR14
zisNeg
LEDR8

CURRENT PARTS





NEW ALU


lab9_ALU
clock
button
Synch
fallEdge
rstButton
reset
syncedRst
a[5:0]
b[5:0]
sd_wr_sw
instruction [2:0]
wr_data [14:0]
lab8_F
wr_data [14:12]
wr_data [11:6]
instr.[2:0]
a[5:0]
dualSevDisp
binToDec
tens[3:0]
ones[3:0]
dispSevSeg
setSevSeg
aTens[6:0]
aOnes[6:0]
bTens[6:0]
bOnes[6:0]
zTens[6:0]
zOnes[6:0]
z[5:0]
logic_opera
Comparison
00 A=B
01 A>B
10 A<B
11 A=0
zlogic [5:0]
comparison [1:0]
data_out [13:12]
instruction[2:0]
data_out [14:12]
data_out [14]
sixBitAddSub
(lab 4 b-bit adder)
adder
fulladder
Subtractor
addOrSub
data_out [12]
overflow

lab9_ALU
Synchronizer
fallEdge
rstSw
(active low)
syncheadRst
dualSeg Disp
dualSeg Disp
dualSeg Disp
a[5:0]
b[5:0]
aTens [6:0]
aOnes [6:0]
bTens [6:0]
bOnes [6:0]
zTens [6:0]
zOnes [6:0]
z[5:0]
rd_wr_sw
wr_data [14:0]
instruction [2:0]
rd_data [14:0]
fifo_full
f:fo_full
LEDR
logic_operation
sixBitAddSub
rd_data[14]
zMath [5:0]
addOrSub

OG 111011

000101

2sC 000101

111010

revert back 111011

OG 101110

010001

2sC 010010

101101

revert back 101110

The ALU will use the following Instruction Set:

- Add = 000
- Subtract = 001
- Equal = 100
- Greater than = 101
- Less than = 110
- A equal 0 = 111

-15 -20

000 110001 101100 011101 29

001 000101 5

30 24

000 011110 011000 110110 -10

001 000110 6

7 -8

000 000111 111000 111111 -1

001 001111 15

-8 7

000 111000 000111 111111 -1

001 110001 -15