

Here is our truth table to select ALU functions based on our ISA op Code.

ISA opfield instruction bits:	ALU control lines:
00 (add signed numbers could be negative)	00 (add)
11 (subtract)	01 (subtract)
01 load	10 (nothing)
10 (compare/print)	11 (set on equal)

Lab 1 OpCodes Revised:

00: Add instruction adds two registers into a third destination

01: Load instruction, loads address into memory

10: Compare and skip, skip none if skip is 0, skip 1 or 2 if skip is 1 or 2, print the value of first register if skip is 3

11: Subtract instructions subtracts two registers into a third destination.

ex: 00110010
 add

 11001001
 sub

 10011011
 print

 10100101
 set on equal

Logic for ISA to instruction:

add = (bit 7)' AND (bit 6)'

sub = (bit 7) AND (bit 6)

nothing = (bit 7) AND (bit 6)' AND (bit 1) AND (bit 0)

set on equal = (bit 7) AND (bit 6)' AND (bit 1 XOR bit 0)

Logic for ISA -> ALU:

op(1) = (bit 7) XOR (bit 6)

op(0) = (bit 7 AND bit 6') OR (bit 7 AND bit 6)