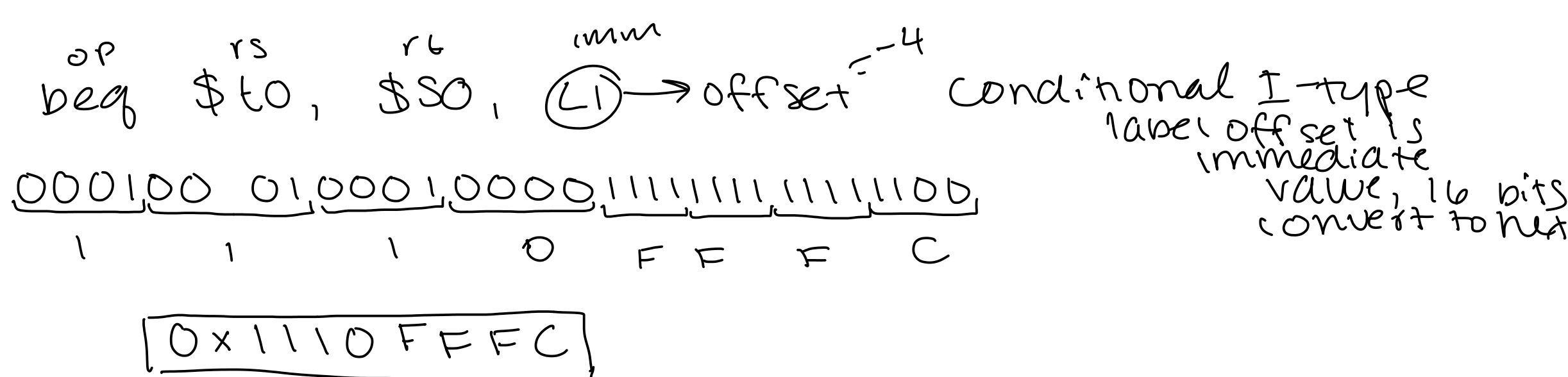
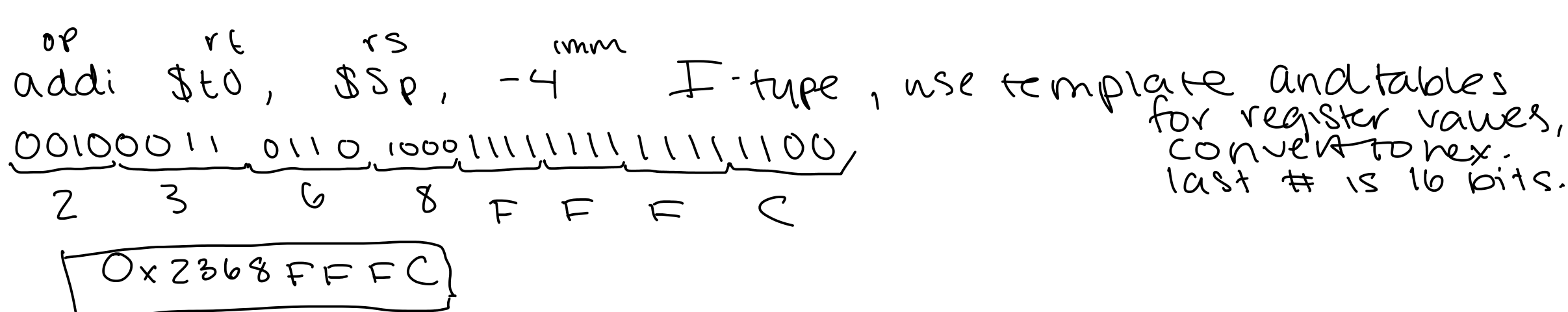
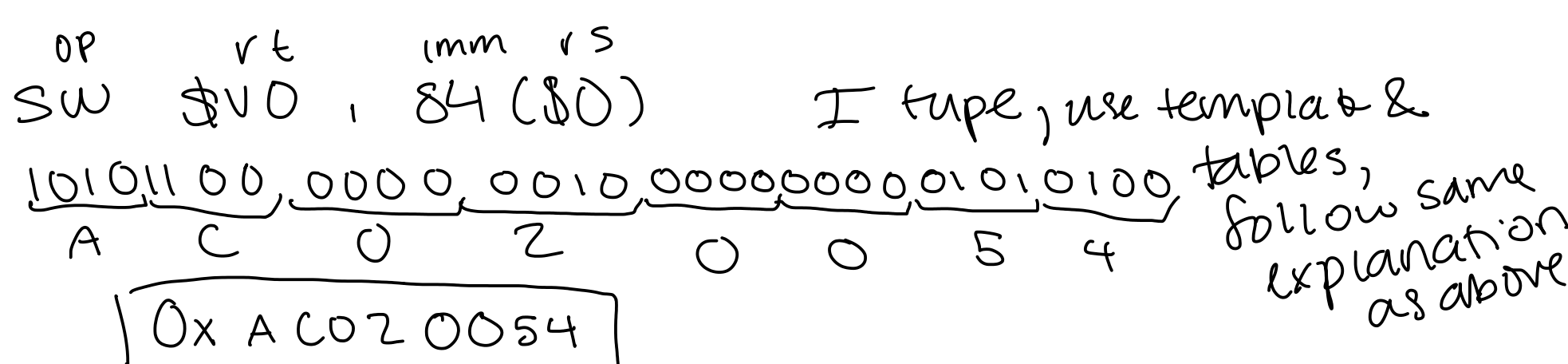
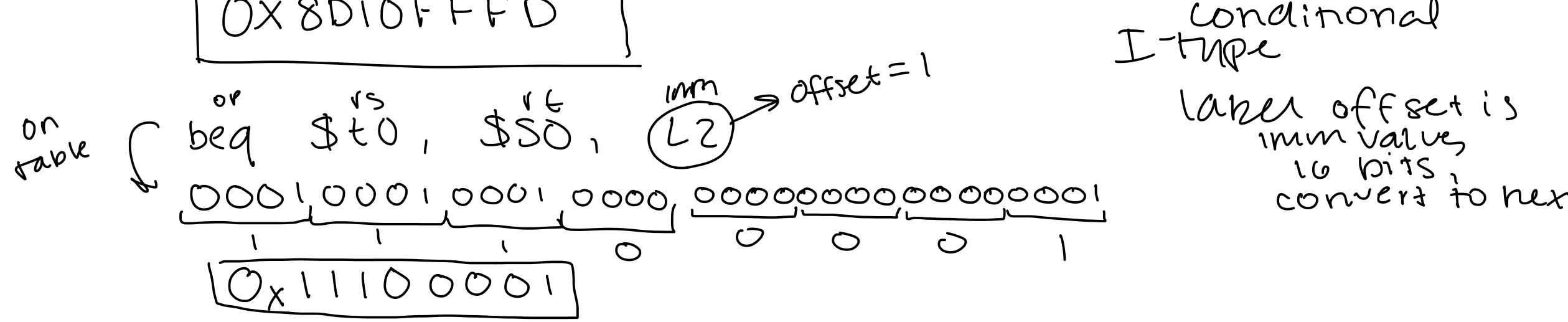
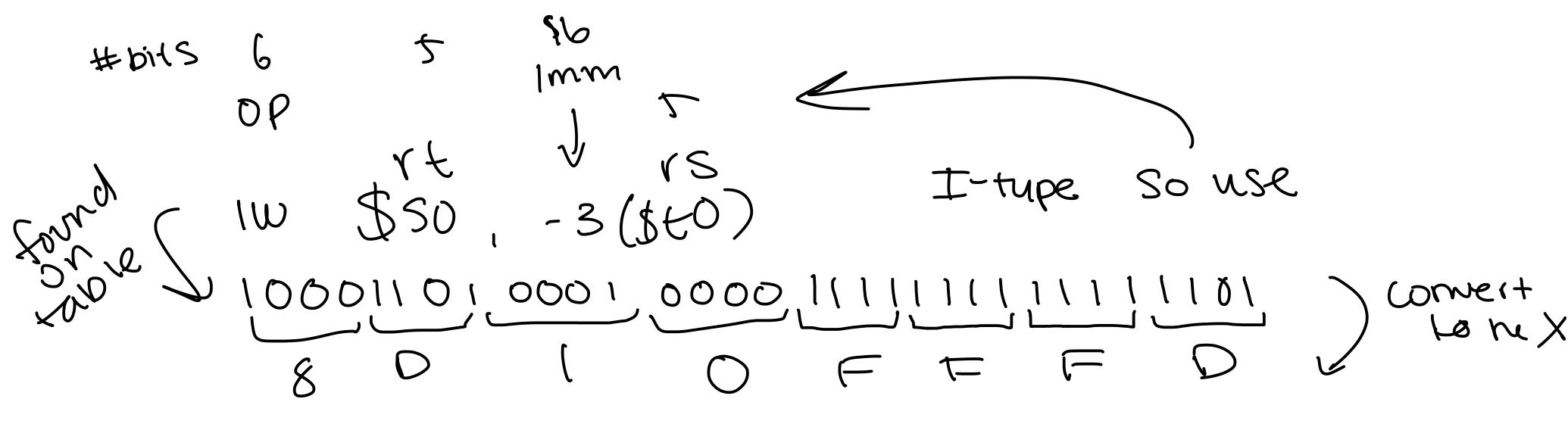
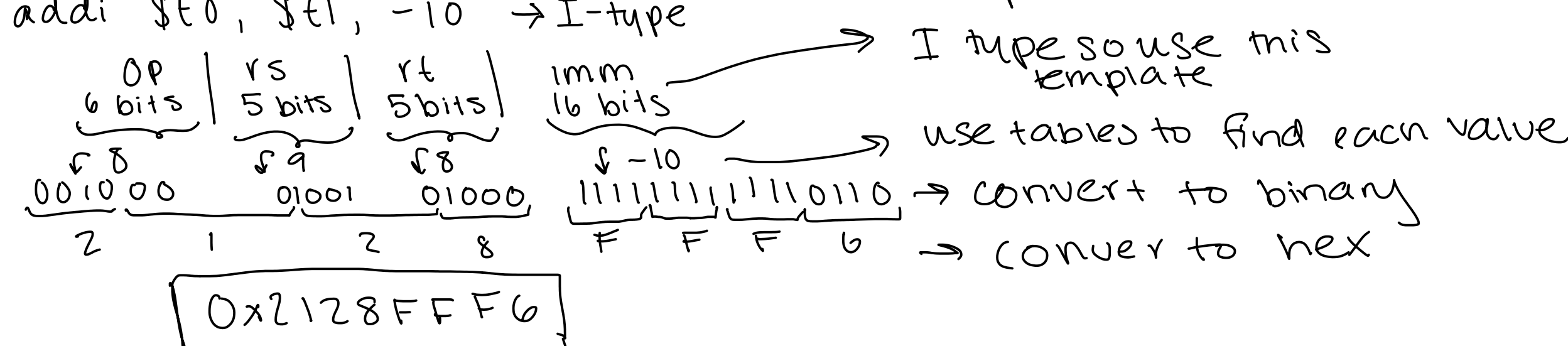


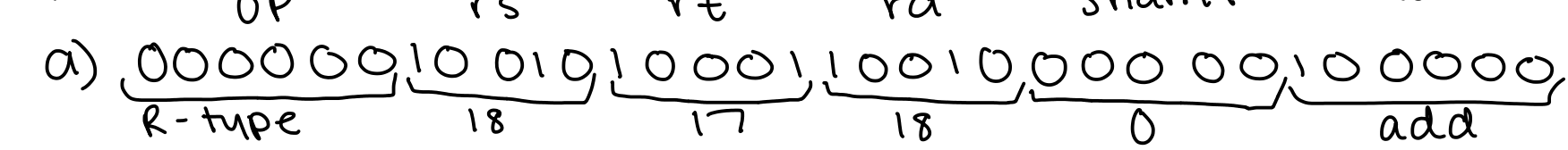
Q1

addi \$t0, \$t1, -10 → I-type

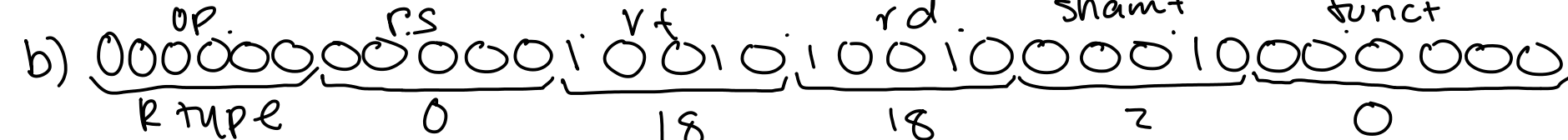
Explanation:



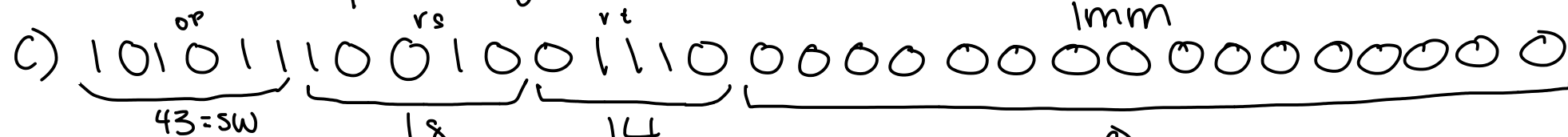
Q2



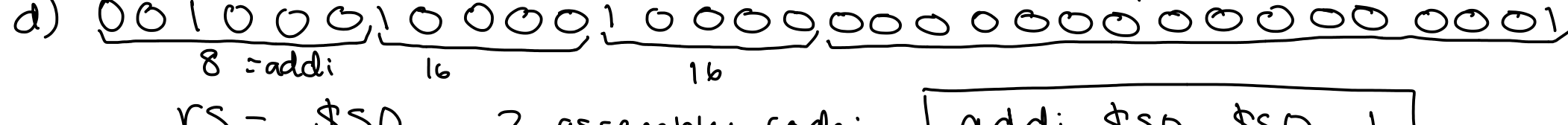
rs = \$s2
rt = \$s1
rd = \$s2
assembly code: add \$s2, \$s2, \$s1



rs = \$0
rt = \$s2
rd = \$s2
assembly code: sll \$s2, \$s2, 2



rs = \$s2
rt = \$t6
assembly code: sw \$t6, 0(\$s2)



rs = \$s0
rt = \$s0
assembly code: addi \$s0, \$s0, 1



rs = \$s0
rt = \$t0
assembly code: bne \$s0, \$t0, -13



rs = \$s2
rt = \$t2
assembly code: lw \$t2, 5(\$s2)

Q3

C code: sum = 0; for(i=0; i<10; i++) { sum = sum + L[i]; }

MIPS: # \$s1 = i, \$s2 = base address array L = 0x55552222

lui \$s2, 0x5555 // load upper immediate
ori \$s2, \$s2, 0x2222
addi \$s1, \$0, 0
addi \$t2, \$0, 10

target: slt \$t0, \$s1, \$t2
beq \$t0, \$0, done // t3 = i * 4 (byte offset)
sll \$t3, \$s1, 2 // address of L[i]
add \$t3, \$t3, \$s0
lw \$t1, 0(\$t3)

Q4

\$t0 = c, \$s0 = x, \$s1 = y, \$s2 = z

Switch (c) {
case 1: x = y + z;
case 2: x = y - z;
default: x++;
}

beq \$t0, \$s0, case0 // case 0
lw \$t1, 1 // load 1 to compare
beq \$t0, \$t1, case1 // case 1
add \$s0, \$s0, 1 // default +
case0: add \$s0, \$s1, \$s2
case1: sub \$s0, \$s1, \$s2

Q5

\$t0 = random 32 bit int. \$s0 = store count

lui \$t0, x
ori \$t0, \$t0, y
addi \$s0, \$0, 0

label 1:
beq \$t0, 0, label 2
and \$s1, \$t0, 1
beq \$s1, 0, label 2
add \$s0, \$s0, 1

label 2:
srl \$t0, \$t0, 1
j label 1

end: ...

Q6

a) range is [0, 8) or (0 ≤ \$s1 ≤ 7)

every shift left multiplies by 2

if (\$s1 < 0) \$t1 = 1
else \$t1 = 0

so if \$t1 != 0 → done

if \$t1 = 0
if (\$s1 < 8) \$t2 = 1
else \$t2 = 0

so if \$t2 = 0 → done
else add 1 + \$s1

done...

b) because it's unsigned so there are no negative numbers. no negative # means must be above 0.

line 1: if (\$s1 < 8) \$t1 = 1
else \$t1 = 0

line 2: if \$t1 = 0 → done
else, continue

for example: if \$s1 = 3
mips
slliu \$t1, \$s1, 8
beq \$t1, \$0, done
addi \$s1, \$s1, 1
if (4 < 8) \$t1 = 1
& repeat until \$s1 = 8, then \$t1 = 0 → done