Q1.

(a)
```
li $s0,4000
lw $t1,44($s0)
add $t1, $t1,$t0
sw  $t1, 40($s0)
```

(b) 80000 is too large to represent with 16 bits (the length of the immediate field in a sw instruction)

(1c) 2 instructions are sufficient

```
addui $t1, $t0, 65535
sw $a0, 14465($t1)
```

Q2. The code detrmies the most frequent word in an array and returns it $v1 and its multiplicity in $v0

Q3. Solution:

(a)

| A | B | Out0 | Out1 | Out2 | Out3 |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

This circuit is a 2:4 decoder (a decoder is sufficient)

(b)

| a | b | Cin | Cout | Sum |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

(c)

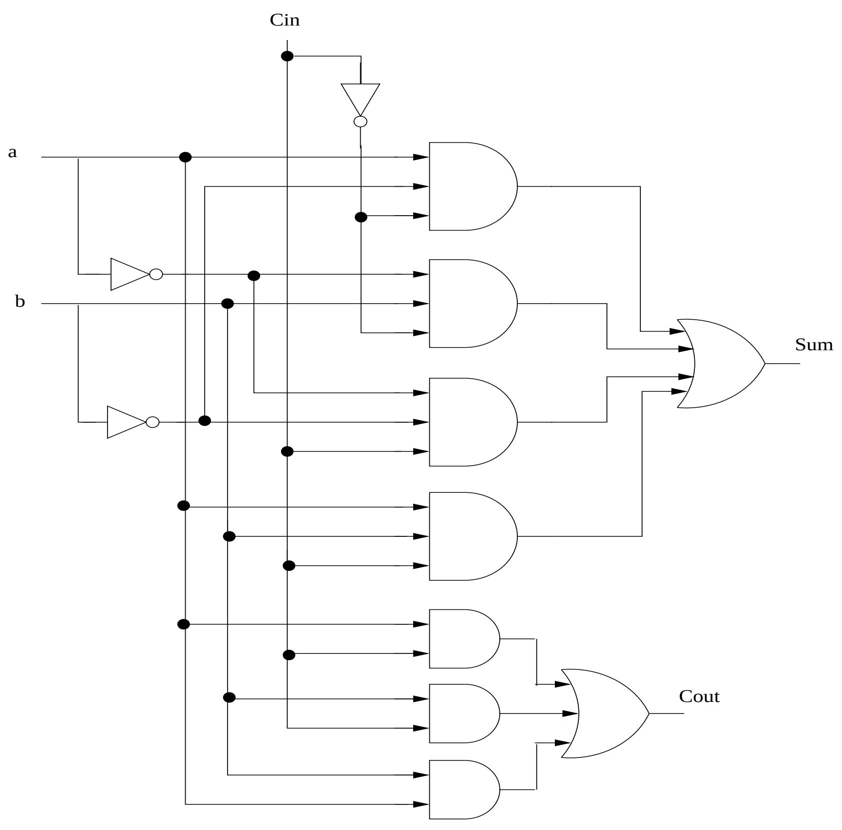
$\text{Sum} = (\overline{a} \,.\, \overline{b} \,.\, \text{Cin}) + (\overline{a} \,.\, b \,.\, \overline{\text{Cin}}) + (a \,.\, \overline{b} \,.\, \overline{\text{Cin}}) + (a \,.\, b \,.\, \text{Cin})$

$\text{Cout} = (\overline{a} \,.\, b \,.\, \text{Cin}) + (a \,.\, \overline{b} \,.\, \text{Cin}) + (a \,.\, b \,.\, \overline{\text{Cin}}) + (a \,.\, b \,.\, \text{Cin})$

which can be simplified into the following

$\text{Cout} = (b \,.\, \text{Cin}) + (a \,.\, \text{Cin}) + (a \,.\, b)$

(d)

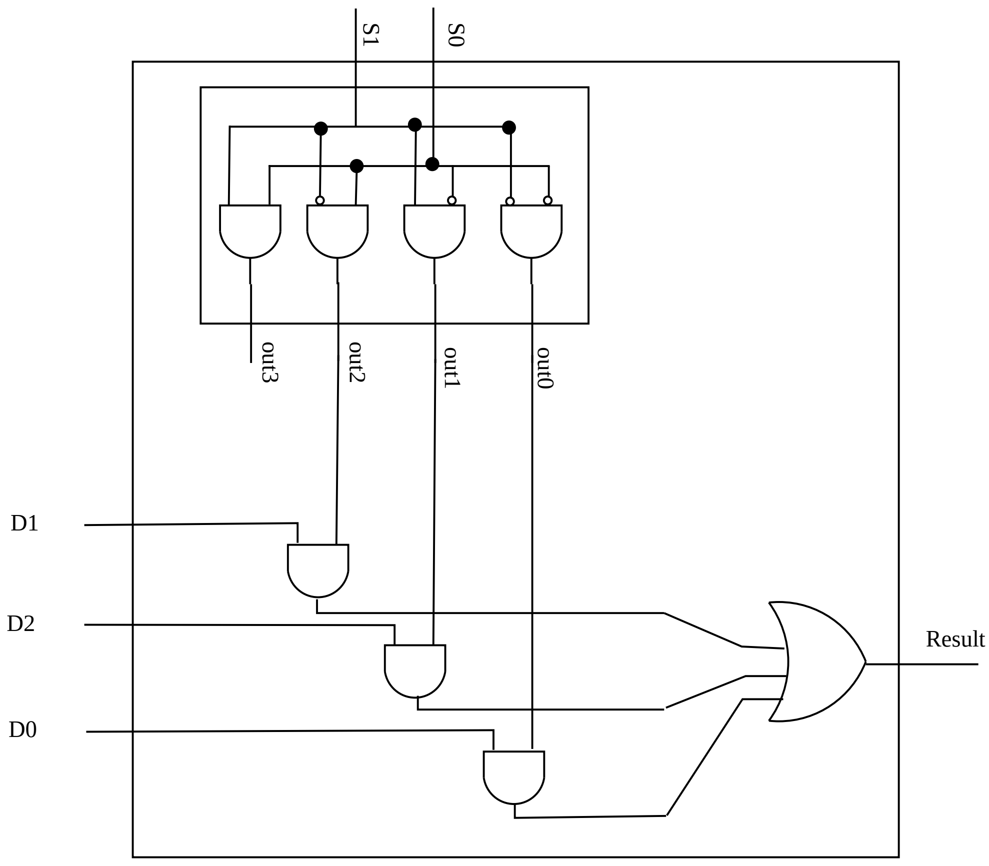


(e)

$$\text{Result} = \overline{S1} \,.\, \overline{S0} \,.\, D0 + \overline{S1} \,.\, S0 \,.\, D1 + S1 \,.\, \overline{S0} \,.\, D2$$

| <u>S0</u> | <u>S1</u> | <u>D0</u> | <u>D1</u> | <u>D2</u> | <u>Result</u> |
|---|---|---|---|---|---|
| 0 | 0 | 1 | X | X | 1 |
| 1 | 0 | X | 1 | X | 1 |
| 0 | 1 | X | X | 1 | 1 |
| 1 | 1 | X | X | X | X |

(f)



Q4. Solution:

(a) CPI for P = 2*0.4 + 3*0.3 + 4*0.2 + 5*0.1
= 3.0

CPI for P' = 2*0.4 + 2*0.3 + 3*0.2 + 4*0.1
= 2.4

(b) Therefore P' is faster than P by
the ratio 50/33.3 = 1.5 or about 50%

(c) CPI of P with new compiler
= 2*0.5 + 3*0.35 + 4*0.1 + 5*0.05 = 2.7

(d) with the new compiler, P is faster by

37.04/33.3 = 1.112 or about 11%

Q5

```
addu $t2, $t3, $t4
sltu $t2, $t2, $t3
```

Q6.

(a) In both cases

step1: the address of the following instruction (i.e., 00004116 respect. 00004212) is stored in the return address register $ra ($31 is also considered correct)

step 2: The PC is loaded with the address of the instruction labelled “term”, respectively “fact” (i.e., 00004200 respect. 00004440).

NB: if answered by: the execution jumps to ... it should also be considered correct

(b) $ra = 00004116

(c) $ra = 00004212

(d) $ra = 00004212

(e) $pc = 00004212

Yes there is a problem. The program has an infinite loop in procedure term. This occured because the first return address has been wiped by the second return address (the action of the first jal in question a.step1 has been erased by the action of the second jal).

(f) register $ra needs to be pushed (saved) on the stack at the beginning of the procedures and poped (restored) just before the instruction jr $ra.

NB: if they do not mention the stack yet mention “save” and “restore” it is also correct.