

Assignment-II

Model.

Part-II : Solution

①
• data

strA: .ascii " After Swapping: x="

strB: .ascii " y="

newline: .ascii "\n"

x: .word 10

y: .word 5

• text

• global main

main:

la \$s0, x

la \$s1, y

lw \$s2, 0(\$s0)

lw \$s3, 0(\$s1)

add \$s2, \$s2, \$s3

sub \$s3, \$s2, \$s3

sub \$s2, \$s2, \$s3

Printing:

li \$v0, 4

la \$a0, strA

syscall

li \$v0, 1

move \$a0, \$t1

syscall

li \$v0, 4

la \$a0, strB

syscall

li \$v0, 1

move \$a0, \$t2

syscall

li \$v0, 4

la \$a0, newline

syscall

li \$v0, 10

syscall

② (a) Word Address

0x10010008

value

78452040

0x0040000c

8 fbe 0000

0x00400000

0c 100007

(b) (i) All of the labels that linker needs to know about: push and pop and main. (for programs where the loader creates code that jump to main). We do not need to store labels from the data segment.

(ii) The 'jal' instructions generate absolute addresses and must have relocation information stored to permit placement of the program starting elsewhere in memory.

(c)
 let 1 0x00400018
 push 0x0040001c

(d) 0x0c100010

Assuming you have memory

Total energy consumed

= 450mJ

total instr executed = 31

∴ power = $\frac{450 \text{ mJ}}{31} = 14.516 \text{ W}$

∴ each instr takes $\frac{1}{31} \text{ sec}$

③

• data

L1: .word 0x44, 22, 33, 55

• text

• global main

main: la \$t0, L1

li \$t1, 4

add \$t2, \$zero, \$zero

loop: lw \$t3, 0(\$t0)

add \$t2, \$t2, \$t3

addi \$t0, \$t0, 4

addi \$t1, \$t1, -1

bne \$t1, \$zero, loop

bgt \$t2, \$0, then

move \$s0, \$t2

j exit

then: move \$s1, \$t2

exit: li \$v0, 10
 syscall

→ la $\begin{matrix} \text{lui} \rightarrow 10 \\ \text{ori} \rightarrow 10 \end{matrix}$
 → li $\begin{matrix} \text{lui} \rightarrow 10 \\ \text{ori} \rightarrow 10 \end{matrix}$

→ 10

→ 40

→ 10

→ 10

→ 10

→ 10

→ 10

→ 10

→ 10

→ 20nJ

→ 20nJ

→ 20nJ

40nJ

320nJ

90nJ

20nJ