

# Akash Palayil Homework - 2

Due : 9/26/2022

Q.1

Ex. 2.7

The given data is Oxabcdef12 and starting address of memory is 0. The word size is 4 bytes.

Data arrangement using little-endian machine in the memory :

Data	Address	Binary data
12	0	0001 0010
ef	1	1110 1111
cd	2	1100 1101
ab	3	1010 1011

Data arrangement using big-endian machine in the memory :

Data	Address	Binary data
ab	0	1010 1011
cd	1	1100 1101
ef	2	1110 1111
12	3	0001 0010

Q.2 Ex 2.14

Using MIPS assembly set:

0000 00 10 0000 10000 10000 00000 10 0000

Last 6 bits corresponds to the opcode of ADD instruction.

Bit fields of ADD are:

Bits 25 to 21: Rs  $\rightarrow$  (10000)<sub>2</sub> : \$0

Bits 20 to 16: Rt  $\rightarrow$  (10000)<sub>2</sub> : \$0

Bits 15 to 11: Rd  $\rightarrow$  (10000)<sub>2</sub> : \$0

Bits 10 to 6 : 0

Thus, the instruction is ADD \$=0, \$=0, \$=0.

Q.3 Ex 2.15

sw \$t1, 32(\$t2)

Comparing the instruction with sw RT, offset(base)

Bits 31 to 26 : sw  $\rightarrow$  (101011)<sub>2</sub>

Bits 25 to 21 : \$t2  $\rightarrow$  (01010)<sub>2</sub>

Bits 20 to 16 : \$t1  $\rightarrow$  (01001)<sub>2</sub>

Bits 15 to 0 : offset  $\rightarrow$  32  $\rightarrow$  (0000 0000 00100000)<sub>2</sub>

Thus, the binary instruction is:

1010 1101 0100 1001 0000 0000 0010 0000  
A D 4 9 0 0 2 0

$\therefore$  The hexadecimal representation is: AD490020

#### Q.4 Ex 2.19

\$t0 = AAAA AAAA , \$t1 = 12345678

Ex 2.19.1

sll \$t2, \$t0, 4

or \$t2, \$t2, \$t1

Shifting 4 bits to left: t2 = 1010 1010 1010 1010 1010 1010 1010 0000

or = 0001 0010 0011 0100 0101 0110 0111 1000

t2 = 1011 1010 1011 1110 1111 1110 1111 1000  
B A B E F E F 8

Ex 2.19.2

sll \$t2, \$t0, 4

andi \$t2, \$t2, -1 [ -1 is represented by 2's compliment.  $-1 = 1111$  ]

Shifting 4 bits to left: t2 = 1010 1010 1010 1010 1010 1010 1010 0000

andi = 1111 1111 1111 1111 1111 1111 1111 1111

t2 = 1010 1010 1010 1010 1010 1010 1010 0000  
A A A A A A A 0

Ex 2.19.3

srl \$t2, \$t0, 3

andi \$t2, \$t2, 0x FF EF

Shifting 3 bits to right: t2 = 0001 0101 0101 0101 0101 0101 0101 0101

andi = 0000 0000 0000 0000 1111 1111 1110 1111

t2 = 0000 0000 0000 0000 0101 0101 0100 0100  
0 0 0 0 5 5 4 5

### Q.5 Ex 2.23

slt \$t2,\$0,\$t0 # \$t2 is set to 1 as 0 is less than 0x00101000  
bne \$t2,\$0, ELSE # the values are unequal ( $0 \neq 1$ ), so ELSE is executed.  
j DONE # Jump to DONE  
ELSE: addi \$t2,\$t2,2 # Adds the value of \$t2 and 2, storing it at \$t2.  
## \$t2 = 2 + 1 = 3

DONE:

### Q.6 Ex 2.27

The MIPS code is:

add \$t0,\$0,\$t0 # i=0

Loop1:

slt \$t2,\$t0,\$s0 # if ( $i < a$ ) set \$t2=1

beq \$t2,\$0, Exit # if (\$t2=0) go to Exit

addi \$t1,\$0,0 # j=0

Loop2:

slt \$t2,\$t1,\$s1 # if ( $j < b$ ) set \$t2=1

beq \$t2,\$0, Loop1 # if (\$t2=0) go to Loop1

add \$t2,\$t0,\$t1 # \$t2 = (i+j)

sll \$t3,\$t1,4 # \$t3 = ( $j \ll 4$ )

add \$t3,\$t3,\$s2 # \$t3 = address of D[ $4^*j$ ]

sw \$t2,0(\$t3) # D[ $4^*j$ ] = (i+j)

addi \$t0,\$t0,1 # i = i+1

j Loop2 # jump to Loop2

Exit:

### Q.7 Ex 2.39

The MIPS instruction is: lui \$t0, 0x2001 with the value  
 $\#t0 = 0010\ 0000\ 0000\ 0001\ 0000\ 0000\ 0000\ 0000$   
ori \$t0, 0x4924 with the value:

# \$t0 = 0100 1001 0010 0100 0000 0000 0000 0000  
add \$t1, \$t0, \$0 the result is stored in the register \$t1 with the  
value : 0010 0000 0000 0001 0100 1001 0010 0100

### Q.8 Ex 2.42

Yes, we can use a single branch instruction, since the  
branch address range from :

$$0 \times 1 F F F F 000 + 4 + 0 \times 1 F F F C = 0 \times 2 0 0 1 F 000$$

to

$$0 \times 1 F F F F 000 + 4 - 0 \times 2 0 0 0 0 = 0 \times 1 F F D F 004$$