CSE 340

Computer Architecture

Student Name : Rejwan Shafi

Student ID : 23241108

Section : 09

Assignment No: : 02

Faculty : STK (Sumaiya Tanjil Khan)

Date of Submission : 09 / 11 / 2023

Answer to the question no 1

Difference between program counter and $zero register is as follows.

Program Counter	$zero register
• Program Counter points to the address of the instruction that is currently being executed.	• $zero register always hold the constant zero.
• Program Counter is read only and can be updated by the CPU. When the Program Counter points to the next instruction	• $zero register is read only and it can't be updated

In case of 16 bit architecture, increment in memory address for sequential instruction execution

$= \frac{16}{8} = 2$

Similarly, in case of 128 bit architecture, Increment in memory address for sequential instruction execution = $\frac{128}{8} = 16$

Answer to the question no 2

The given MIPS instruction is,

lw $4, X ($5)

As we are using 256 bit architecture ; so, incnement in memory address would be $= \frac{256}{8} = 32$

We want to load the content of A[5], Given that array's base address in stored in $5. So, the offsetis value, $x = 5 \times 32 = 160$

∴ Value of X is 160.

Answer to the question no 3

```
sll   $t0,   $s1,  0      # As I is already 32 bit
add   $t0,   $t0,  $s0
lb    $t1,   0 ($t0)
sw    $t1,   0 ($s2)
```

Answer to the question no 4

Given that,

X is stored in $s0

Y is stored in $s1.

Base address of Arr is stored in $s4.

and the given code is,

```
X = 15Y - 5;
Arr[5] = 2X + Arr[10];
```

MIPS Code:

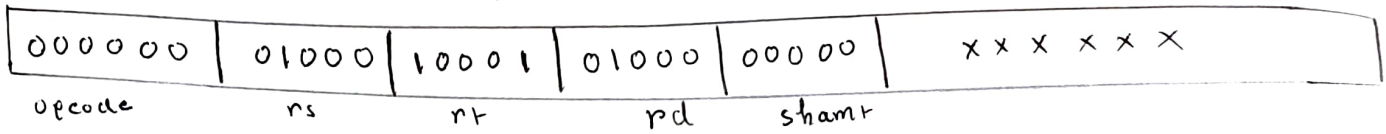
```
sll   $t0,  $s1, 4
sub   $t0,  $t0, $s1.
addi  $s0,  $t0, -5
lw    $t1,  40 ($s4)
add   $t2,  $s0, $s0
add   $t1,  $t2, $t1
sw    $t1,  20 ($s4)
```

Machine Code for each instruction.

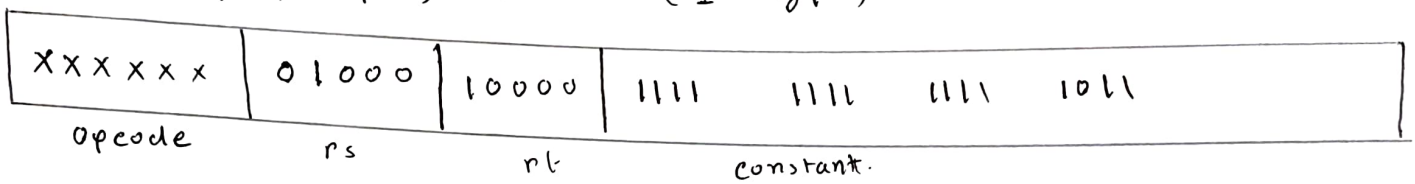
- sll $t0, $s1, 4 (R type)

000000	00000	10001	01000	00100	xxxxxx
opcode	rs	rt	rd	shamt	funct

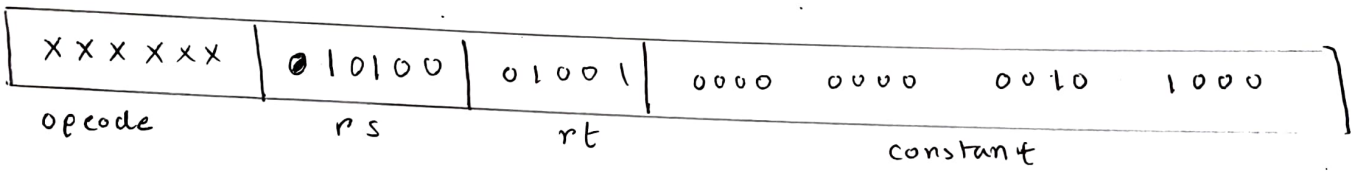
• sub $t0, $t0, $s1, (R type)



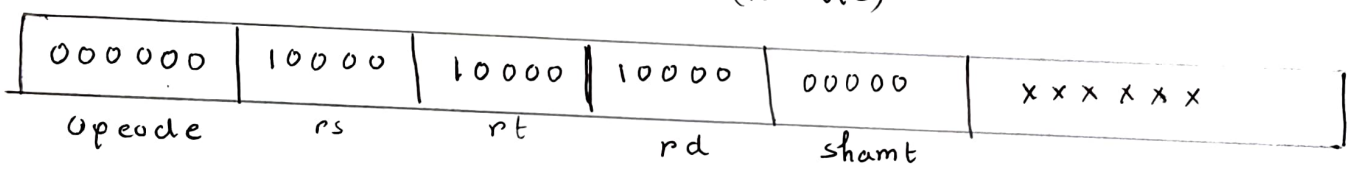
• addi $s0, $t0, -5 (I type)



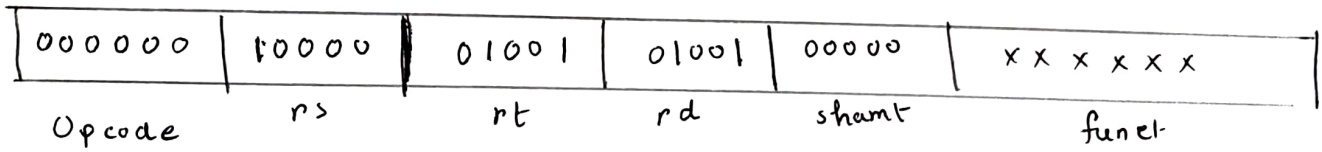
• lw $t1, 40 ($s4), (I type)



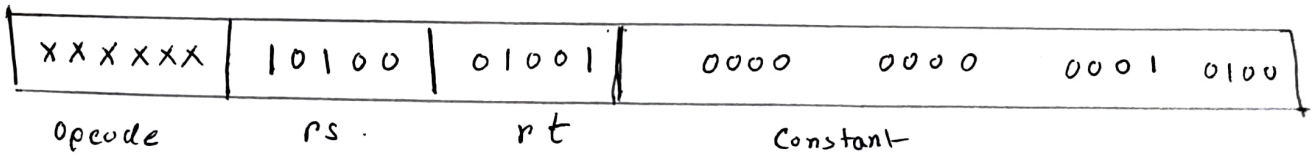
• add $s0, $s0, $s0 (R type)



• add $t1, $s0, $t1. (R type)



• sw $t1, 20 ($s4) (I type)



Answer to the question no 5

Given that,

MIPS instruction

beq $9, $8, 124

and PC's value is = $(1278A4B1)_{16}$

from the MIPS instruction we can get the offset value which is $(124)_{10}$ = $(0000\ 0000\ 0111\ 1100)_2$

After 2 bit left shift 18 bit representation of offset value would be = $(0000\ 0000\ 0111\ 1100\ 00)_2$

After sign extension ; 32 bit representation of offset value would be, = $(0000\ 0000\ 0000\ 0000\ 0000\ 0001\ 1111\ 0000)_2$

$= (00000\ 1F0)_{16}$

∴ Branch address in hex would be

$= (PC + 4 + 00000\ 1F0)_{16}$

$= (1278A4B1 + 4 + 00000\ 1F0)_{16}$

$= (1278A6A5)_{16}$

∴ Branch address in hex would be, 0X1278A6A5

Answer to the question no 6

Given that

offset = 1590

PC holds the address = $(00\ AB\ 1203)_{16}$

$\therefore PC+4 = (00AB1207)_{16} = (0000\ 0000\ 1010\ 1011\ 0001\ 0010\ 0000\ 0111)_2$

$\therefore$ (PC+4)'s MSB 4 bits = 0000

$\therefore$ Jump address = (PC+4 's MSB 4 bits) + (offset × 4)

$= (0000)_2 + (1590 \times 4)_{10}$

$= (0000)_2 + (6360)_{10}$

$= (0000)_2 + (0000\ 0000\ 0000\ 0000\ 0001\ 1000\ 1101\ 1000)_2$

$= (0000\ 0000\ 0000\ 0000\ 0001\ 1000\ 1101\ 1000)_2$

$= (00001\ 8D8)_{16}$.

$\therefore$ Jump address = 0x 0000 18D8

Answer to the question no 7

Given that,

MIPS instruction is,

lw $8, 52 ($17)

offset = $(52)_{10}$ = (offset x 4)

and PC holds the address = $(1563\,2017)_{16}$

= $(358\,817\,815)_{10}$

∴ Memory address = Base address + (offset x 4).

= $(358\,817\,815)_{10}$ + $(52)_{10}$

= $(358\,817\,867)_{10}$

= $(1563\,204B)_{16}$.

∴ Memory address = 0x 1563 204B

Answer to the question no 8

```
add $s3, $s3, $zero.  # initializing i = 0.
Loop:
    slti $t0, $s3, 10
    beq  $t0, $zero, Exit
    sll  $t0, $s3, 2
     add  $t0, $t0, $s1. # Memory address of A[i]
     lw   $t1, 0($t0).
     beq  $t1, $s4, Else.
      sll  $t1, $s3, 2.
       add  $t1, $t1, $s2 # Memory address of B[i].
       lw   $t2, 0($t1). # Content of B[i]
        sll  $t2, $t2, 2
        add  $t2, $t2, $s1. # Memory address of A[B[i]]
        lw   $t3, 0($t2). # Content of A[B[i]]
        add  $t3, $t3, $s5  # A[B[i]] + 1
         sw   $t3, 0($t2) # A[B[i]] = A[B[i]] + 1.
         add  $s3, $s3, $s5.  # i += 1
         j Loop.
Else:
      add  $t2, $s3, $s5.
      sll  $t2, $t2, 2
       add  $t2, $t2, $s2. # Memory address of B[i+1]
       lw   $t3, 0($t2) # Content of B[i]
       sw   $t3, 0($t0). # A[i] = B[i+1]
       add  $s3, $s3, $s5. # i += 1
       j Loop
Exit:
```

Answer to the question no 9

```
addi  $s1,  $s1, 20
addi  $s2,  $s1, -10
addi  $s0,  $s0, 7
add   $s3,  $s2, $s0
jal   sum
j Exit

sum:
addi  $sp,  $sp, -4
sw    $s0, 0($sp)
add   $t0,  $a0, $a1   # (x+y)
add   $t0,  $t0, $a2   # (x+y+z)
add   $s0,  $t0, $zero # a = x+y+z
add   $v0,  $s0, $zero
lw    $s0, 0($sp)
addi  $sp,  $sp, 4
jr    $ra

Exit:
```

Answer to the question no 10

As we are considering 64 bit MIPS Architectures. Therefore, each memory slot address is 64 bits. Whereas, in each memory slot we can store only 8 bit of data.

∴ The size of data memory for a 64 bit MIPS architecture is $= 2^{64} \times 8$ bits

$= 147573952589676412928$ bits

$= 18446744073709551616$ byte [∵ 1byte = 8 bit]

$= 1.8446744073709551616 \times 10^{16}$ Kilobyte.

$= 18446744073709.55$ Megabyte

$= 18446744.073709548$ Terabyte

$= 18.4467$ Exabyte.