1.

2.14 Provide the type and assembly language instruction for the following binary value: 0000 0010 0001 0000 1000 0000 0010 0000two0000 0010 0001 0000 1000 0000 0010 0000

opcode rs rt rd shamt funct

000000 10000 10000 10000 00000 100000

R-type format add $s0, $s0, $s0

2.17 Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields:

op=0x23, rs=1, rt=2, const=0x4

I-type format lw $v0, 4($at)

Opcode rs rt immediate

100011 00001 00010 0000 0000 0000 0100

2.23 Assume ＄t0 holds the value 0x00101000. What is the value of ＄t2 after the following instructions?

slt ＄t2, ＄0, ＄t0 $t2 = ($0 < $t0) ? 1:0 🡪 $t2 = 1

bne ＄t2, ＄0, ELSE $t2 = 1 =/= 0 = $0 🡪ELSE

j  DONE

ELSE:  addi  ＄t2, ＄t2, 2 $t2 = $t2 + 2 = 1 + 2 = 3

DONE:

So, the value of $t2 is 3.

2. Assemble the following MIPS assembly language code snippet into machine code (each instruction is a 32-bit word shown in hexadecimal), assuming that the first instruction locates at memory address 0x00400000.

Show your work by describing the steps done to reach the machine code representation for each instruction (i.e. get the instruction format, opcode, and the operands for each of the instruction.) Your work should be typed, not handwritten.

main:     add $t0, $zero, $zero

              addi  $t1, $zero, 1

next:       sub   $s3, $s2, $s7

loop:       add $t0, $t0, $t1

              sub   $s1, $s1, $s3

             slti     $s0, $t0, 100

             bne  $s0, $zero, loop

             j next

add $t0, $zero, $zero

add = R-format, opcode = 0, funct = 2016, $t0 = 8, $zero = 0

0x00400000 0 0 0 8 0 20

addi  $t1, $zero, 1

addi = I-format, opcode = 816, $t1 = 9, $zero = 0, imm = 1 = 116

0x00400004 8 0 9 1

sub   $s3, $s2, $s7

sub = R-format, opcode = 0, funct = 2216, $s3 = 19, $s2 = 18, $s7 = 23

0x00400008 0 18 23 19 0 22

add $t0, $t0, $t1

add = R-format, opcode = 0, funct = 2016, $t0 = 8, $t1 = 9

0x0040000c 0 8 9 8 0 20

sub   $s1, $s1, $s3

sub = R-format, opcode = 0, funct = 2216, $s1 = 17, $s3 = 19

0x00400010 0 17 19 17 0 22

slti     $s0, $t0, 100

slti = I-format, opcode = a16, $s0 = 16, $t0 = 8, imm = 100 = 416

0x00400014 a 8 16 4

bne  $s0, $zero, loop

bne = I-format, opcode = 516, $s0 = 16, $zero = 0, loop = (12) 🡪 24 – 12 = 12

0x00400018 5 0 16 -c

j next

j = J-format, opcode 216, next = 0x00400008

jmp dst = next = 0x00400008 = 0000 0000 0100 0000 0000 0000 0000 1000

= (0000)2 0000 0100 0000 0000 0000 0000 102 (00)2

0x0040001c 2 0000 0100 0000 0000 0000 0000 102

MIPS assembly language code:

main:     add $t0, $zero, $zero

              addi  $t1, $zero, 1

next:       sub   $s3, $s2, $s7

loop:       add $t0, $t0, $t1

              sub   $s1, $s1, $s3

             slti     $s0, $t0, 100

             bne  $s0, $zero, loop

             j next

Machine code:

0x00400000 0 0 0 8 0 20

0x00400004 8 0 9 1

0x00400008 0 18 23 19 0 22

0x0040000c 0 8 9 8 0 20

0x00400010 0 17 19 17 0 22

0x00400014 a 8 16 4

0x00400018 5 0 16 -c

0x0040001c 2 0000 0100 0000 0000 0000 0000 102