Exercise 2.10

In the following problems, the data table contains bits that represent the opcode of an instruction. You will be asked to interpret the bits as MIPS instructions into assembly code and determine what format of MIPS instruction the bits represent.

a.	$0000\ 0010\ 0001\ 0000\ 1000\ 0000\ 0010\ 0000_{two}$
b.	$0000\ 0001\ 0100\ 1011\ 0100\ 1000\ 0010\ 0010_{two}$

2.10.1 [5] <2.5> For the binary entries above, what instruction do they represent?

2.10.2 [5] <2.5> What type (I-type, R-type, J-type) instruction do the binary entries above represent?

2.10.1

a.	`add $s0, $s0, $s0`
b.	`sub $t1, $t2, $t3`

2.10.2

a.	r-type
b.	r-type

In the following problems, the data table contains MIPS instructions. You will be asked to translate the entries into the bits of the opcode and determine the MIPS instruction format.

a.	`addi $t0, $t0, 0`
b.	`sw $t1, 32($t2)`

2.10.4 [5] <2.4, 2.5> For the instructions above, show the binary then hexadecimal representation of these instructions.

2.10.5 [5] <2.5> What type (I-type, R-type, J-type) instruction do the instructions above represent?

2.10.4 & 2.10.5

a. 000100 01000 01000 0000 0000 0000 0000 => 0x11080000 I type

b. 101011 01010 01001 0000 0000 0010 0000 => 0xAD490020 I type

Exercise 2.12

In the following problems, the data table contains various modifications that could be made to the MIPS instruction set architecture. You will investigate the impact of these changes on the instruction format of the MIPS architecture.

a.	128 registers
b.	Four times as many different instructions

2.12.1 [5] <2.5> If the instruction set of the MIPS processor is modified, the instruction format must also be changed. For each of the suggested changes above, show the size of the bit fields of an R-type format instruction. What is the total number of bits needed for each instruction?

2.12.2 [5] <2.5> If the instruction set of the MIPS processor is modified, the instruction format must also be changed. For each of the suggested changes above, show the size of the bit fields of an I-type format instruction. What is the total number of bits needed for each instruction?

2.12.1

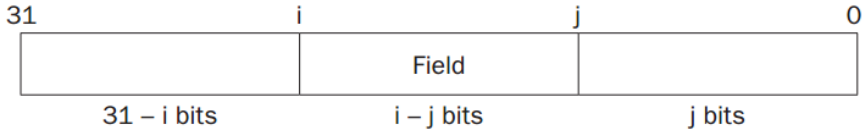
	Type	opcode	rs	rt	rd	shamt	funct	
a.	r-type	6	7	7	7	5	6	total bits = 38
b.	r-type	8	5	5	5	5	6	total bits = 34

2.12.2

	Type	opcode	rs	rt	immed	
a.	i-type	6	7	7	16	total bits = 36
b.	i-type	8	5	5	16	total bits = 34

Exercise 2.14

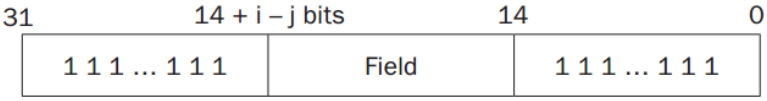
The following figure shows the placement of a bit field in register `$t0`.



In the following problems, you will be asked to write MIPS instructions to extract the bits "Field" from register `$t0` and place them into register `$t1` at the location indicated in the following table.

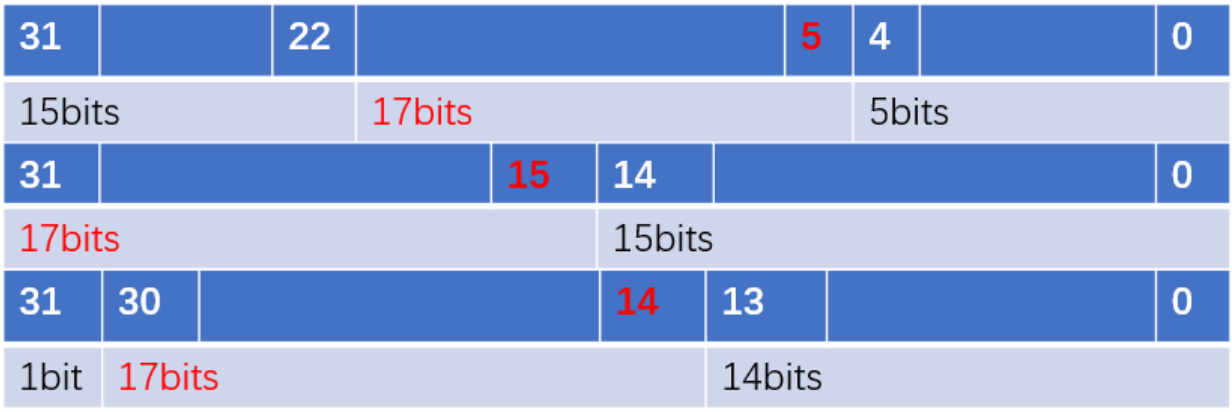
a.

b.



2.14.1 [20] <2.6> Find the shortest sequence of MIPS instructions that extracts a field from `$t0` for the constant values i = 22 and j = 5 and places the field into `$t1` in the format shown in the data table.

2.14.1 [20] <2.6> Find the shortest sequence of MIPS instructions that extracts a field from `$t0` for the constant values i = 22 and j = 5 and places the field into `$t1` in the format shown in the data table.



a. add $t1, $t0, $zero
srl $t1, $t1, 5

sll $t1, $t1, 15

b. the same as a, and then
srl $t1, $t1, 1

Exercise 2.16

For these problems, the table holds various binary values for register `$t0`. Given the value of `$t0`, you will be asked to evaluate the outcome of different branches.

a.	$0010\ 0100\ 1001\ 0010\ 0100\ 1001\ 0010\ 0100_{two}$
b.	$0101\ 1111\ 1011\ 1110\ 0100\ 0000\ 0000\ 0000_{two}$

2.16.4 [5] <2.7> Suppose that register `$t0` contains a value from above. What is the value of `$t2` after the following instructions?

```
        slt  $t2, $0,   $t0
        bne  $t2, $0,   ELSE
        j    DONE
ELSE:   addi $t2, $t2,  2
DONE:
```

2.16.4

a. $t0=0x24924924 $t2 = 1 -> branch to ELSE -> $t2 = 3

b. $t0 > 0 $t2 = 1 -> branch to else -> $t2 = 3

2.16.6 [5] <2.7> Suppose the program counter (PC) is set to 0x2000 0000. Is it possible to use the jump (j) MIPS assembly instruction to get set the PC to the address as shown in the data table above? Is it possible to use the branch-on-equal (beq) MIPS assembly instruction to set the PC to the address as shown in the data table above? Note the format of the J-type instruction.

2.16.6

jump instruction the same [31:28]: a. yes b. no

branch instruction: immediate 16 bits, *4 -> 18 bits a. no b. no

Exercise 2.17

For these problems, there are several instructions that are not included in the MIPS instruction set are shown.

a.	`subi $t2, $t3, 5  # R[rt] = R[rs] - SignExtImm`
b.	`rpt $t2, loop  # if(R[rs]>0) R[rs]=R[rs]-1, PC=PC+4+BranchAddr`

2.17.3 [5] <2.7> For each instruction in the table above, find the shortest sequence of MIPS instructions that performs the same operation.

For these problems, the table holds MIPS assembly code fragments. You will be asked to evaluate each of the code fragments, familiarizing you with the different MIPS branch instructions.

a.	`LOOP: addi $s2, $s2, 2`
	`subi $t1, $t1, 1`
	`bne $t1, $0, LOOP`
	`DONE:`

b.	`LOOP: slt $t2, $0, $t1`
	`beq $t2, $0, DONE`
	`subi $t1, $t1, 1`
	`addi $s2, $s2, 2`
	`j LOOP`
	`DONE:`

- 2.17.2
  - *(a)*

    addi $t2, $t3, -5
  - *(b)* This is a quite common way to write loop.

    slt $t0, $0, $t2

    bne $t0, $0, IF

    IF: addi $t2, $t2, -1

    j LOOP

Exercise 2.18

For these problems, the table holds some C code. You will be asked to evaluate these C code statements in MIPS assembly code.

a.	`for(i=0; i<a; i++)` `  a += b;`

b.	`for(i=0; i<a; i++)`
	`  for(j=0; j<b; j++)`
	`  D[4*j] = i + j;`

2.18.1 [5] <2.7> For the table above, draw a control-flow graph of the C code.

2.18.2 [5] <2.7> For the table above, translate the C code to MIPS assembly code. Use a minimum number of instructions. Assume that the values of `a`, `b`, `i`, and `j` are in registers `$s0`, `$s1`, `$t0`, and `$t1`, respectively. Also, assume that register `$s2` holds the base address of the array `D`.

2.18.3 [5] <2.7> How many MIPS instructions does it take to implement the C code? If the variables `a` and `b` are initialized to 10 and 1 and all elements of D are initially 0, what is the total number of MIPS instructions that is executed to complete the loop?

a.

```
        addi $t0, $0, 0
        beq  $0, $0, TEST
LOOP:   add  $s0, $s0, $s1
        addi $t0, $t0, 1
TEST:   slti $t2, $t0, 10
        bne  $t2, $0, LOOP
```

b.

```
LOOP:   slti $t2, $s0, 10
        beq  $t2, $0, DONE
        add  $t3, $s1, $s0
        sll  $t2, $s0, 2
        add  $t2, $s2, $t2
        sw   $t3, ($t2)
        addi $s0, $s0, 1
        j LOOP
DONE:
```

a.	6 instructions to implement and 44 instructions executed
b.	8 instructions to implement and 2 instructions executed

- 2.18.3
  - *(a)*

    2+2 (first TEST) + 4*10
  - *(b)*

    Directly go to DONE

Exercise 2.26

For this exercise, you will explore the range of branch and jump instructions in MIPS. For the following problems, use the hexadecimal data in the table below.

a.	0x00020000
b.	0xFFFFFF00

2.26.3 [10] <2.6, 2.10> In order to reduce the size of MIPS programs, MIPS designers have decided to cut the immediate field of I-type instructions from 16 bits to 8 bits. If the PC is at address 0x0000000, how many branch instructions are needed to set the PC to the address in the table above?

2.26.3 Branch range is 0x00000200 to 0xFFFFFE04.

a.	256 branches
b.	one branch

0xFFFFFE04 = -256,
256<2^7*4

For the following problems, you will be using making modifications to the MIPS instruction set architecture.

a.	128 registers
b.	Four times as many different operations

2.26.4 [10] <2.6, 2.10> If the instruction set of the MIPS processor is modified, the instruction format must also be changed. For each of the suggested changes above, what is the impact on the range of addresses for a beq instruction? Assume that all instructions remain 32 bits long and any changes made to the instruction format of i-type instructions only increase/decrease the immediate field of the beq instruction.

2.26.4

a.	branch range is 16× smaller
b.	branch range is 4× smaller

Exercise 2.27

In the following problems, you will be exploring different addressing modes in the MIPS instruction set architecture. These different addressing modes are listed in the table below.

a.	Base or Displacement Addressing
b.	Pseudodirect Addressing

2.27.1 [5] <2.10> In the table above are different addressing modes of the MIPS instruction set. Give an example MIPS instructios that shows the MIPS addressing mode.

2.27.2 [5] <2.10> For the instructions in 2.27.1, what is the instruction format type used for the given instruction?

a) sw I-type

b) j J-type