Name-Last Name: ______________________________ **Student ID:** _________________________

Hacettepe University	Computer Engineering Department
BBM234 Computer Organization	Instructor: Suleyman TOSUN
Midterm Exam	
Duration: 2 hours (14:00-16:00)	Exam Date: 19.04.2021

Questions	**1**	**2**	**3**	**Total**
Marks	**40**	**30**	**30**	**100**
Earned				

Read the following statements first and obey the rules.

1) You must solve each question on a separate sheet. You can print this document and solve the questions on it if you have a printer ready.
2) Write your name on the first sheet.
3) Make a single pdf file in the order of the questions. Zip the file directly in a zip archive under the name b<yourstudentID>.zip.
4) Upload the pdf file through submit.cs.hacettepe.edu.tr in the following format:
    - **b<yourstudentID>.zip**
        - **name_surname.pdf**
5) **You can submit only once. If you submit more than once, only your first submission will be graded.**
6) This is an open-book exam.

Q1. You are given the following MIPS program and the data memory. Assume that *K* is the last digit of your student ID.

Instruction memory		
Address	**Instrustions**	**Some info for part b**
0x20	addi $a0, $0, 1	
0x24	ori $a1, $0, ***K***	
0x28	jal Label1	#opcode = 0x03
0x2C	add $s3, $v0, $0	
….	….	
0x3C	Label1: sll $a0, $a0, 1	#funct = 0x00, $a0=$4
0x40	div $a1, $a0	
0x44	mflo $s0	
0x48	mfhi $s1	
0x4C	beq $s1, $0, atla	#opcode = 0x04, $s1 = $17
0x50	lw $s2, 0($0)	
0x54	j skip	
0x58	atla: lw $s2, 4($0)	
0x5C	skip: add $v0, $s2, $0	
0x60	jr $ra	#funct = 0x08, $ra = $31

Data Memory	
Address	**Data**
0x00	33
0x04	44
0x08	55
0x0C	66
…	…

a) Write the values of following registers after the code finishes its execution.

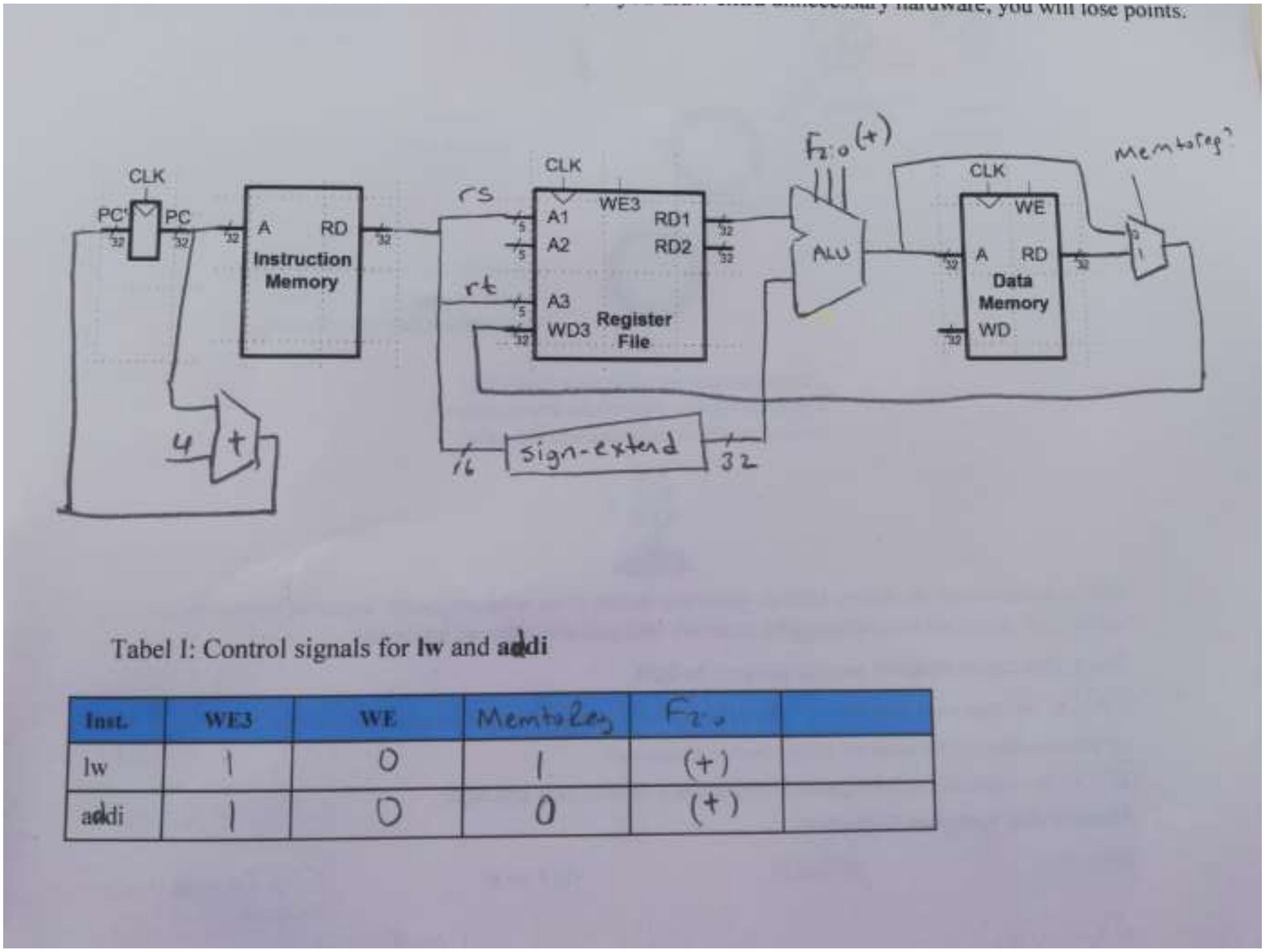
	a0	a1	s0	s1	s2	ra
	2K	K	K/2	Remainder of K/2 (0 if K is even; 1 otherwise)	44 if K is even, 33 if K is odd.	0x0000002C
Points	*2*	*1*	*2*	*5*	*5*	*5*

b) Write the machine codes of the following instructions. Draw their formats with corresponding values in binary in each field. Then, convert it to hexadecimal.

Instruction	**Binary machine code**	**Hexadecimal code**
jal Label1	0000 1100 0000 0000 0000 0000 0000 1111 Opcode Address of sll (0x3C)	0x0C00 000F
sll $a0, $a0, 1	0000 0000 0000 0100 0010 0000 0100 0000 Opcode rs rt rd shamt funct	0x0004 2040
beq $s1, $0, atla	0001 0010 0010 0000 0000 0000 0000 0010 Opcode rs rt Immediate =2	0x1220 0002
jr $ra	0000 0011 1110 0000 0000 0000 0000 1000 Opcode rs rt rd shamt funct	0x03E0 0008

Q2. You have 32-bit Program Counter (PC), instruction memory, register file, and data memory.

Draw the data-path of single-cycle processor that can execute <u>ONLY **lw** and **addi** instructions</u>.
Define necessary control signals for data memory, register file, and the multiplexers if you have any. Fill out the control signal table with newly added control signals and appropriate values for each signal. Do not use the control signals for the data-path of whole MIPS processor we have seen in the class.
You may need to add extra hardware. However, if you draw extra unnecessary hardware, you will lose points.



Tabel I: Control signals for **lw** and **addi**

Inst.	WE3	WE	MemtoReg	F2:0	
lw	1	0	1	(+)	
addi	1	0	0	(+)	

Q3. You are given an array A[100] with 100 integer elements. Assume that the address of the first element is 0xABCD8000.

The last digit of your student ID is *K*.

Write a MIPS code that calculates the average of the elements in A that are greater than *K*. Average value can be integer rather than floating point. Store the result in v0 register.

```
            lui $t0, 0xABCD             #t0=0xABCD0000
            ori $t0, $t0, 0x8000        #address of A[0] in $t0 ($t0=0xABCD8000
            addi $s0, $0, K             #$s0=K
            addi $t1, $0, 100           #$t1=100
            addi $t2, $0, 0             #$t2=0 (i=0)
            addi $s1, $0, 0             #total = $s1 = 0
            addi $s2, $0, 0             #$s2 = Number of elements greater than K
loop:       beq $t1, $t2, done          #if i=100, done
            lw $t3, 0($t0)              #$t3=A[i]
            slt $t4, $s0, $t3           #if K<A[i], $t4=1
            beq $t4, $0, skip           #if $t4=0, skip ( do not add the number to the total)
            add $s1, $s1, $t3           #Total = Total + A[i]
            addi $s2, $s2, 1            #Number of elements is increased by one
skip:       addi $t2, $t2, 1            #i=i+1
            addi $t0, $t0, 4            #Increment the address by 4 for next array element
            j loop
done:       div $s1, $s2                #Calculate average
            mflo $v0                    #Store the result in v0
```