Name-Last Name: _____ **Student ID:** _____

Section No (1,2,3?): ____

Hacettepe University	Computer Engineering Department
BBM234 Computer Organization	Instructors: Assoc. Prof. Dr. Suleyman TOSUN
Midterm Exam	Assist. Prof. Dr. Mehmet KOSEOGLU
Duration: 120 minutes	Exam Date: 18.04.2017

Questions	1	2	3	4	4	5	Total
Marks	20	20	15	10	20	15	100
Earned							

Q1. a) Write the values of the registers after the following MIPS program finishes its execution.

		Register (1 point each)
	lui $s0, 0x1234	s0=0x12340000
	ori $s0, $s0, 0x0335	s0=0x12340335
	andi $s0, $s0, 0x000F	s0=5
	sra $s1, $s0, 2	s1=1
	or $s2, $s0, $s1	s2=5
	slt $s3, $s1, $s2	s3=1
	bne $s1, $s3, else	s1=1
	addi $s2, $s2, -1	s2=4
else:	sll $s4, $s2, 2	s4=16
	jr $ra	

b) For the given "*number*" value, what does function f1 do? Write output values (value in s0) for the given *number* values in the table.

```
main: addi $a0, $0, number
      addi $sp, $sp, -4
      sw $ra, 0($sp)
      jal f1
      add $s0, $v0, $0
      lw $ra, 0($sp)
      addi $sp, $sp, 4
      jr $ra   #exit

f1:   addi $t0, $0, 0
      addi $v0, $0, 1
      bne $a0, $0, else
      jr $ra
else: beq $a0, $t0, done
      addi $t0, $t0, 1
      mul $v0, $v0, $t0
      mflo $v0
      j else
done: jr $ra
```

Write the description of f1 below:

F1 calculates the factorial of given number. [5points]

Number	0	3	5
S0 [2 points each]	1	6	120

Q2. You have four instructions stored in the memory as given in the following table:

Instructions	Address	Instruction
Inst1	0x00400000	0x14100003
Inst2	0x00400004	0x012A4025
Inst3	0x00400008	0x2210FFFB
Inst4	0x0040000C	0x08100000
Inst5	0x00400010	---

a) Write the binary values for each instruction. Clearly show which bits corresponds to which field in the instruction format (opcode, rs, rt, rd.. etc?).

Instructions	Instruction format
0x14100003	0001 0100 0001 0000 0000 0000 0000 0011
0x012A4025	0000 0001 0010 1010 0100 0000 0010 0101
0x2210FFFB	0010 0010 0001 0000 1111 1111 1111 1011
0x08100000	0000 1000 0001 0000 0000 0000 0000 0000

b) Write down the corresponding MIPS assembly code below for each machine code.

Instructions	MIPS Code [5 points(3+2) each]
Inst1	Label: bne $s0, $0, Done
Inst2	or $t0, $t1, $t2
Inst3	addi $s0, $s0, -5
Inst4	j Label
Inst5	Done:

Name	Register
$0	0
$at	1
$v0-$v1	2-3
$a0-$a3	4-7
$t0-$t7	8-15
$s0-$s7	16-23
$t8-$t9	24-25
$k0-$k1	26-27
$gp	28
$sp	29
$fp	30
$ra	31

Instruction	Opcode
j	000010
jal	000011
beq	000100
bne	000101
addi	001000
slti	001010
andi	001100
ori	001101
xori	001110
lui	001111
lw	100011
sw	101011

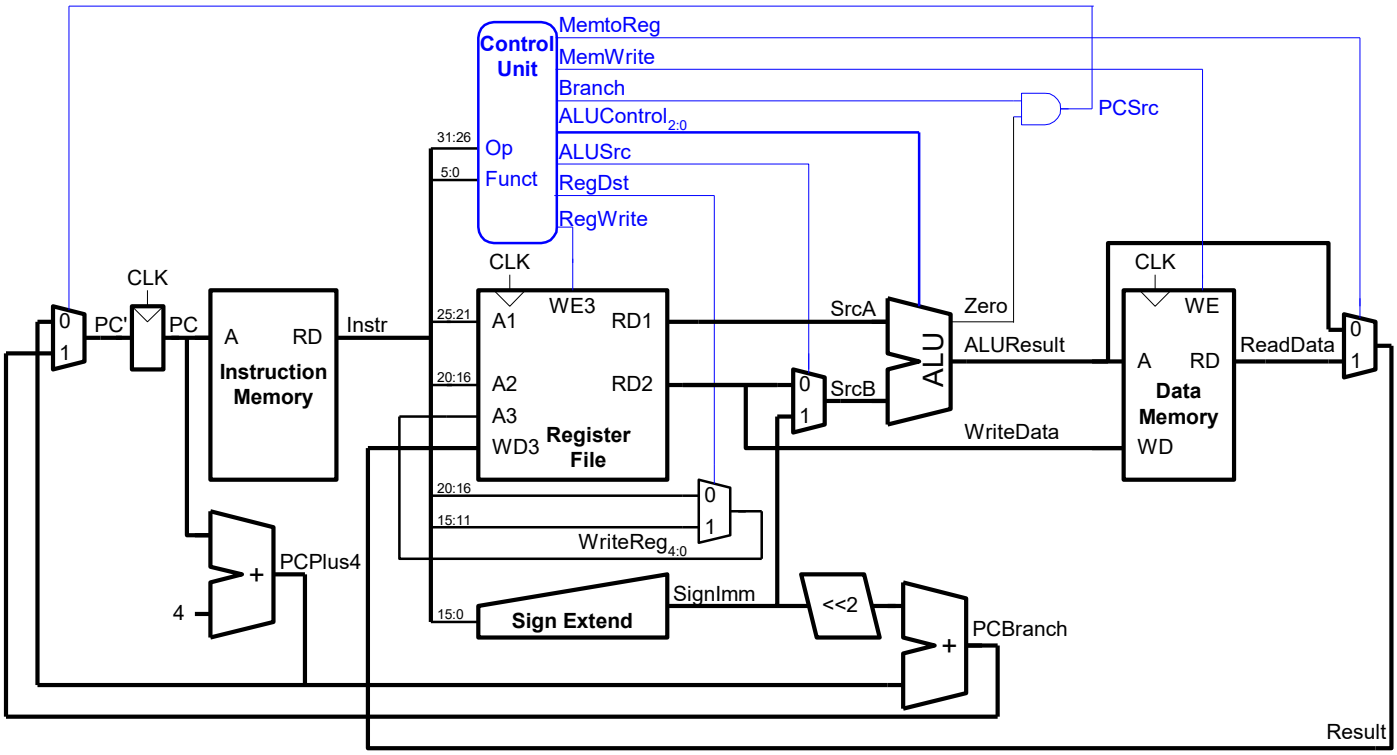
Instruction	Funct
sll	000000
srl	000010
sra	000011
jr	001000
div	011010
add	100000
sub	100010
and	100100
or	100101
xor	100110
nor	100111
slt	101011

Q3. MIPS architesture has some conditional branches and unconditional jumps. We list some of them below. For each instruction type, write the maximum number of instructions between the current program counter (PC) and the target instruction. You should also write the instruction type.

Instruction	Maximum number of instructions that we can jump over (2 points each)	Instruction type (1 point each)
J	2^{26}	J
JR	2^{30}	R
JAL	2^{26}	J
BEQ	$2^{15}+1$	I
BNE	$2^{15}+1$	I

Q4. We would like to add R-type lw instruction (**lwr $rd, $rt($rs)**) to single cycle MIPS processor. The lwr instruction is similar to lw but it sums two registers (specified by $rs, $rt) to obtain the effective load address and uses the R-Type format. Loaded word from memory is written to register rd.

a) Show the necessary changes on data-path of single-cycle processors if any.
b) Fill the control signals in Table I.



There is no need to change data-path.

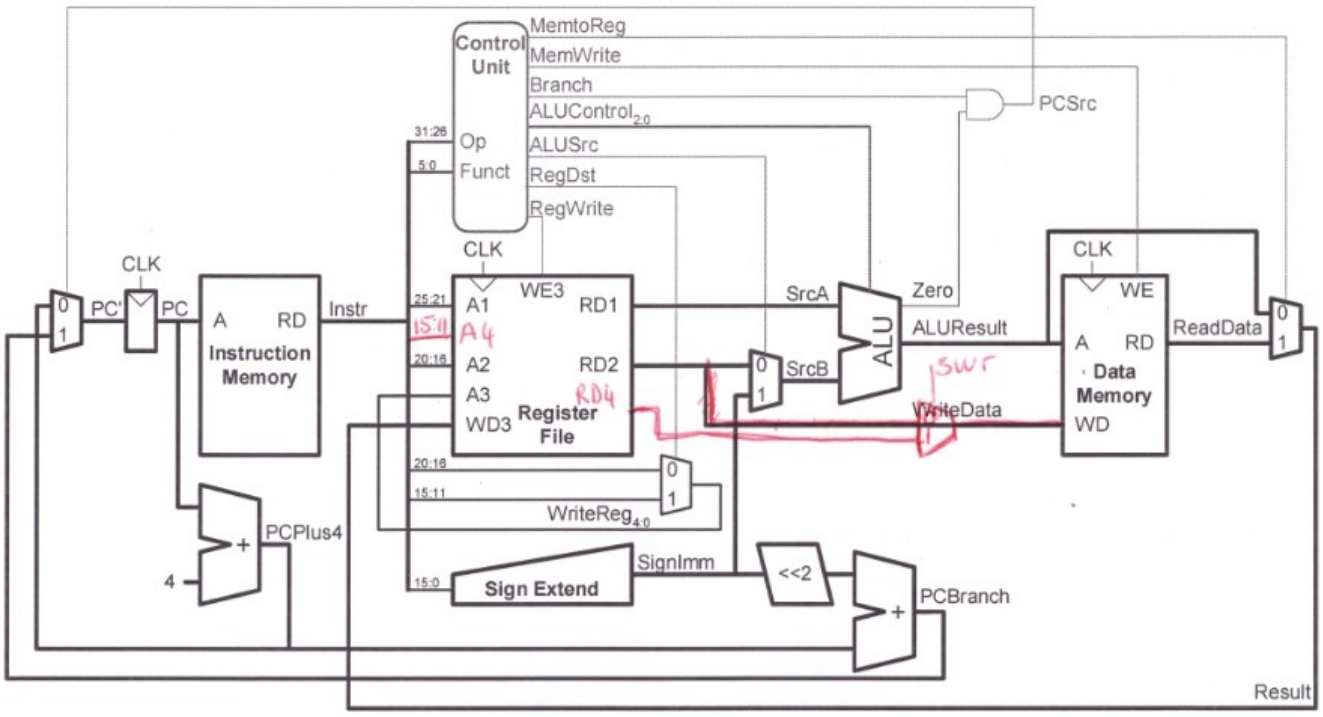
Tabel I: Control signals for **lwr** instruction

Inst.	$Op_{31:26}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	$ALUOp_{1:0}$
lwr	XXXXX	1	1	0	0	0	1	00

Q5. We would like to also add R-type sw instruction (**swr $rd, $rt($rs)**) to single cycle MIPS processor. The swr instruction is similar to sw but it sums two registers (specified by $rs, $rt) to obtain the effective load address and uses the R-Type format. Then, it writes the word in rd register to the memory address.

a) Show the necessary changes on data-path of single-cycle processors if any. Be aware that register file has only two read ports!

b) Fill the control signals in Table II.



We add another read port to be able to read rd register. We add multiplexer in front of WD port of data memory and we select rd by setting srw select line of multiplexer. Control signals are below.

Tabel II: Control signals for **swr** instruction

Inst.	$Op_{31:26}$	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	$ALUOp_{1:0}$	swr
swr	XXXX	0	X	0	0	1	X	10 (or 00)	1

Q6. For the multi-cycle processor given in Figure 3, how many clock cycles does it take to execute following MIPS program? Write how many times each instruction is fetched, clock cycle for each execution and total cycles for each instruction in the table.

	How many times is it fetched? [5 points]	Number of clock cycles for each execution [5 points]	Total execution times for each instruction [2 points]
main: addi $a0, $0, ***3***	1	4	4
addi $sp, $sp, -4	1	4	4
sw $ra, 0($sp)	1	4	4
jal f1	1	3	3
add $s0, $v0, $0	1	4	4
lw $ra, 0($sp)	1	5	5
addi $sp, $sp, 4	1	4	4
jr $ra	1	3	3
f1: addi $t0, $0, 0	1	4	4
addi $v0, $0, 1	1	4	4
bne $a0, $0, else	1	3	3
jr $ra	0	3	0
else: beq $a0, $t0, done	4	3	12
addi $t0, $t0, 1	3	4	12
mult $v0, $v0, $t0 #R-type	3	4	12
mflo $v0 #R-type	3	4	12
j else	3	3	9
done:jr $ra	1	3	3
Total:	28		102

What is the CPI for this program? CPI=102/28 [3 points]

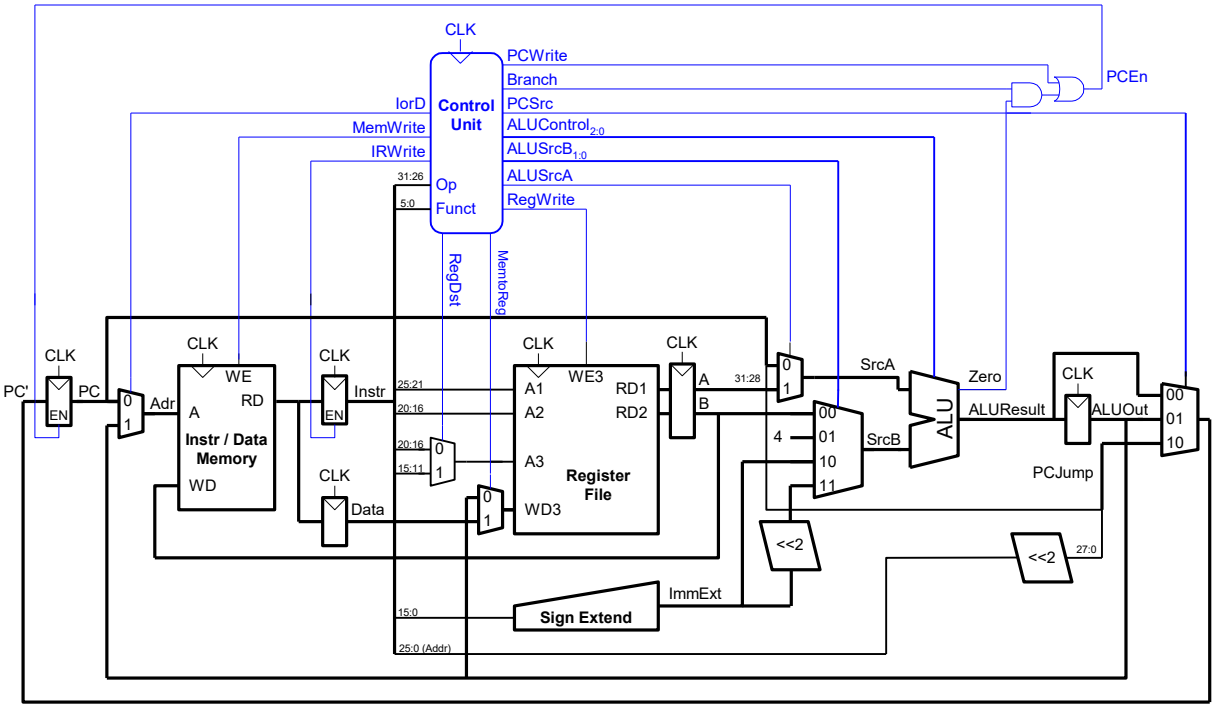


Figure 3:Multi cycle processor