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

Hacettepe University	Computer Engineering Department
BBM234 Computer Organization	Instructors: Assoc. Prof. Dr. Suleyman TOSUN
Final Exam	Assist. Prof. Dr. Mehmet KOSEOGLU
Duration: 120 minutes	Exam Date: 01.06.2017

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

Q1. a) Write the values of the registers and the stack for the following MIPS program. The value of the stack pointer is initially sp=0x7FFFFFFC.

Address		Instructions			
0x00400000		lui $s0, 0x1000		s0=	0x10000000
0x00400004		ori $s0, $s0, 0x0008		s0=	0x10000008
0x00400008		lw $a0, -4($s0)		a0=	10
0x0040000C		addi $a1, $s0, -8		a1=	0x10000000
0x00400010		lw $s1, 4($s0)		s1=	15
0x00400014		add $a2, $s1, $0		a2=	15
0x00400018		jal Proc1			
0x0040001C		addi $s2, $v0, $0		s2=	5
0x00400020					
0x00400024	Proc1:	addi $sp, $sp, -12		sp=	0x7FFFFFF0
0x00400028		sw $ra, 8($sp)		Write the stored values on stack!	
0x0040002C		sw $s0, 4($sp)			
0x00400030		sw $s1, 0($sp)			
0x00400034		addi $v0, $0, 0			
0x00400038	Loop:	beq $a0, $0, Done			
0x0040003C		lw $s0, 0($a1)			
0x00400040		slt $s1, $s0, $a2			
0x00400044		beq $s1, $0, Next			
0x00400048		addi $v0, $v0, 1			
0x0040004C	Next:	addi $a0, $a0, 4			
0x00400050		j Loop			
0x00400054	Done:	lw $ra, 8($sp)		ra=	0x0040001C
0x00400058		lw $s0, 4($sp)		s0=	0x10000008
0x0040005C		lw $s1, 0($sp)		s1=	15
0x00400060		addi $sp, $sp, 12		sp=	0x7FFFFFFC
0x00400064		jr $ra			

Address	Data
0x100000000	13
0x100000004	10
0x100000008	21
0x10000000C	15
0x100000010	7
0x100000014	16
0x100000018	11
0x10000001C	6
0x100000020	30
0x100000024	28

Address	Stack Data
0x7FFFFFFC	XXXXXXXX
0x7FFFFFF8	0x0040001C
0x7FFFFFF4	0x10000008
0x7FFFFFF0	15

b) Briefly describe what Proc1 function does.

It counts the number of values in the array that are less than 15.

Q2. Consider a virtual memory system that can address a total of 2^{32} bytes. You have only 8 MB of physical (main) memory. Assume that page size is 4 KB.

(a) How many bits is the physical address?

8MB = 2^{23}B => 23 bits

(b) What is the maximum number of virtual pages in the system?
$2^{32}/2^{12}=2^{20}$

(c) How many physical pages are in the system?
$2^{23}/2^{12}=2^{11}$

(d) How many bits are the virtual page numbers?
20

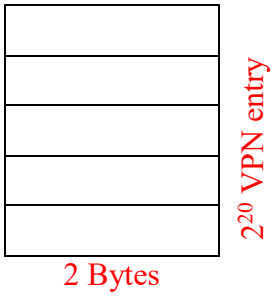
(e) How many bits are the physical page numbers?
11

(f) Assume that, in addition to the physical page number, each page table entry (each page table line) also contains some status information in the form of a valid bit (*V*) and a dirty bit (*D*). How many bytes long is each page table entry? (Round up to an integer number of bytes.)

D	V	PPN
1	1	11

Total of 13 bits. When we round up, we need 2 Bytes for each entry.

(g) Sketch (draw) the layout of the page table. What is the total size of the page table in bytes? [4]



Total size = 2.2^{20}Bytes=2MB

(h) Assume we have TLB with 4 entries. Sketch the TLB by clearly labeling all fields and indicating the number of bits for each field. Assume TLB uses LRU replacement. [4]

	Entry 3				Entry 2			Entry 1			Entry 0		
	U	V	VPN	PPN	V	VPN	PPN	V	VPN	PPN	V	VPN	PPN
Bits	2	1	20	11	1	20	11	1	20	11	1	20	11

(i) What is the total size of the TLB in bits?

130

Q3. a) Suppose we have a five stage (IF, ID, EX, MEM, WB) pipeline architecture with no hazard unit. Assume register write and read can be done at the same clock cycle. You are given the following MIPS code that executes on this architecture. Fill the given table by writing corresponding stages for each clock cycle. You must add NOP instructions if necessary.

MIPS code:

```
add $s0, $s0, $s1

add $s2, $s3, $s4

addi $t0, $t0, 2

sub $s0, $s0, $s2

mul $s5, $s2, $t0

div $s6, $s0, $t0

add $s6, $s5, $s6
```

Clock Cycle / Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
add $s0, $s0, $s1	IF	ID	EX	ME	WB																	
add $s2, $s3, $s4		IF	ID	EX	ME	WB																
addi $t0, $t0, 2			IF	ID	EX	ME	WB															
NOP				IF	ID	EX	ME	WB														
sub $s0, $s0, $s2					IF	ID	EX	ME	WB													
mul $s5, $s2, $t0						IF	ID	EX	ME	WB												
NOP							IF	ID	EX	ME	WB											
div $s6, $s0, $t0								IF	ID	EX	ME	WB										
NOP									IF	ID	EX	ME	WB									
NOP										IF	ID	EX	ME	WB								
add $s6, $s5, $s6											IF	ID	EX	ME	WB							

b) Suppose that EX stage takes different number of clock cycles for different instructions as given in the following table. Fill the given table for this pipeline architecture by inserting necessary NOP instructions.

Instruction type	Number of EX clock cycles
add, sub	2
mul, div	4

MIPS code:

add $s0, $s0, $s1

add $s2, $s3, $s4

addi $t0, $t0, 2

sub $s0, $s0, $s2

mul $s5, $s2, $t0

div $s6, $s0, $t0

add $s6, $s5, $s6

Clock Cycle / Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
add $s0, $s0, $s1	IF	ID	EX1	EX2	ME	WB																
add $s2, $s3, $s4		IF	ID	EX1	EX2	ME	WB															
addi $t0, $t0, 2			IF	ID	EX1	EX2	ME	WB														
NOP				IF	ID	EX	ME	WB														
NOP					IF	ID	EX	ME	WB													
sub $s0, $s0, $s2						IF	ID	EX1	EX2	ME	WB											
mul $s5, $s2, $t0							IF	ID	EX1	EX2	EX3	EX4	ME	WB								
NOP								IF	ID	EX	ME	WB										
NOP									IF	ID	EX	ME	WB									
div $s6, $s0, $t0										IF	ID	EX1	EX2	EX3	EX4	ME	WB					
NOP											IF	ID	EX	ME	WB							
NOP												IF	ID	EX	ME	WB						
NOP													IF	ID	EX	ME	WB					
NOP														IF	ID	EX	ME	WB				
NOP															IF	ID	EX	ME	WB			
add $s6, $s5, $s6																IF	ID	EX1	EX2	ME	WB	