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

Hacettepe University	Computer Engineering Department
BBM234 Computer Organization	Instructors: Suleyman TOSUN, Mehmet KOSEOGLU
Midterm Exam (Take Home due to Covid-19 lock down)	
Duration: 3 hours (13:00-16:00)	Exam Date: 11.05.2020

Questions	1	2	3	4	Total
Marks	25	25	30	20	100
Earned					

Question 0: Write the following statements on a separate sheet by hand and sign. Include the sheet as the first page of your pdf file. If you do not include these statements in your final submission, your exam will not be graded.

1) I will solve each question on a seperate sheet. I will write my name on top of each sheet.
2) I will make a single pdf file in the order of the questions.
3) I will upload the pdf file through submit.cs.hacettepe.edu.tr.
4) I understand that computer-typed solutions will not be accepted. I must handwrite.
5) I will not attempt any form of cheating.
6) I am aware that this is an open-book exam.

Q1. Suppose we have two arrays A[10] and B[10]. Write a MIPS code that does the following:

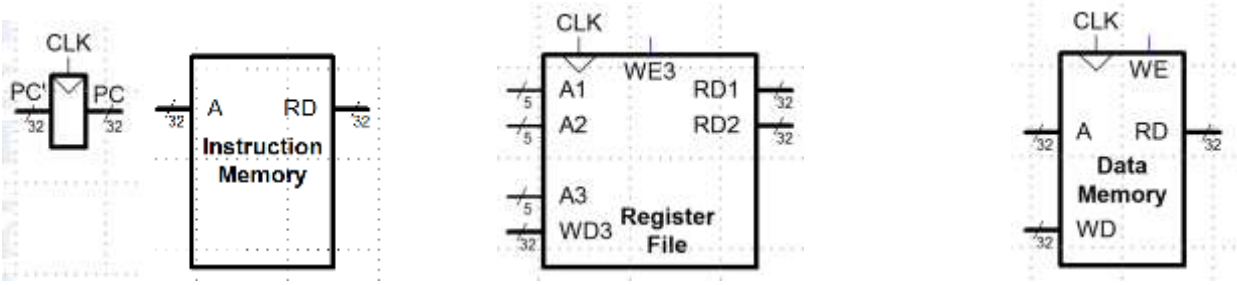
It compares A[i] and B[i]. If A[i] <B[i], it swaps their values. Otherwise, there is no swap. There is an example below. Suppose the addresses of A[0] and B[0] are in registers t0 and t1, respectively.

Inputs			**Outputs**	
A	**B**		**A**	**B**
5	6	5<6; swap	6	5
10	8	10>8; no swap	10	8
2	5	2<5; swap	5	2
8	8	8=8; no swap	8	8
-5	4	-5<4; swap	4	-5
4	-2	4>-2; no swap	4	-2
20	10	20>10; no swap	20	10
3	5	3<5; swap	5	3
10	11	10<11;swap	11	10
5	3	5>3; no swap	5	3

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

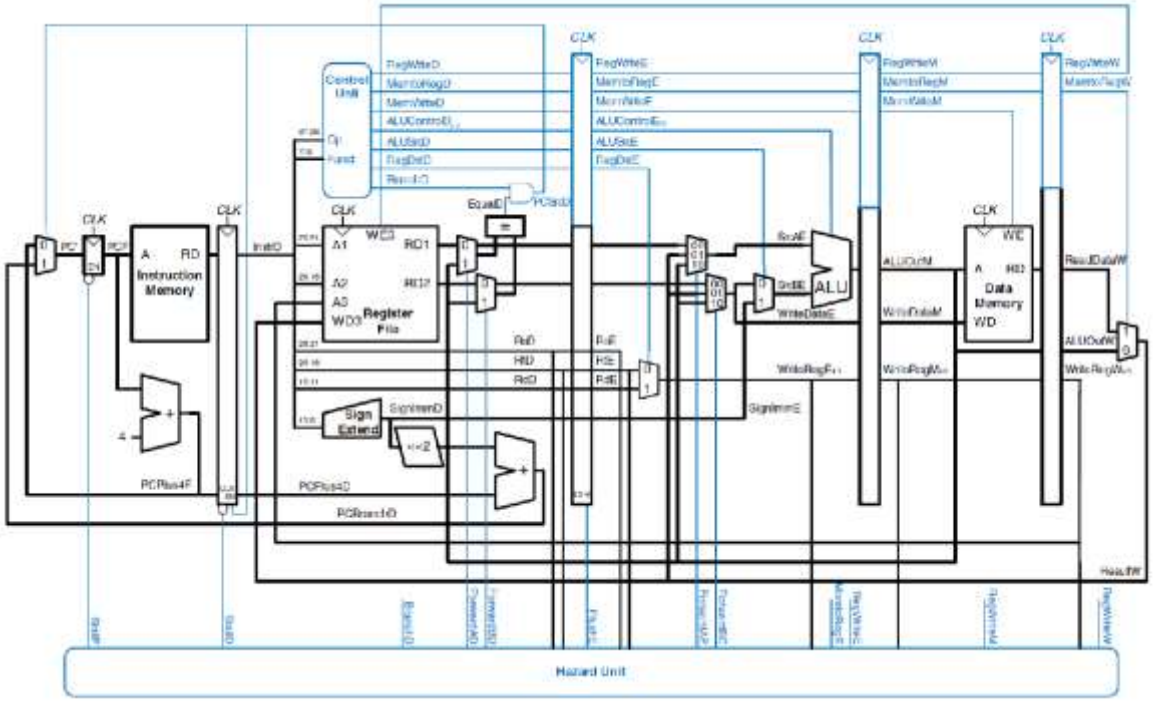
Draw a single-cycle processor that can execute ONLY the following type of instructions. Define necessary control signals and write their values in the table. Note that you do not need some of the control signals in the table. Write NA (Not Applicable) to them if they are not used in your microarchitecture.

```
lui $s0, $s1, 0x3428
ori $t0, $s0, 0xA65F
```



Tabel I: Control signals for **lui** and **ori**

Inst.	RegWrite	RegDst	AluSrc	Branch	MemWrite	MemtoReg	ALUOp
lui							
ori							



Q3. Solve the part a and b according to the following sample program, which is executed in the pipelined architecture above. The current stages of the instructions are shown next to the instructions in parentheses.

```
sw  $t2, $t3(0) (WriteBack)
add $s0, $s2, $s3 (Memory)
or $s4, $s0, $s1 (Execute)
sw  $t5, $t4(0) (Decode)
sub $s0, $s1, $s2 (InstructionFetch)
```

a) (5 points) Fill the control signals of the instructions in the following table. Note that you get points only if the majority of your answers are correct.

MemWriteD	ALUSrcE	RegDstE	MemWriteM	RegWriteW

When the code above is executed, the hazard unit sets ForwardAE to 10 because `add $s0, $s2, $s3` instruction changes the register $s0, which is used by the `or $s4, $s0, $s1` instruction. Considering this hint, answer the following questions:

b) (8 points) Modify the above program such that the hazard unit to set the ForwardBE to 10. Indicate the stages of each instruction similar to the example above. Explain your reasoning

c) (8 points) Write a 5-line MIPS program such that the hazard unit to set the ForwardAE to 01. Indicate the stages of each instruction similar to the example above.

d) (9 points) Write a 5-line MIPS program, which causes the hazard unit to set the ForwardAD to 1. Indicate the stages of each instruction similar to the example above.

Q4.

a) (3 points) Assume we have a 5-stage MIPS processor and it has a forwarding unit (but no stalls). For the given code executing on this processor, insert NOPS to the code if necessary. How many clock cycles does it take to finish to execute this code.

```
lw $s1, 0($s0)
lw $s2, 4($s0)
add $s3, $s1, $s2
sw $s3, 12($s0)
lw $s4, 8($s0)
add $s5, $s1, $s4
sw $s5, 16($s0)
```

b) (5 points) Rearrange the code given above if possible to solve the data hazard(s). Is there any reduction on the total clock cycles?

c) (4 points) Assuming no hazard unit (no flush, no stall, no forwarding) and branch decision is done in the memory stage, insert NOPs to the following code to solve the hazard(s).

```
add $s4, $s5, $s6
beq $s1, $s2, Target
lw $s3, 300($s0)
sub $s7, $s8, $s9
sw $t1, 4($8)
Target:
```

d) (4 points) Assuming no hazard unit (no flush, no stall, no forwarding) and branch decision is done in the decode stage, insert NOPs to the following code to solve the hazard(s).

```
add $s4, $s5, $s6
beq $s1, $s2, Target
lw $s3, 300($s0)
sub $s7, $s8, $s9
sw $t1, 4($8)
Target:
```

e) (4 points) Assuming no hazard unit (no flush, no stall, no forwarding) and branch decision is done in the decode stage, insert NOPs to the following code to solve the hazard(s).

```
add $s4, $s5, $s1
lw $s1, 0($s4)
beq $s1, $s0, Target
lw $s3, 300($0)
.
.
.
Target:
```