İbrahim Burak Tavşanlı 21827852

I will solve each question on a seperate sheet. I will write my name on top of each sheet.

I will make a single pdf file in the order of the questions

I will upload the pdf file through submit.cs.hacettepe.edu.tr.

I understand that computer-typed solutions will not be accepted. I must handwrite

I will not attempt any form of cheating

I am aware that this is an open book exam.

B. Tavşanlı

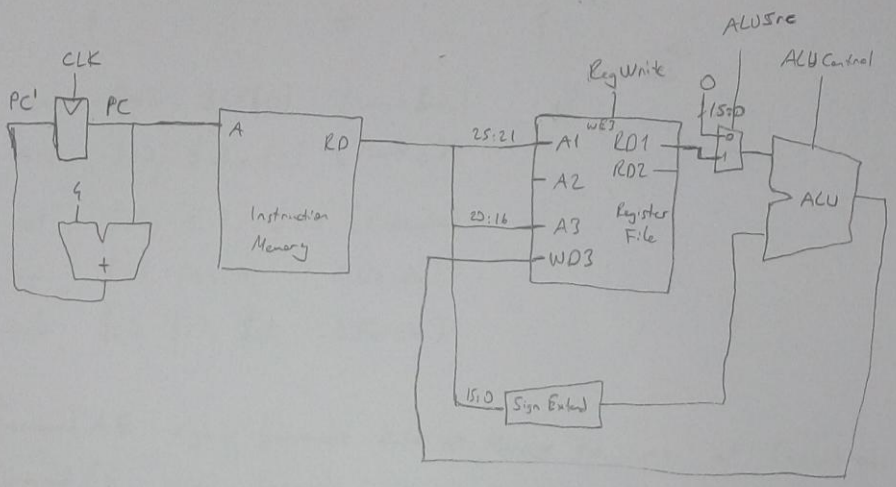
İbrahim Burak Tanrıkulu 21827852

Q1)

```
        addi  $t2, $0, 10
        addi  $t3, $0, 0      # i
loop:   beq   $t2, $t3, done  # i<10
        sll   $t4, $t3, 2     # byte offset
        add   $t5, $t4, $t0   # $t5 = A[i] address
        add   $t6, $t4, $t1   # $t6 = B[i] address
        lw    $s0, 0($t5)     # $s0 = A[i]
        lw    $s1, 0($t6)     # $s1 = B[i]
        slt   $t7, $s0, $s1
        beq   $t7, $0, skip
        sw    $s1, 0($t5)     # swap
        sw    $s0, 0($t6)     # swap
skip:   addi  $t3, $t3, 1     # i++
        j     loop
done:
```

İbrahim Burak Tanrıkulu 21827852

Q2)



	RegWrite	RegDst	ALUSrc	Branch	MemWrite	MemtoReg	ALUOp
lui	1	NA	0	NA	NA	NA	10
ori	1	NA	1	NA	NA	NA	10

İbrahim Burak Tarıkulu 21827852

Q3)

a)

MemWrite D	ALUSrc E	RegDst E	MemWrite M	RegWrite M
1	0	1	0	0

b)

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

ForwardAE signal forwards data at Memory to "rs" at Execution.
ForwardBE signal forwards data at Memory to "rt" at Execution.
We need a "rt" variable which will be determined at Memory.
I just translocated rs and rt variables.

c)

```
lw   $s0, 0($t0)   (WriteBack)
nop                (Memory)
add  $s1, $s0, $0  (Execution)
nop                (Decode)
nop                (Fetch)
```

d)

```
nop                (WriteBack)
add  $s0, $s1, $0  (Memory)
nop                (Execute)
beq  $s0, $s1, jump (Decode)
nop                (Fetch)
jump:
```

Ibrahim Burak Tanrıkulu 21827852

Q4)

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

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

Yes, total clock cycle reduced by 2 cycle.

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

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

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