Hacettepe University
Computer Engineering Department
BBM234 Computer Organization

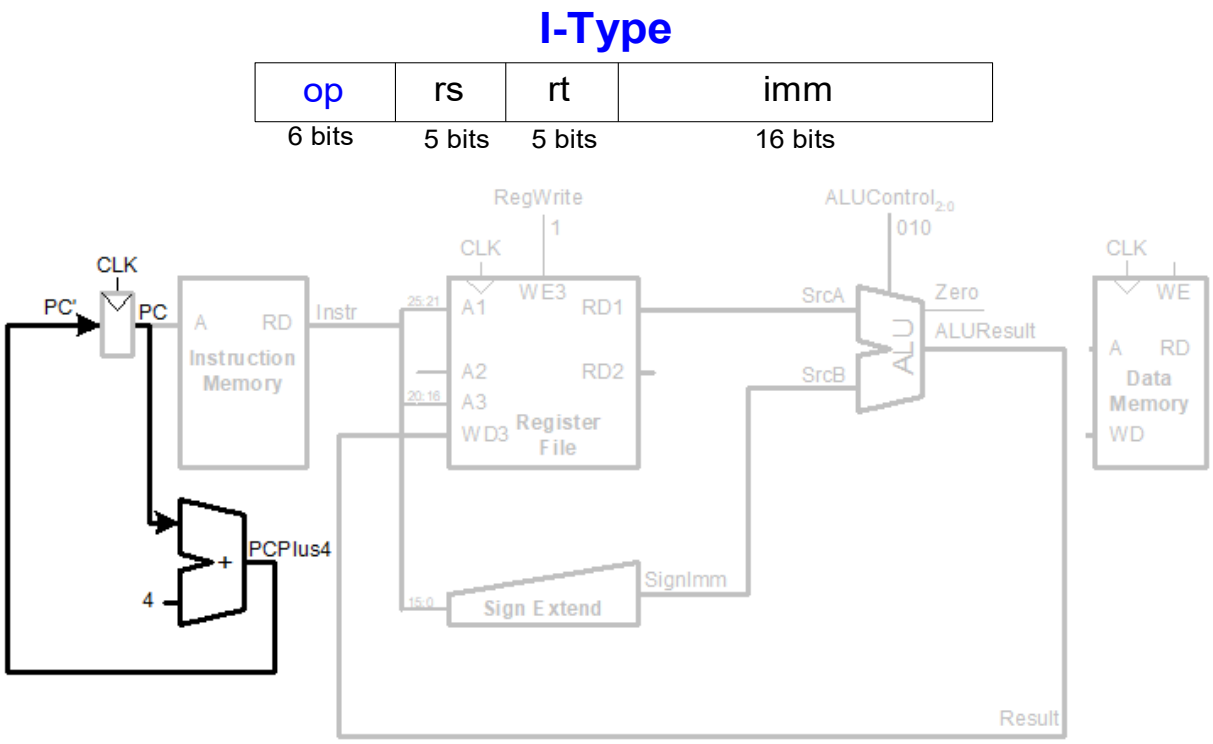
Homework 3

Assigned date : 19.04.2018
Due date : 26.04.2018 (All Sections)

You can bring your homework to the class. Or, you can give them to TA. Selma Dilek or slide under my office door. Don't email your homework!
LATE HOMEWORKS WILL NOT BE ACCEPTED…

Questions: (Each one is 20 points.)

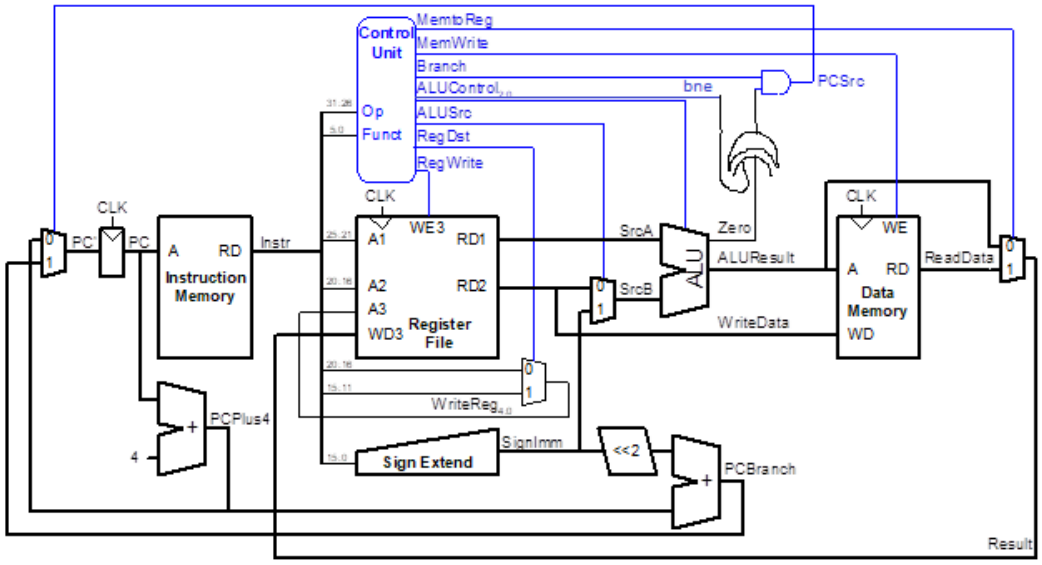
Q1.
You are given 32-bit Program Counter (PC), instruction memory, register file with 32 registers, and a data memory. First, show the machine code fields (instruction format) of the addi instruction. Then, draw the data-path of the addi for the single-cycle processor by using the following units and adding new hardware. Show your extra equipment and what bits are connected to which inpruts of the components.



Q2. We would like to add bne instruction to the single cycle architecture given below. bne instruction is a branch instruction and it loads branch target address (BTA) to the PC (PC=BTA) is [rs]!=[rt].

a) Show the necessary changes on data-path on Figure 1 and explain your changes. Your new architecture should be able to execute both beq and bne instructions.[12]

b) Fill the control signals in Table I. Add new control signal/signals to the table if necessary. [8]



a)

Şekil 1: Single cycle işlemci

We have to determine PCSrc signal function by using Branch, zero and newly added bne control signal. Truth table is below. If PCsrc = 1 then PC=BTA. The new control logic function is drawn above.

Branch	Zero	Bne	PCsrc
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

PCsrc=(Zero XOR bne) AND Branch

b)

Tablo I: **bne** komutunun kontrol sinyalleri

Inst.	**$Op_{31:26}$**	**RegWrite**	**RegDst**	**AluSrc**	**Branch**	**MemWrite**	**MemtoReg**	**$ALUOp_{1:0}$**	**Bne**
bne	000101	0	X	0	1	0	X	01 (Subtract)	1

Q3. You are given the following MIPS code. You have 5 stage pipelined MIPS processor running the code.

a) If there is no hazard unit in the MIPS processor, **insert enough NOPs between the instructions** to have correct execution of the code.
   i. How many cycles does it take to execute all instructions if the branch is not taken?
   ii. How many cycles does it take to execute all instructions if the branch is taken? Show your calculations or explain how you found the cycle time.

```
# MIPS assembly code
      lw  $s0, 0($0)

      lw  $s1, 4($0)
      2 NOPS
      beq $s0, $s1, L

      add $t0, $t0, $s0

      add $t1, $t1, $s1
      2 NOPS
      add $t2, $t0, $t1
      2 NOPS
L:    addi $t2, $t2, 1
```

If branch is not taken, after beq, all instructions execute. We have total of 13 instructions including NOPs. Then, we fetch 13 instrcutions in 13 cycles. Last instruction needs 4 more cycles to complete. Then, it takes 17 cycles to execute all instrcutions if branch is not taken.

If branch is taken, after beq, it takes 3 cycles to determine the next address of the branch. We have 4 instructions before beq, beq itself, 3 instcrutions after beq and the last instruction where branch is (L). Then, there will be 9 instructions that will execute. The total cycles will be 13.

b) If there is data hazard unit (data forwarding, stalling, and early branch resolution hardware) in the MIPS processor, insert enough NOPs between the instructions to have correct execution of the code.
   i. How many cycles does it take to execute all instructions if the branch is not taken?
   ii. How many cycles does it take to execute all instructions if the branch is taken? Show your calculations or explain how you found the cycle time.

```
# MIPS assembly code
      lw  $s0, 0($0)

      lw  $s1, 4($0)
      2 NOPs
      beq $s0, $s1, L

      add $t0, $t0, $s0

      add $t1, $t1, $s1

      add $t2, $t0, $t1
L:    addi $t2, $t2, 1
```

If branch is not taken, after beq, all instructions execute. We have total of 9 instructions including NOPs. Then, we fetch 9 instrcutions in 9 cycles. Last instruction needs 4 more cycles to complete. Then, it takes 13 cycles to execute all instrcutions if branch is not taken.

If branch is taken, after beq, it takes 1 cycles to determine the next address of the branch since we have early branch resolution hardware. We have 4 instructions before beq, beq itself, 1 instcrution after beq and the last instruction where branch is (L). Then, there will be 7 instructions that will execute. The total cycles will be 11.

Q4. You are given the following MIPS code. You have 5 stage pipelined MIPS processor running the code.

a) If there is no hazard unit in the MIPS processor, **insert enough nop's between the instructions** to have correct execution of the code. How many cycles does it take to execute all instructions? Show your calculations or explain how you found the cycle time.

 # MIPS assembly code

```
lw  $s0, 0($0)
lw  $s1, 4($0)
nop
nop  (the value of s1 is updated in fourth cycle. Since add uses s1, we need 2 nops)
add $t0, $s0, $s1
or $t1, $s2, $s3
nop
nop (the value of t1 is updated in fourth cycle. Since add uses s1, we need 2 nops)
and $t1, $t1, $t0
```

There are 9 instructions with nops. That takes 9 fetch cycles to fetch instructions. To finish the execution of last instruction, we need 4 more cycles. **The total of 13 cycles.**

b) If there is data hazard unit (data forwarding and stalling hardware) in the MIPS processor, insert enough nop's between the instructions to have correct execution of the code. How many cycles does it take to execute all instructions? Show your calculations or explain how you found the cycle time.

If we use data forwarding and stall in hazard unit, we need only one nop after lw instruction.

 # MIPS assembly code

```
lw  $s0, 0($0)
lw  $s1, 4($0)
nop (because of data forwarding, only one nop şs necessary)
add $t0, $s0, $s1
or $t1, $s2, $s3
and $t1, $t1, $t0
```

There are 6 instructions with nops. That takes 6 fetch cycles to fetch instructions. To finish the execution of last instruction, we need 4 more cycles. **The total of 10 cycles.**

c) If there is data hazard unit (data forwarding and stalling hardware) in the MIPS processor, rearrange the code if possible and your code still executes correctly. How many cycles does it take to execute all instructions? Show your calculations or explain how you found the cycle time.

Or instruction is independent of the previous instructions. We can move it up, after lw.

```
lw  $s0, 0($0)
lw  $s1, 4($0)
or $t1, $s2, $s3
add $t0, $s0, $s1
and $t1, $t1, $t0
```

There are 5 instructions. That takes 5 fetch cycles to fetch instructions. To finish the execution of last instruction, we need 4 more cycles. **The total of 9 cycles.**

Q5. The distcribution of the instcructions for the program with one billion (10^9) instructions is given below.

%40 load, %10 stores, %10 branch, %10 jump, %30 R-type

- Suppose
  - %50 load instruction results are used by the next instruction.
  - %50 R-type instruction results are used by the next instruction.
  - 50% branches are taken.
  - All jumps flush the next instruction.

a) How many clock cycles does this program take in a single cycle MIPS processor?

CPU_Time(SC)= #_of_instructions.CPI.Clock_Period = 10^9.1. Clock=10^9 CC

b) How many clock cycles does it take on a pipelined MIPS processor with no hazard unit (no forwarding and no early branch resolution)?

If there is no forwarding, the dependent instructions will take 3 cc.

CPI(lw)=0.50.1+0.50.3=2
CPI(R)==0.50.1+0.50.3=2

If the branch taken, 3 instructions will be flushed. Branch will take 4 instruction (branch instruction plus three flushed ones). Otherwise, it will take only one cc.

CPI(branch)=0.50.1+0.50.4=2.5

CPI(J)=2

CPI(avr)= 0.40.2+0.10.1+0.10.2.5+0.10.2+0.30.2=1.95

CPU_Time(Pipe_no_hazard_unit)=10^9.1.95.cc=1.95. 10^9

c) How many clock cycles does it take on a pipelined MIPS processor with hazard unit (with forwarding and early branch resolution)?

If there is forwarding and early branch resolution, the dependent instructions will take 2 cc for lw.

CPI(lw)=0.50.1+0.50.2=1.5
CPI(R)==0.50.1+0.50.1=1

If the branch taken, 1 instructions will be flushed. Branch will take 2 instruction (branch instruction plus one flushed instruction). Otherwise, it will take only one cc.

CPI(branch)=0.50.1+0.50.2=1.5

CPI(J)=2

CPI(avr)= 0.40.1.5+0.10.1+0.10.1.5+0.10.2+0.30.1=1.95

CPU_Time(Pipe_no_hazard_unit)=10^9.1.35.cc=1.35. 10^9