İTÜ Computer Engineering Department **Number:**
12.03.2015 **Name Surname:**
Time: 1 hour 45 minutes **Signature:**

COMPUTER ARCHITECTURE 1ST MIDTERM

QUESTION 1: (45 Points)
A CPU has an instruction pipeline with the following 4 segments:
1. FD (*Fetch and decode Instruction)*
2. FO (*Fetch Operand*), Operands of the operations or the relative address of the branch are fetched.
3. EX (*Execution*): The operation is performed or the target address of the branch is calculated (the result has not written yet)
4. WR (*Write the result*): Result of the execution or the target address is written to the destination (registers, flags, memory)
Hardware-based mechanisms are used to solve branch hazards.
Answer the questions (a-d) according to the given piece of code below.

```
Line number:           Instructions:
1:                     Counter ← initial_value
2:       LOOP:         ------                  ; Any instruction
3:                     ------                  ; Any instruction
4:                     Counter ← Counter - 1
5:                     BNZ   LOOP              ; Branch if not zero
6:                     ------                  ; Instruction after the loop
```

a) The initial value of the counter is 2. The loop runs two times and then the instruction at line 6 is completed. Assume that that the time starts at line 2, where the first instruction of the loop is fetched. Draw the space-time diagram of the pipeline that shows the operations that start with the instruction at line 2, until the completion of the instruction at line 6 if the branch prediction mechanism is
i) static prediction "Always predict not taken"
ii) dynamic prediction with one bit, and the initial decision is to take the branch.

b) Show the branch penalties on your diagrams. How many clock cycles is the branch penalty in this system in case of misprediction?

c) Assume that the initial value of the counter is 10 and dynamic prediction mechanism with one bit is used, and the initial decision is to take the branch. If time starts at line 2, how many clock cycles does it take until the completion of the instruction after the loop at line 6? Do not draw the time-space diagram.

d) The clock cycle of the pipeline is 20 ns and without the pipeline, one machine-language instruction would have been completed in 60 ns. What is the speedup achieved in c)?

e) What is the maximum speedup that can be achieved by this pipeline (without any hazards)?

QUESTION 2: (30 Points)
A RISC CPU has an instruction pipeline with 3 stages: F: Fetch and decode instruction
A: Operand (register) read, ALU operation, and write the result of ALU to the register
M: Memory access if necessary (from register to memory or from memory to register)

a) Examine the given program by drawing the timing diagram of the given pipeline and show the problems. Solve all data and branch problems by inserting NOP instructions.

```
       ...
       LD     0(R10), R1          ; R1 <-- M[R10]
       LD     0(R11), R2          ; R2 <-- M[R11]
       ADD    R1, R2, R3          ; R3 <-- R1+R2
       ADD    R0, R3, R4          ; R4 <-- R3
       ST     0(R12), R3          ; M[R12]<-- R3
       BA     L2                  ; Branch always
L1:    ADD    R0, 0, R4           ; R4 <-- 0
       ...
L2:    ADD    R0, 0, R3           ; R3 <-- 0
```

b) Give the software-based solutions to the branch problem without decreasing the performance of the pipeline. Do not change the algorithm.

INSTRUCTION SET:

LD	X(Rs), Rd	*Rd ← M[Rs + X]*	*Load*
ST	X(Rs), Rd	*M[Rs + X] ← Rd*	*Store*
ADD	Ri,Rj,Rd	*Rd ← Ri + Rj*	
BA	Y	*PC←PC + Y*	*Branch always (relative)*

QUESTION 3: (25 Points)

Write hardware-based and software (compiler)-based solutions that can be used to solve different problems in instruction pipelines into the given tables below. Write the name of only one solution to the proper cell of the table if such a solution exist. If such a solution does not exist write “NONE”.

Solution:

a) Solutions to the operand dependency:

	Hardware-based	Software(compiler)-based
Without degrading the performance		
Degrading the performance		

b) Solutions to the branch hazards:

	Hardware-based	Software(compiler)-based
Without degrading the performance		
Degrading the performance		

c) Explain only the hardware-based solutions shortly.