Burak Yesil
“I pledge my honor that I have abided by the Stevens Honor System.”

Instruction Types:
R-Type Instructions (ADD, SUB, AND, ORR)
LDUR
STUR
CBZ
B

Problem #1:

A. CBZ, B
B. LDUR, STUR
C. LDUR, R-Type Instructions

Problem #2:

A. You need **two** additions: A **second write port** and a **second ALU** in order to read and write at the same time.

Problem #3:

A. **With** ALU: CT = 200 + 250 + 150 + 300 + 200 + **200** = **1300ps**
Without ALU: CT = 200 + 250 + 150 + 300 + 200 = **1100ps**

B. Speed Up: (Old CPU Time) / (New CPU Time) = (1*1*1100)/(.8*1*1300) = **1.0576923 = 5.7 percent faster**
C. 1/(new time) > 1/(old time) -> 1/(.8*1*(1100+**X**) > 1/(1*1*1100)
1100 > .8(1100 + **X**)
(1100/.8) - 1100 > **X**
275 PS > X

Problem #4:

A.)

- In a pipelined processor, the cycle time is determined by the longest stage, so **(300 PS)**.
- In a non-pipelined processor, the cycle time is determined by completing all of the stages so, 200 + 250 + 150 + 300 + 200 = **(1100 PS)**.

B.)

- Pipelined: (LDUR uses all 5 stages)
    - (5 cycles) * (300 PS) = **1500 PS**
- Non-pipelined: (you have to complete the full set)

- 200 + 250 + 150 + 300 + 200 = **1100 PS**

C.)

We would split the MEM stage in half because it takes the longest. So it will be in the following format:

Original:	IF (200)	ID (250)	EX (150)	MEM (300)	WB (200)	
New:	IF (200)	ID (250)	EX (150)	MEM1 (150)	MEM2 (150)	WB (200)

New Clock Time:
(pipelined) = **250 PS**
(Non-pipelined) = **1100 PS**

Problem #5:

	1	2	3	4	5	6	7	8	9
LDUR X20, [X19, #0]	IF	ID	EX	MEM	WB				
LDUR X21, [X19, #8]		IF	ID	EX	MEM	WB			
NOP					EX	MEM	WB		
ADD X22, X21, X20				IF	ID	EX	MEM	WB	
SUB X23, X23, X22					IF	ID	EX	MEM	WB

- The X21 register in the MEM stage in the **(LDUR X21, [X19, #8])** instruction gets forwarded to the EX stage of the **(ADD 22, X21, X20)** instruction.
- The value of X22 in the EX stage in the **(ADD X22, X21, X20)** instruction gets forwarded to the EX stage of the **(SUB X23, X23, X22)** instruction.

Problem #6:

a.)

```
LDUR X1, [X6, #8]
NOP
NOP
ADD X0, X1, X0
NOP
NOP
STUR X0, [X10, #4]
LDUR X2, [X6, #12]
NOP
NOP
SUB X3, X0, X2
NOP
NOP
STUR X3, [X8, #24]
CBZ X2, 40
```

b.)

```
LDUR X1, [X6, #8]
LDUR X2, [X6, #12]
ADD X0, X1, X0
SUB X3, X0, X2
STUR X0, [X10, #4]
STUR X3, [X8, #24]
CBZ X2, 40
```