

- 1) (7/40) In the following fragment of code, let's identify: i) true dependencies; ii) anti-dependencies; iii) output dependencies; iv) use renaming technique to prevent the problems that are caused by dependencies.

```

LW      R1, 0(R7)
ADD     R1, R2, R4
SUBI    R2, R4, 25
ADD     R4, R1, R3
ADDI    R1, R1, 30

```

- 2) (16/40) In the following sequence of execution we have in-order dispatch and in-order complete for a two-way superscalar processor; answer the following questions:
- identify the most probable reason why the instruction I2 cannot enter into Issue+Execute+Writeback before cycle 4;
 - would out-of-order complete and/or dispatch permit to avoid the delay of I2 in the decode stage (explain why yes or why not)
 - identify the most probable reason why the instruction I6 cannot enter into the commit stage before cycle 9;
 - would the out-of-order complete and/or dispatch permit to avoid the dispatch-delay or complete-delay of I6 (explain why yes or why not)

Decode		(Dispatch)	Issue+Execute+Writeback			(Complete)	Commit		Cycle
Slot1	Slot2		FU1	FU2	FU3		Slot1	Slot2	
I1	I2								1
	I2				I1				2
	I2				I1				3
I3	I4			I2					4
I5	I6			I4	I3		I1	I2	5
					I3				6
			I5	I6			I3	I4	7
			I5						8
							I5	I6	9

- 3) (17/40) Given the sequence P1: R, P2: R, P1: W, P2: W, P1: W, P2: W (Px:R = read by the processor Px, Px:W write by the processor Px), with respect to a certain variable 'a', show for each processor the sequence of states, and transactions on the bus that occur in a multiprocessor UMA with write-back cache of each processor and in the case of the coherence protocol Dragon and in the case of the coherence protocol MESI. When the transactions costs are Cbusrd=Cbusrdx=150, Cbusupg=40, Cbusupd=20, Cflush=Cflush*=20, what is the total cost in case of Dragon and in case of MESI ?