ⓐ

DEPENDINȚE

INSTRUCȚIUNE	RAW	WAR	WAW	CONTROL
i1) ET1: LW $s2, 0($s1)	(s2) i1 → i2 (s2) i1 → i3 (s2) i1 → i4	—	(s2) i1 → i2	—
i2) OR $s2, $s3, $s2	(s2) i2 → i3 (s2) i2 → i4	(s3) i2 → i4	—	—
i3) BEQ $s2, $s0, ET2	—	(s2) i3 → i6	—	(ET2) i3 → i4, i5, i6
i4) LW $s3, 0($s2)	i4 → i6 (s3)	(s2) i4 → i6 (s2) i4 → i7	—	—
i5) J ET1	—	—	—	(ET1) i5 → i6, i7
i6) ET2: ADD $s2, $s3, $s2	i6 → i7 (s2)	~~——~~ —	~~——~~ —	—
i7) SW $s2, 0($1)	—	—	—	—

Doar cele RAW și de control pot deveni hazarduri!

i1: IF→ID→EX→M→WB

i2: IF→ID→EX→M→WB

i3: IF→ID→EX→M→WB

i4: IF→ID→EX→M→WB

i5: IF→ID→EX→M→WB

i6: IF→ID→EX→M→WB

i7: IF→ID→EX→M→WB

RAW, RAW, RAW, CTRL, RAW, CTRL, CTRL, RAW, CTRL, RAW

Nu este hazard pentru că WB scrie pe I. pont și iD citește pe al II-lea
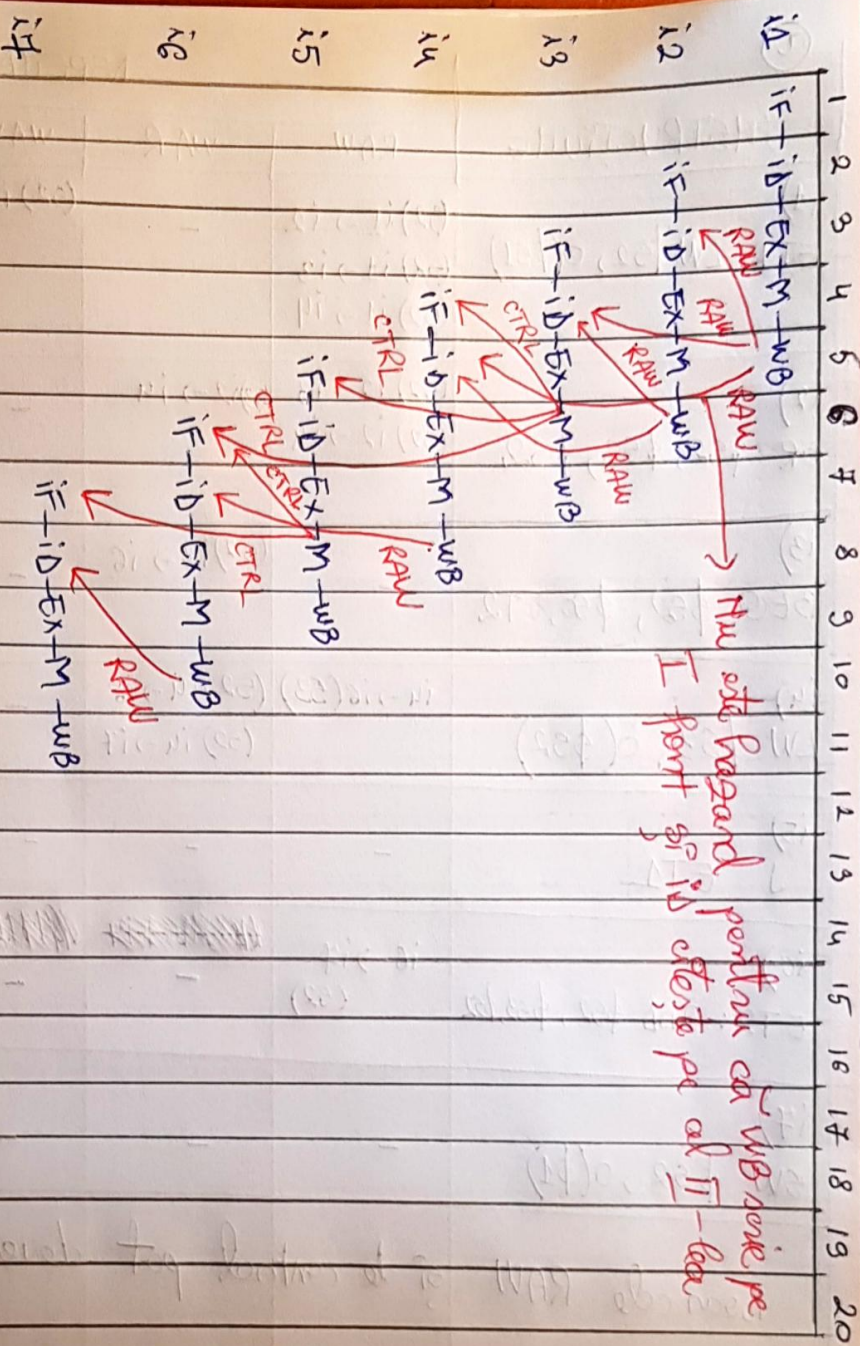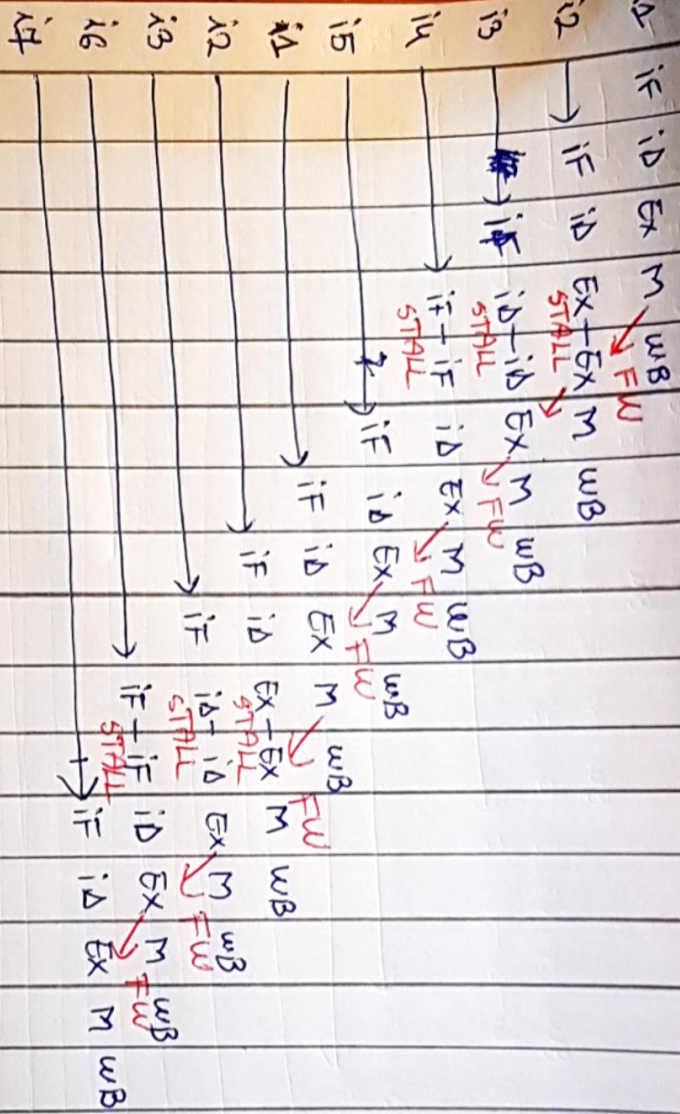
=> 11 cicluri de ceas

=> **6** hazarduri RAW și 2 de control.

- RAW apar pentru că a doua instrucțiune citește sursa înainte ca prima să o scrie anterior

- cele de CTRL apar pentru că, până la evaluarea condiției de salt, se introduc în bandă de asamblare, ce este posibil sau nu să fie execuate în funcție de rezultatul evaluării

Pipeline hazard diagram (MIPS — IF/ID/EX/M/WB stages).

Cycle numbers (columns): 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 ... 29 30 34

Instruction rows (left labels, top to bottom): i1, NOP, NOP, i2, NOP, i3, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, NOP, i3, NOP, NOP, i4

instr	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	34
i1	IF	ID	EX	M	WB																										
NOP		IF	ID	EX	M	WB																									
NOP			IF	ID	EX	M	WB																								
i2				IF	ID	EX	M	WB																							
NOP					IF	ID	EX	M	WB																						
i3						IF	ID	EX	M	WB																					
NOP							IF	ID	EX	M	WB																				
NOP								IF	ID	EX	M	WB																			
NOP									IF	ID	EX	M	WB																		
NOP										IF	ID	EX	M	WB																	
NOP											IF	ID	EX	M	WB																
NOP												IF	ID	EX	M	WB															
NOP													IF	ID	EX	M	WB														
NOP														IF	ID	EX	M	WB													
NOP															IF	ID	EX	M	WB												
NOP																IF	ID	EX	M	WB											
NOP																	IF	ID	EX	M	WB										
NOP																		IF	ID	EX	M	WB									
NOP																			IF	ID	EX	M	WB								
NOP																				IF	ID	EX	M	WB							
NOP																					IF	ID	EX	M	WB						
NOP																						IF	ID	EX	M	WB					
NOP																							IF	ID	EX	M	WB				
i3																								IF	ID	EX	M	WB			
NOP																									IF	ID	EX	M	WB		
NOP																										IF	ID	EX	M	WB	
i4																											IF	ID	EX	M	WB

3) Cicluri de ceas (eliminarea hazardurilor RAW și a celor de control prin introducerea de NOP-URI)

(c)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

i1: IF ID EX M WB

i2: →IF ID→ID→ID EX M WB

i3: →IF→IF→IF ID→ID→ID EX M WB

i4: →IF IF IF ID EX M WB

i5: IF IF IF IF ID→ID→ID EX M WB

i1̃ : IF IF IF IF IF ID EX M WB

i2: IF IF IF ID→ID→ID EX M WB

i1: IF ID EX M WB

i3: IF→IF→IF ID→ID→ID EX M WB

i4: IF IF ID EX M WB (galben i4) → IF ID EX

i6: M WB → IF ID

i4: 30 31 32 33 → M WB → IF ID

i7: IF ID EX M WB

32 coloane de ceas prin utilizarea de STALL-uri (alegem cost de
hardware de STALL)

Pipeline hazard diagram (columns numbered 1–28):

- i1: IF ID EX M WB — FW
- i2: →IF ID EX→EX M WB — STALL
- i3: →if→if ID→ID EX M WB — STALL, FW
- i4: →if→if ID EX M WB — STALL, FW
- i5: →if ID EX M WB — FW
- i4: IF ID EX M WB — FW
- i5: IF ID EX M WB
- i1: if→if ID EX→EX M WB — STALL, FW
- i2: →if→if ID→ID EX M WB — STALL, FW
- i3: →if ID EX M WB — FW
- i6: →if ID EX M WB
- i7: →if ID EX M WB

FORWARDING + STALL + URI + DETECT.
HAZARDURI
⇒ 16 cicluri de caas

② Semnal 1 : ALUsrc și Branch ⟶ generate la ID (unitatea de control) și
  Semnal 2 : PCSrc { și RegWrite ⟶ ocié într-un registru utilizate în ALU (Ex)
                    { și ReWrite ⟶ ocié într-un registru
       ↳ controlează a doua intrare pt. ALU

            ↳ unitatea de control
                 ↳ generat la Ex și utilizat la Memory.