

TEST 8 CURS CN1

SUBIECT MIPS ÎN BANDĂ DE ASMABLARE

Folosind cunoștințele predate online la cursurile 7.1, 11.1, 11.2 și Curs practic – Micro-arhitectura și performanța procesorului didactic MIPS postate pe cursul CN1 de pe Moodle și prezentate live prin Curs 11 video MIPS Pipeline, pe baza schemei procesorului MIPS de la pagina 98/136 din Cursul practic și executând pe procesor următoarea secvență de instrucțiuni:

```
ADD    $s7, $s6, $s5
LW     $s1, 0($s1)
AND    $s1, $s1, $s2
ET1:   LW     $s2, 0($s1)
BEQ    $s2, $s0, ET1    ; se presupune că BEQ se execută o dată (True) apoi nu (Not True)
OR     $s2, $s2, $s3
SW     $s2, 0($s3)
```

Se cer următoarele:

- Găsiți toate dependențele din această secvență de instrucțiuni specificând tipul hazardurilor găsite (structural, RAW, WAR, WAW sau de control) explicând pe scurt și din ce cauză apar.
- Dacă **nu există** hardware de Forwarding sau pentru detecție hazarduri, să se corecteze execuția doar prin inserarea de NOP-uri pentru eliminarea hazardurilor de la punctul a).
- Repetăți punctul b) dar utilizând NOP-uri doar când un hazard nu poate fi evitat prin schimbarea sau rearanjarea acelor instrucțiuni. Puteți presupune că registrul \$t0 poate fi utilizat pentru a păstra valori temporare în codul vostru modificat.
- Să se repete punctul b) utilizând STALL-uri în loc de NOP-uri.
- Dacă **există** hardware de Forwarding sau pentru detecție hazarduri, să se corecteze execuția precizând când și cum are loc Forwarding. Atunci când nu se poate realiza doar cu Forwarding, se pot introduce și STALL-uri și/sau NOP-uri și/sau se poate rearanja codul dacă este posibil.
- Dacă se întrerupe legătura ALUOutM de la intrarea multiplexorului de Forwarding legat la RD2 din schema procesorului MIPS de la pagina 98/136, ce se întâmplă cu execuția de la punctul e)? Propuneți o metodă de corecție în caz de apariția unui hazard datorat acestui lucru în secvența corectată la punctul e).

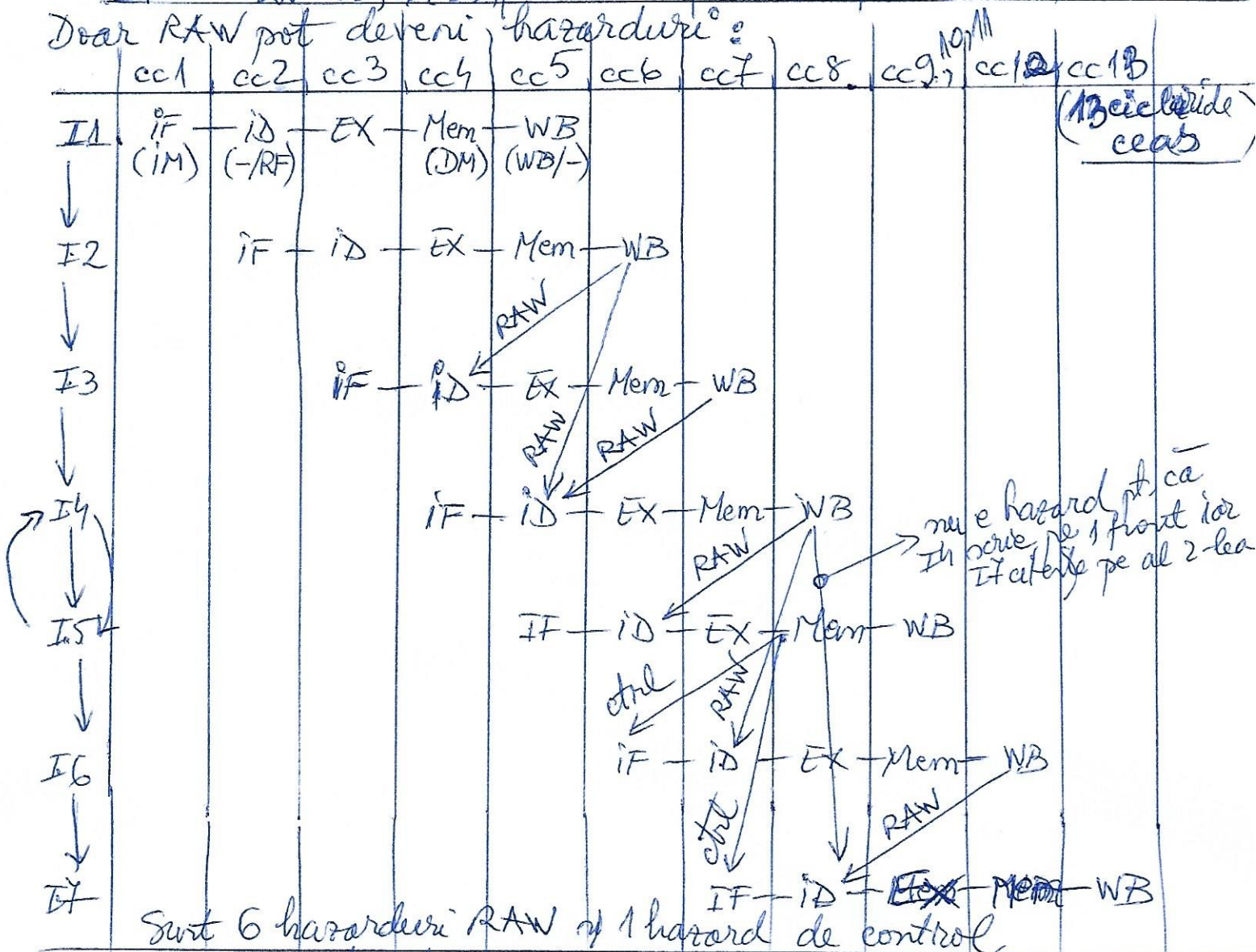
NOTARE: 1,5 puncte pentru toate subpunctele cerute plus 1 punct din oficiu.

REZOLVARE PROBLEMA MIPS

a)

	Secvența instrucțiunilor	Dependențe instr.			
		RAW	WAR	WAW	Control
I1)	ADD s7, s6, s5	—	—	—	—
I2)	LW s1, 0(s1)	(s1) I2 → I3 (s1) I2 → I4	(s1) I2 → I3	(s1) I2 → I3	—
I3)	AND s1, s1, s2	(s1) I3 → I4	(s2) I3 → I4 (s2) I3 → I6	—	—
I4)	ET1: LW s2, 0(s7)	(s2) I4 → I5 (s2) I4 → I6 (s2) I4 → I7	—	(s2) I4 → I6	—
I5)	BEQ s2, s0, ET1	—	(s2) I5 → I6 (s2) I5 → I4	(s2) I5 → I4	(ET1) I5 → I6, I7
I6) I7	OR s2, s2, s3 SUB s2, 0(s3)	(s2) I6 → I7	—	—	—

Doar RAW pot deveni hazarduri:



IM - memoria de instr.

DM - memoria de date

/-RF

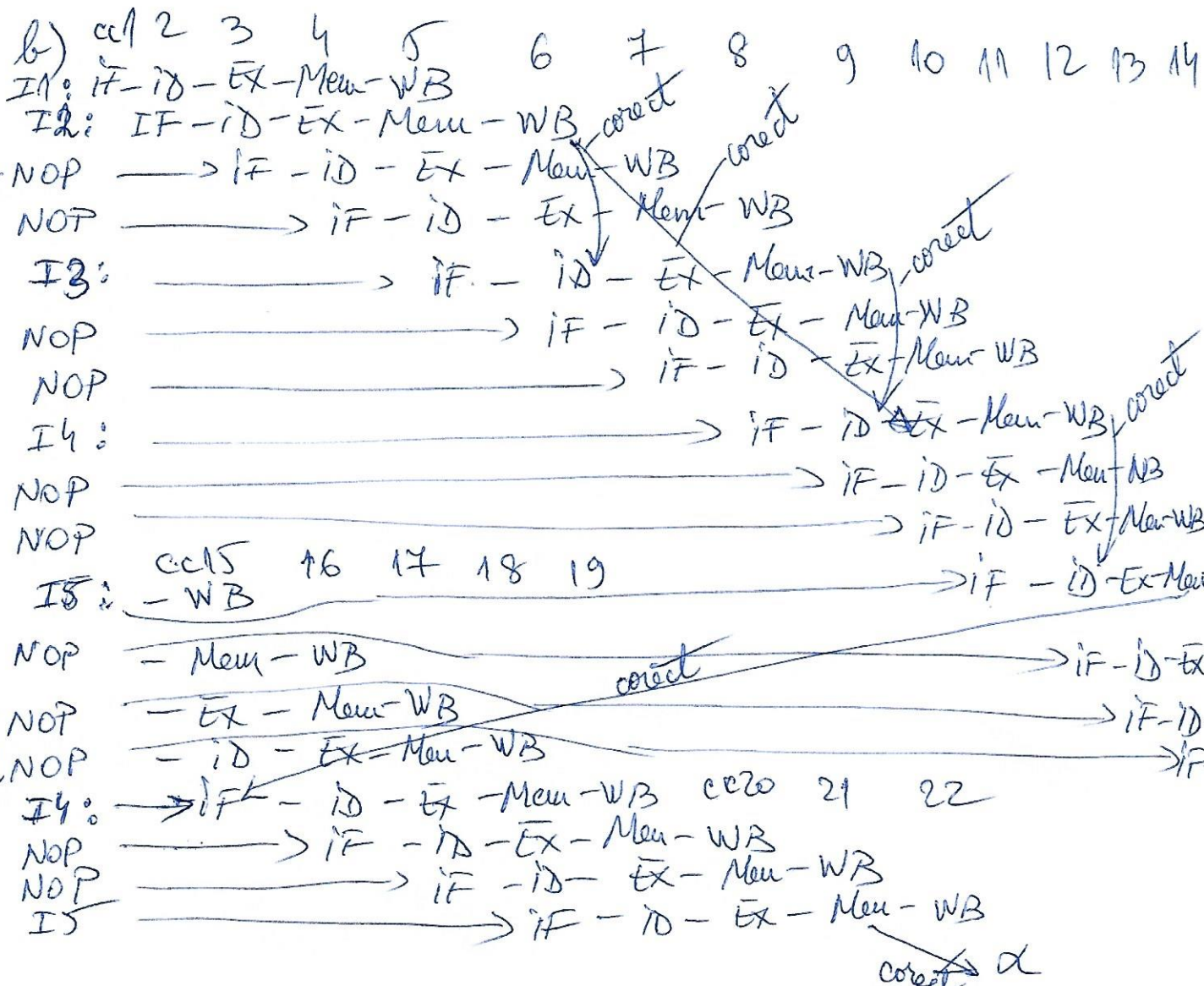
ID citește pe al 2-lea front

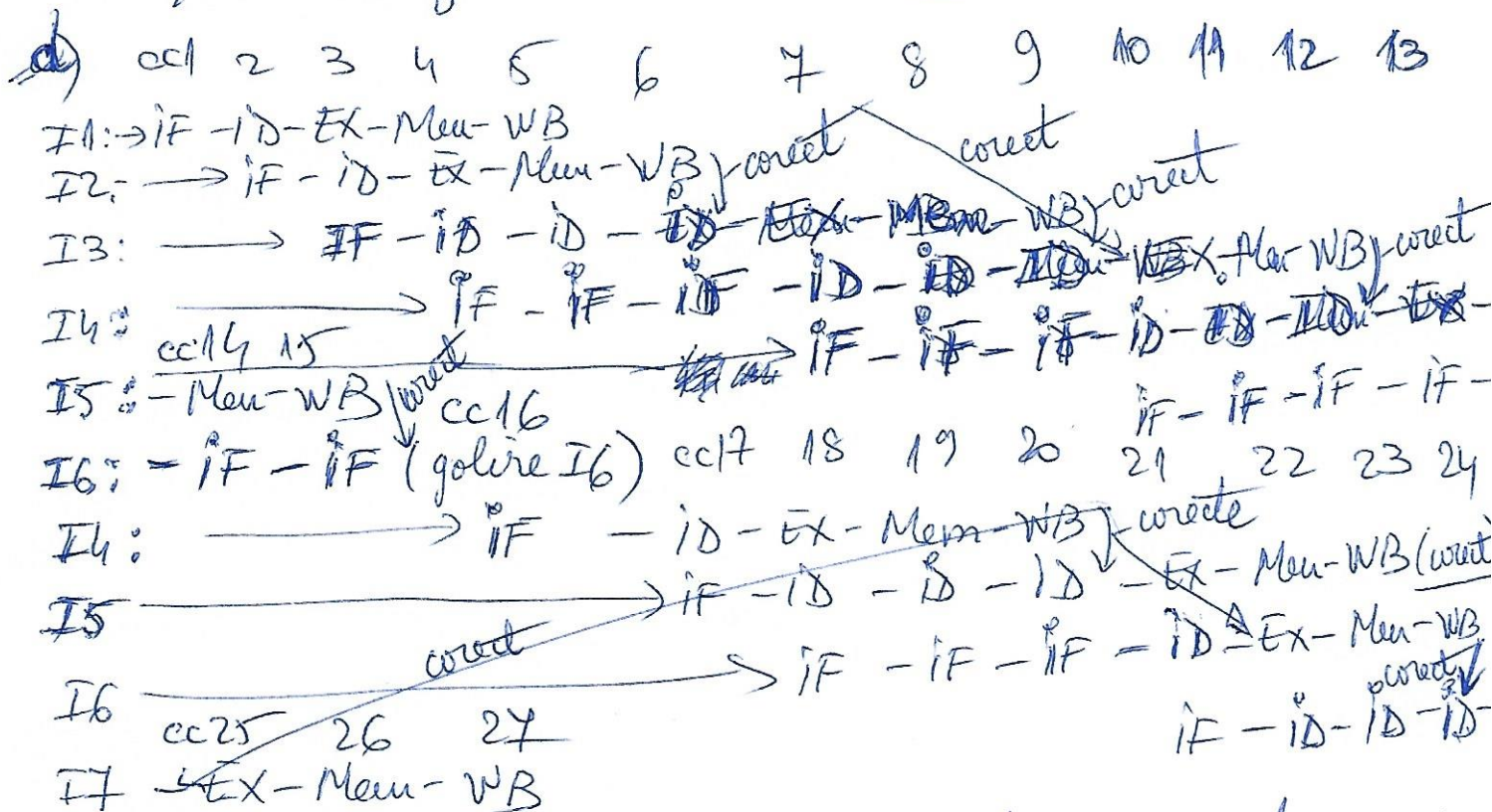
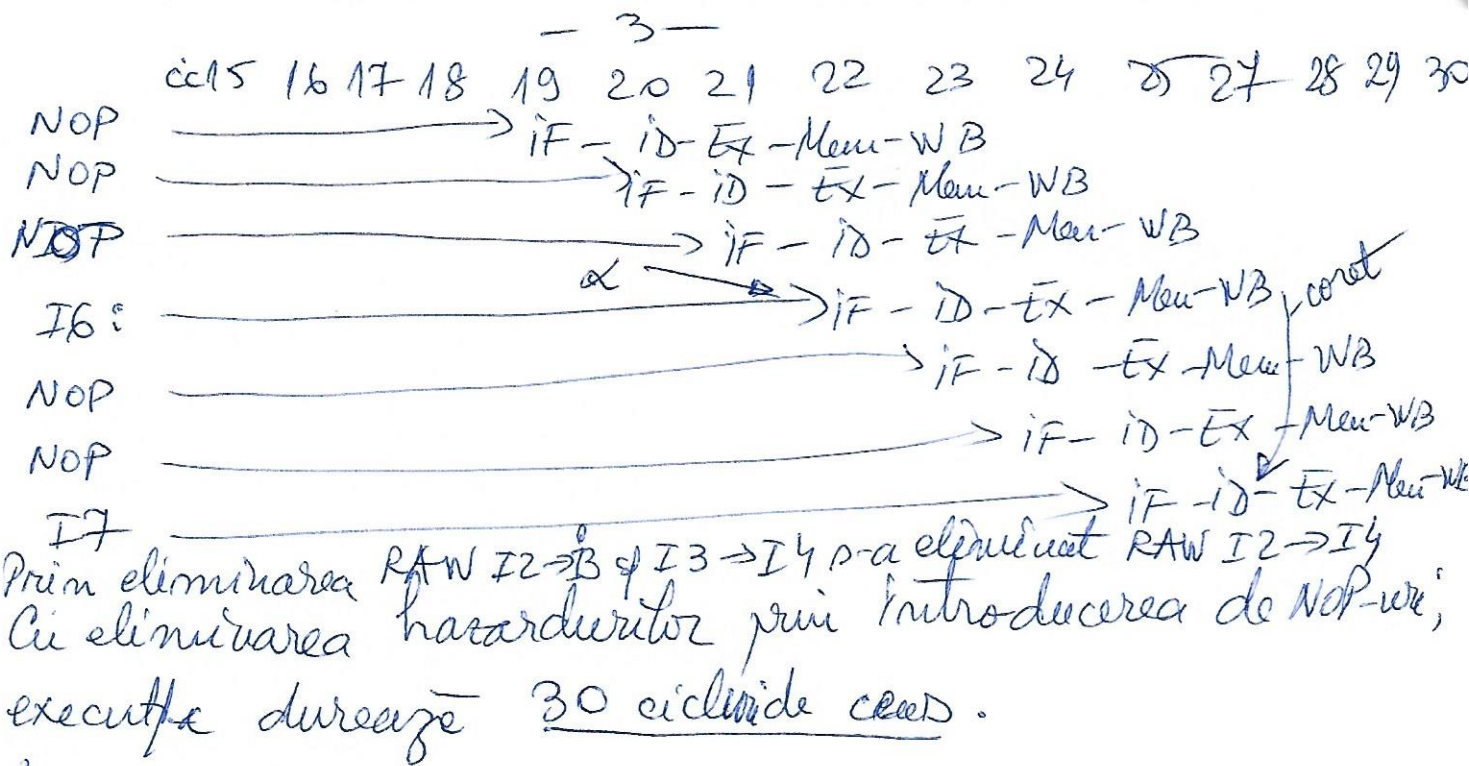
WB/-

WB scrie pe 1 front

Hazardurile RAW apar pentru că a doua instrucțiune citește sursoa înainte ca prima să o scrie anterior. (nu se respectă dependența RAW existentă).

Hazardul de control apare pentru că, pînă la
~~calcularea adresei de salt~~
evaluarea condiției de salt executat sau neexecutat,
se introduce în banda de asamblare încă 2 instr.
care, pentru că s-a dat că BEA va fi prima sarcină
executată, trebuie să fie golite din banda înainte
de a se continua cu IH de la adresa saltului
condiționat.



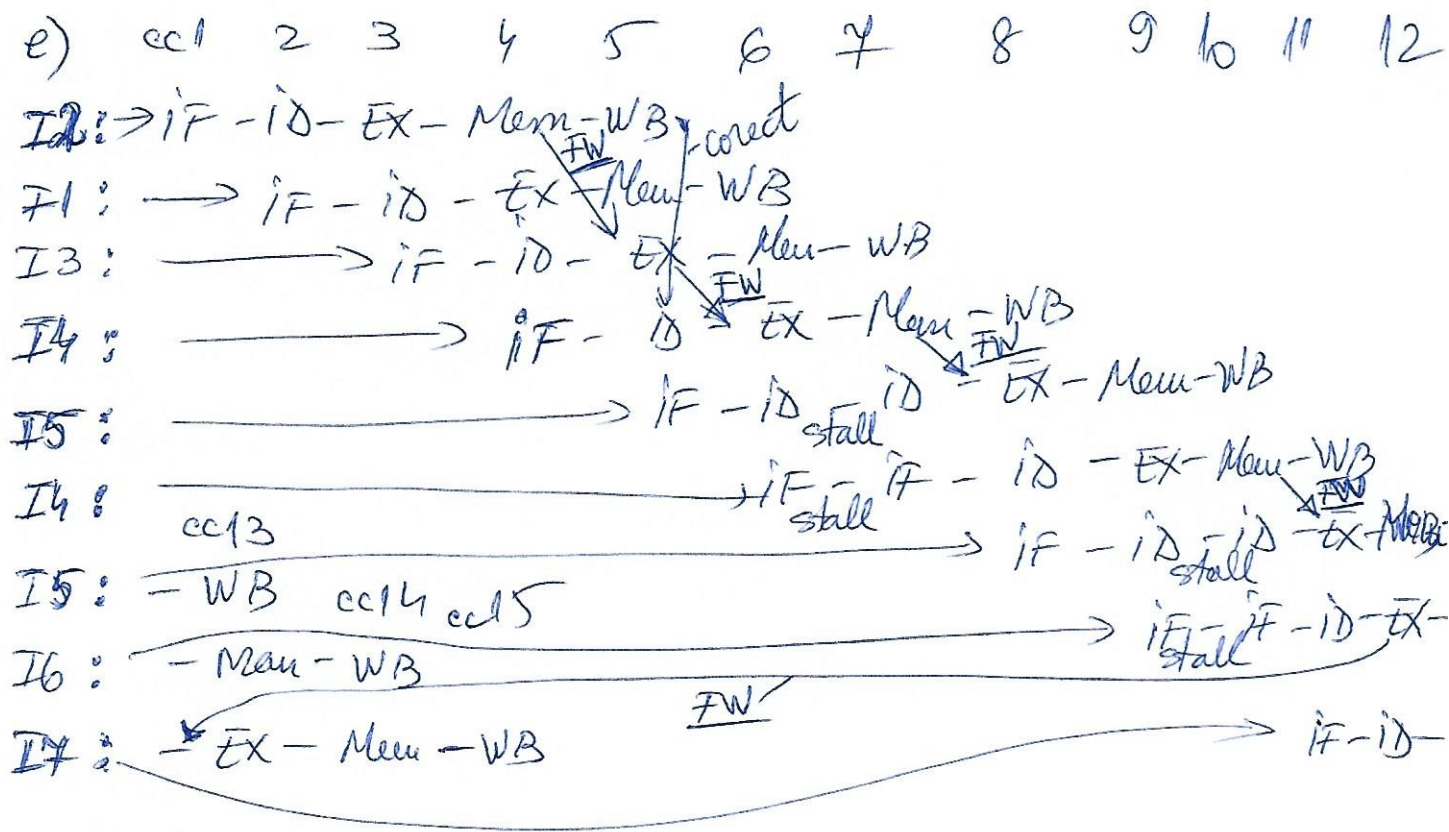


Cu eliminarea hazardurilor prin introducerea de STALL,
 execuția durează 27 cicluri de ceas (mai puțin ca la b
 dar cu un cost hardware de STALL).

c) Se urcă al doilea I4 în loc de NOP-ul β și se pot
 NOP-urile γ apoi se mută I1 în loc de NOP-ul θ.
 Execuția va dura acum 26 cicluri de ceas.

Se poate face același lucru și la d) câștigând
 tot 4 cicluri de ceas.

Se mai poate și combina NOP-uri cu STALL-uri
 dacă se mai câștigă cicluri de ceas.



Cu hardware de accelerare și stall-uri, execuția durează 15 cicluri de ceas.

f) Întreruperea legăturii de FW ALUOutM pentru mersa RD2 conduce la incapacitatea de a mai accelera s2 executat de OR la SW pentru a scrie la DM. Înseamnă că, în loc de accelerare, va trebui să folosim NOP-uri sau STALL-uri.

Execuția va dura cu 2 cicluri de ceas mai mult. Am putea elimina cele 2 cicluri de ceas suplimentare dacă am introduce și un hardware de accelerare de la ResultW, printr-un MUX 2:1 la intrarea WD a memoriei DM.