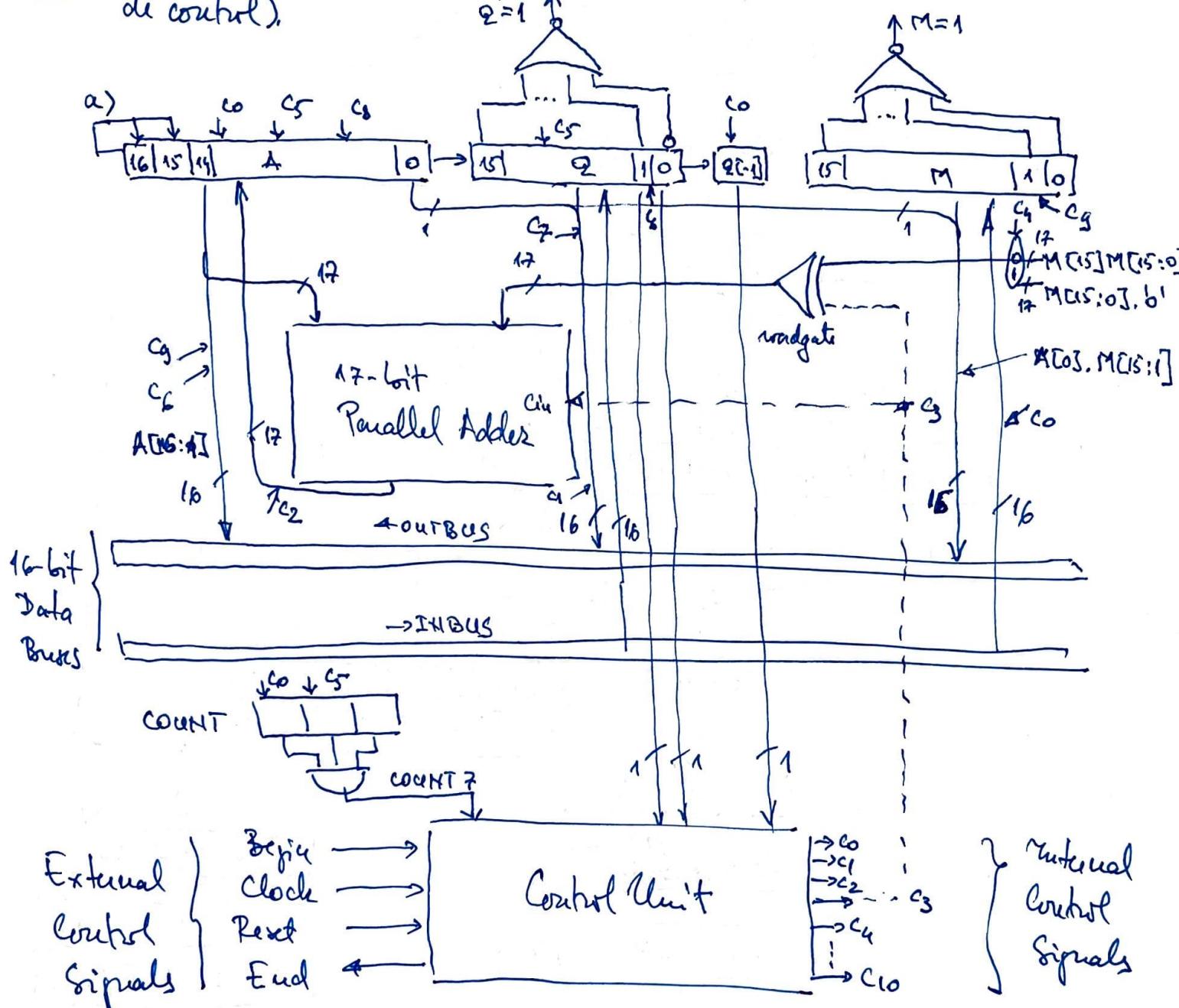
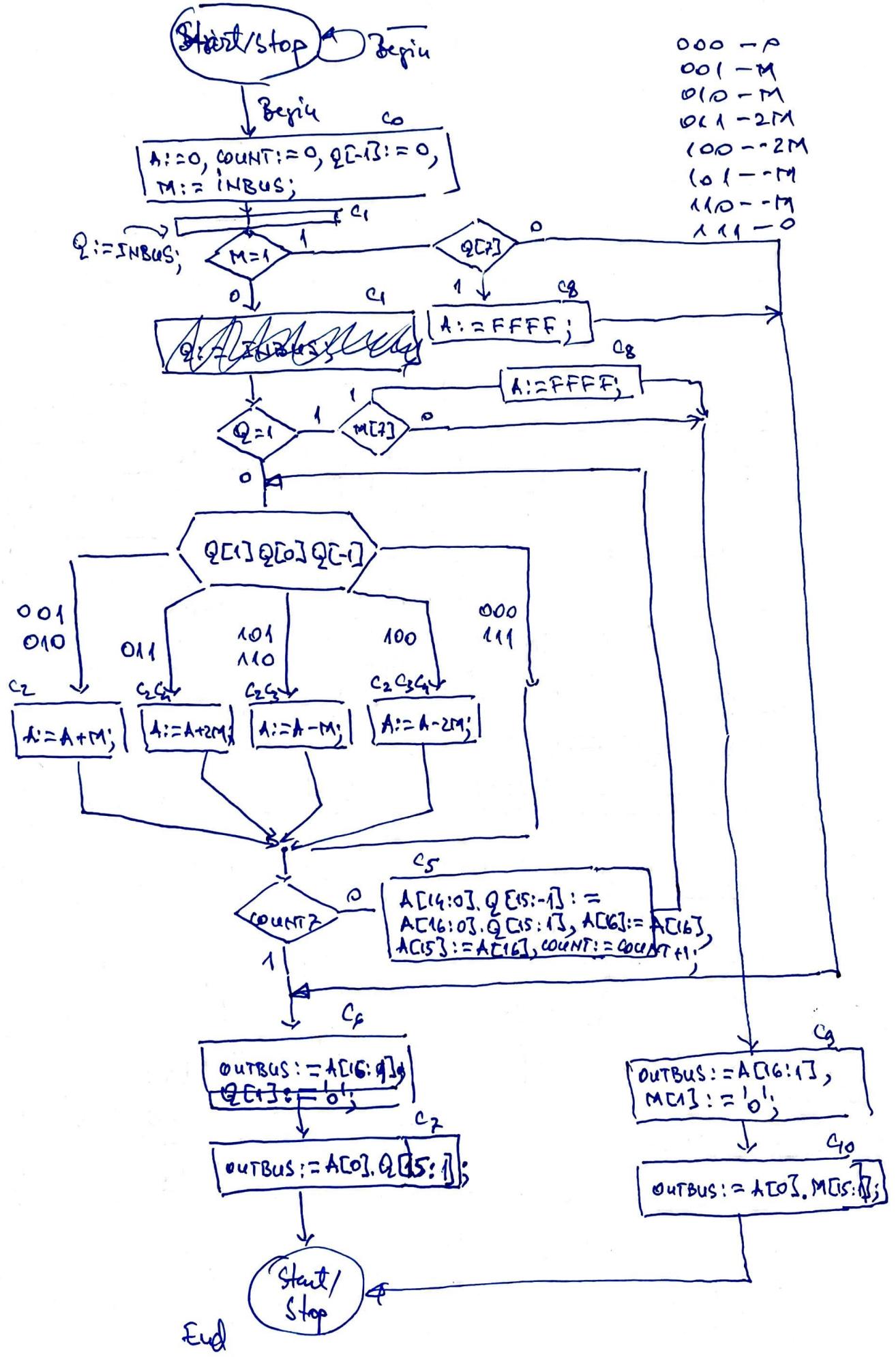


① Să se modifice dispozitivul de înmulțire Booth radix-4, astfel încât acesta să înmulțească numere fraționare și să trateze în mod distinct situațiile de înmulțire cu zero. (Rezultatul este egal cu celălalt operand, exprimat pe 32 biti.) Se cere:

- a) Schema bloc hardware pe platforma ~~baseboard~~ Hayes, (cu semnale de ctrl.)
- b) Ordinograma ce descrie funcționarea unității de control (cu semnalele de control).





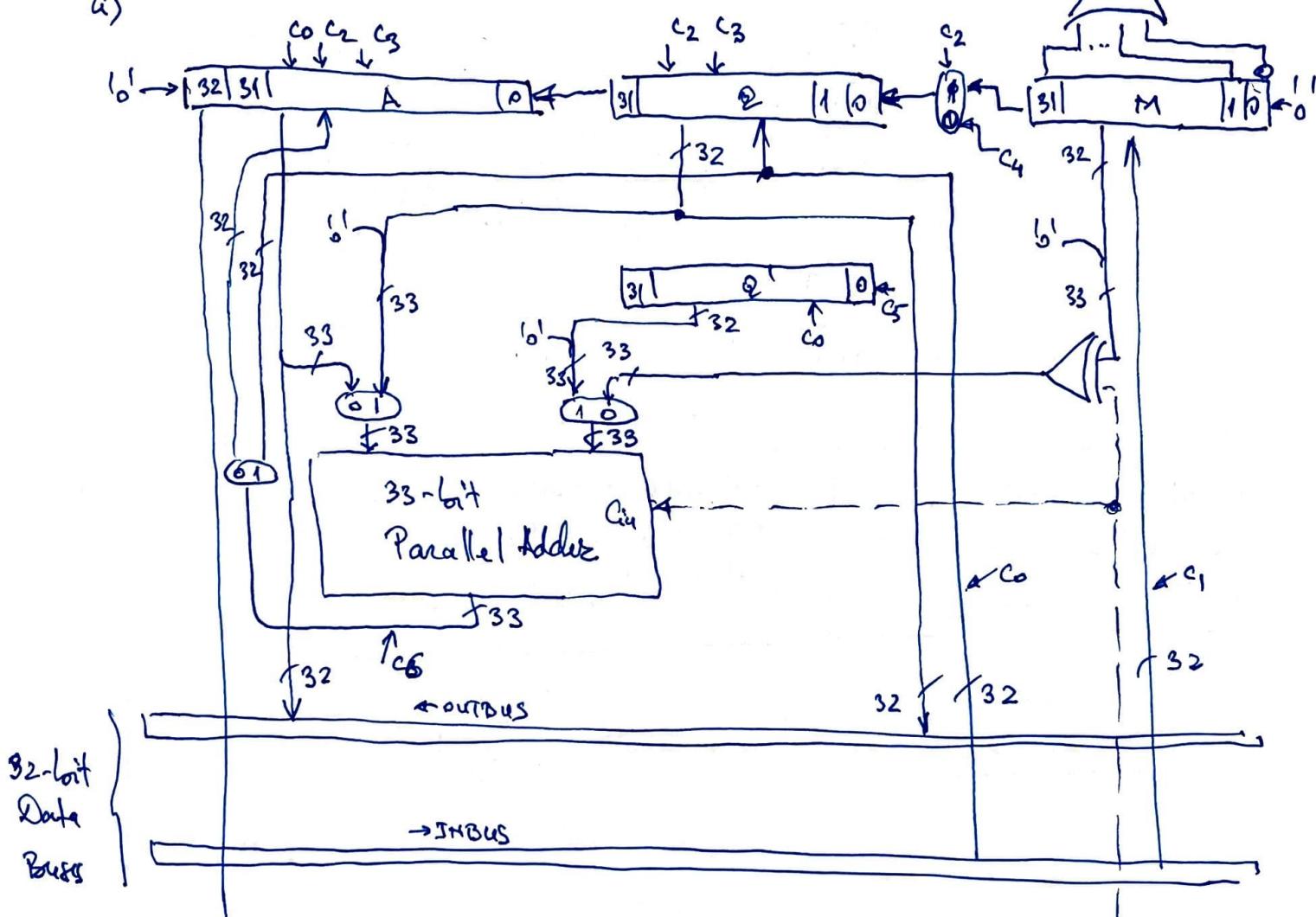
000	-P
001	-M
010	-M
011	-2M
100	-2M
101	-M
110	-M
111	-0

1) Să se modifice dispozitivul de împărțire SRF radix-2 astfel încât acesta să împarte numere pe 32 de biți fără a genera cîteva cîifre cînd rezultatul este 0 (deorice să nu se genereze 0 în mod distinct situația de împărțire la 1, de cînd:

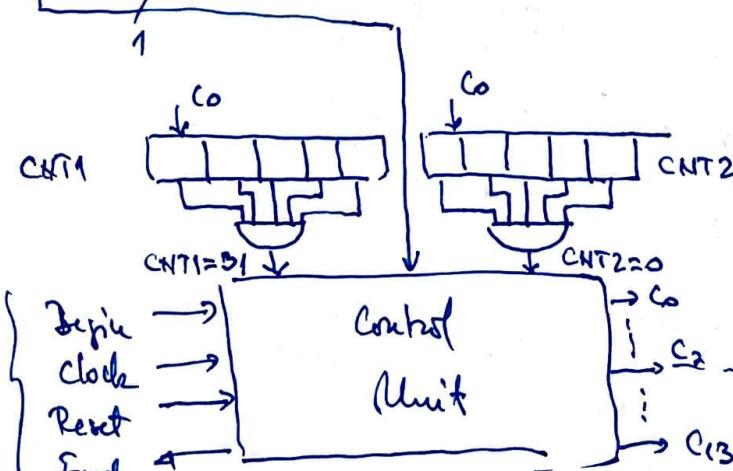
a) Schema bloc hardware folosind platforma Hayes + funcționala de control

b) Ordinograma ce descrie funcțiile unității de control (+ funcționala de control)

a)

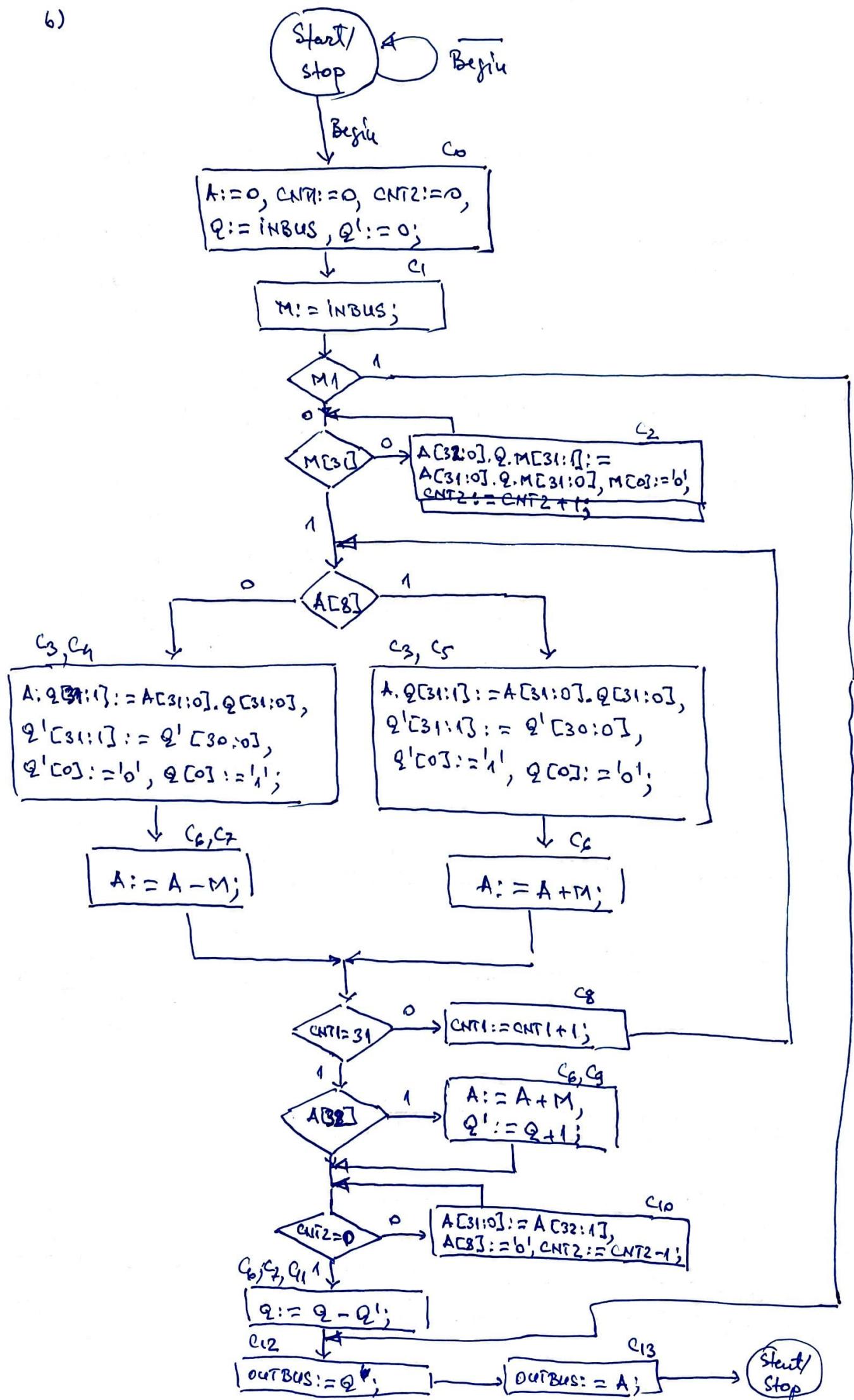


External
Control
signals



Internal
Control
Signals

6)



- ② Considerăm o mașină de calcul × caracterizată printr-un clock frequency I de 5 GHz, care rulează un benchmark trist ce generează următoarea statistică pe tipuri de instrucțiuni.

Inst. type	Clock cycles	Instruction Count
AL	5	5000
MOV	3	10000
L/S	3	3000
Other	2	2000

Restul rămâne
necalcu**late**

Această mașină este optimizată de la echipă, una de hardware, alta de software. Echipa hardware optimizează implementarea setului de instrucțiuni, reducând numărul de cicluri de clock rezultate din instrucțiunile AL de la 5 la 3 și de cele L/S de la 3 la 2. Această optimizare are deopotrivă degradarea clock ~~frequency~~ cycle time la nivel de instrucție set cu 5%.)

Echipa software oferă o optimizare de compilator, ce reduce numărul de instrucțiuni MOV cu 30% și cel de instrucțiuni Other cu 10%.

Să se calculeze care echipă - cea HW sau cea SW - a făcut o treabă mai bună, în conformitate cu

$$\frac{a) \text{ CPU time}}{\text{IP}} \quad \frac{b) \text{ TIPS}}{\text{IP}}$$

$$a) \text{ Clock cycle time}_x = \frac{1}{5 \text{ GHz}} = \frac{1}{5 \cdot 10^9} \Delta = 0.2 \text{ us}$$

$$\begin{aligned} \text{CPU time}_x &= (25000 + 30000 + 9000 + 4000) \times 0.2 \text{ us} = \\ &= 68000 \times 0.2 \text{ us} = 13600 \text{ us} = \boxed{13.6 \mu\text{s}} \end{aligned}$$

~~$$\text{Clock cycle time}_{HW} = 5 \text{ GHz} - 0.05 \times 5 \text{ GHz} = 0.95 \text{ GHz}$$~~

$$\text{Clock cycle time}_{HW} = \frac{1}{0.95 \times 10^9} \Delta =$$

$$\text{Clock cycle time}_{HW} = 0.2 \text{ us} \times 1.05 = 0.21 \text{ us}$$

$$\begin{aligned} \text{CPU time}_{HW} &= (3 \times 5000 + 3 \times 10000 + 2 \times 3000 + 2 \times 2000) \times 0.21 \text{ us} \\ &= 55000 \times 0.21 \text{ us} = 11550 = \boxed{11.55 \mu\text{s}} \end{aligned}$$

CPU times SW

$$IC_{AL_{SW}} = 5000$$

$$IC_{MOV_{SW}} = 8000$$

$$IC_{LS_{SW}} = 3000$$

$$IC_{other_{SW}} = 1800$$

$$\begin{aligned} CPUtime_{SW} &= (5000 \times 5 + 8000 \times 3 + 3000 \times 3 + \\ &\quad + 1800 \times 2) \times 0.2 \mu s = \\ &= 61600 \times 0.2 \mu s = 12320 \mu s = \\ &= \boxed{12.320 \mu s} \end{aligned}$$

\Rightarrow HW team is better!

b) $IC_{HW} = 20000$

$$IC_{SW} = 17800$$

$$\begin{aligned} MIPS_{HW} &= \frac{20000}{11550 \mu s \times 10^6} = \\ &= \frac{20000}{11550 \times 10^{-9} \times 10^6} = \frac{20000}{11550} \times 10^3 \\ &= 17316 \times 10^3 = \boxed{1731.6} \end{aligned}$$

$$MIPS_{SW} = \frac{17800}{12320 \times 10^{-9} \times 10^6} = \boxed{1444,8}$$

④ Considerăm o mașină × caracterizată printr-un clock frequency de 4.5 GHz, care rulează un buclă de lucru în care generează următoarele statistici pe tipuri de instrucțiuni

Instr. type	Clock cycles	Instruction Count
AL	4	10 000
Mov	3	10 000
L/S	2	7 000
Other	1	3 000

Această mașină este optimizată de 2 echipe, una HW și alta SW. Echipa HW optimizată implementarea setului de instrucțiuni, reducând numărul de cicluri de clock recomandate de instrucțiunile AL de la 4 la 3 și Mov de la 3 la 2. Restul rămâne reziliabilă. Această optimizare ~~se~~ a fost doar de alimentare CPU-ului la o frecvență crescută și implică o întărire a clock cycle time cu 5%.

Echipa SW oferă o optimizare de compilator, ce reduce numărul de instrucțiuni AL și Mov cu 15% și cel de L/S cu 10%.

Să se calculeze care echipă - cea HW sau cea SW - a făcut o treabă mai bună, în conformitate cu

a) CPU time
CPU

b) MIPS
MIPS

$$a) \text{Clock cycle time}_x = \frac{1}{4.5 \text{ GHz}} = \frac{1}{4.5 \times 10^9} \text{ s} = 0.222 \mu\text{s} = \underline{\text{Clock cycle time}_{SW}}$$

$$\begin{aligned} \text{CPU time}_x &= (4 \times 10000 + 3 \times 10000 + 2 \times 7000 + 1 \times 3000) \times 0.222 \\ &= 87000 \times 0.222 = 19331.4 \mu\text{s} = \boxed{19.3314 \mu\text{s}} \end{aligned}$$

$$\text{Clock cycle time}_{HW} = 0.95 \times 0.222 \mu\text{s} = \underline{0.211 \mu\text{s}}$$

$$\text{CPU time}_{HW} = (3 \times 10000 + 2 \times 10000 + 2 \times 7000 + 1 \times 3000) \times 0.211 \mu s$$

$$= 14.137 \mu s = \boxed{14.137 \mu s}$$

$$IC_{AL_{SW}} = 0.85 \times 10000 = 8500$$

$$IC_{MOV_{SW}} = 0.85 \times 10000 = 8500$$

$$IC_{LG_{SW}} = 0.9 \times 7000 = 6300$$

$$IC_{OP_{HW\ SW}} = 3000 = 3000$$

$$\left. \begin{array}{l} \text{CPU time}_{SW} = (7 \times 8500 + 2 \times 6300 \\ \quad + 3000) \times 0.2222 \mu s \\ = 16.687.22 \mu s \\ = \boxed{16.68722 \mu s} \end{array} \right.$$

b)

$$IC_{HW} = 30000$$

$$IC_{SW} = 26300$$

$$MIPS_{HW} = \frac{30000}{14.137 \cdot 10^{-6} \times 10^6} = 2122.09$$

$$MIPS_{SW} = \frac{26300}{16.68722 \cdot 10^{-6} \cdot 10^6} = 1576.0564$$

③ Asun un computer load/store cu un sistem cache ce se caracterizează prin următoarele valori empirice atunci când redăm un același program: 10% Data Miss Rate, 8% Instruction Miss Rate și 20% instrucțiuni load/store. În orice moment, 20% din bufferi sunt modificate (i.e., dirty). Programul determină un CPI de 2 clock cycles dacă cache-ul este perfect și constă din 10 000 de instrucțiuni în cod masină. Calculatorul are o frecvență a clock-ului de 4.5 GHz. Penalitatea de miss este determinată de faptul că un access la BLS ~~intervadă~~^{intervadă} (o cicluri de clock) latimea BLS-ului este de 4 cicle, block-ul are 16 cicle, iar cache-ul implementează o politică Write Back cu Write Allocate în cazul unei miss. Calculați:

① a) TOTAT, știind că tac = 2 ce.

① b) CPI time.

a)

$$\begin{aligned}
 \text{Read Miss Penalty} &= \underbrace{\frac{16 \text{ words}}{4 \text{ words}}}_{\text{Update}} \times 0.2 \text{ Bus units} + \underbrace{\frac{16 \text{ words}}{4 \text{ words}}}_{\text{Allocate}} \text{ BUS Reads} \\
 &= \text{Write Miss Penalty} \quad \text{Allocate}
 \end{aligned}$$

$$= 4 \cdot 0.2 + 4 = 4(1.2) = 4.8 \text{ BUS Accesses}$$

$$\Rightarrow \text{Miss Penalty} = 4.8 \times 10 = \boxed{48 \text{ C.C.}}$$

$$\text{Clock cycle time} = \frac{1}{4.5 \cdot 10^9 \text{ s}^{-1}} = 0.2222 \text{ us}$$

$$\text{Misses Per Instruction} = \text{Miss Rate} \times \overbrace{\text{Num.accesses per instruction}}^{1.2}$$

$$\text{Misses Per Instruction} = 1 \times \text{IMR} + 0.2 \times \text{DMR} = 0.08 + 0.2 \times 0.1$$

$$\text{Miss Rate} = \frac{0.1 \cancel{+ 0.08}}{1.2} = 0.0833 \approx 0.1$$

$$AMAT = 0.4444 \mu s + 0.0833 \times 48 \text{cc} \times 0.2222 = \\ = \boxed{1.3328 \mu s}$$

6.8

$$b) CPU_{time} = 10000 \underbrace{(2 + 0.1 \times 48)}_{6.8} \times 0.2222 = \\ = 15109.6 \mu s = \boxed{15,109.6 \mu s}$$

- (3) Avem un computer LIS cu un sistem cache ce are caracteristicile următoarele valori empirice atunci când rularea unui anumit program: 9% Data Miss Rate, 8% Instruction Miss Rate și 10% instrucțiuni L/S. În orice moment 10% din blocuri sunt modificate (i.e., dirty). Programul determină un CPT de 3 c.c. decât cache-ul este perfect și conține din 1000 de instrucțiuni în cod mașină. Calculatorul are o frecvență a clock-ului de 5 GHz. Penalitatea de miss este determinată de faptul că un access la BCS rezulă în 8 cicluri de clock, latenția BCS-ului este de 2 c.c., block-ul are 32 de adrese iar cache-ul implementează o politică WriteThrough sau Write No Allocate în cazul unui miss.
- 10% ~~sunt~~ din accese sunt scrise, restul sunt citiri.

Calculări:

- 1P a) ATAT, știind că $t_{ac} = 1 \text{ cc}$.
- 1P b) CPU time

$$\text{a) Clock cycle time} = \frac{1}{5 \cdot 10^9 \text{ s}^{-1}} = \boxed{0.2 \text{ ns}}$$

$$\text{Read miss Penalty} = \frac{\underbrace{32 \text{ words}}_{\substack{\text{Allocate}}} \overline{\text{2 words}}}{\text{Allocate}} = 16 \text{ BUS reads}$$

$$\text{Write Miss Penalty} = 1 \text{ BUS write}$$

$$\begin{aligned} \text{Miss Penalty} &= 0.1 \text{ Write miss Penalty} + 0.9 \text{ Read miss Penalty} \\ &= 0.1 \times 1 + 0.9 \times 16 = 14.5 \text{ BUS accesses} \\ &= 14.5 \times 8 = \boxed{116} \text{ clock cycles} \end{aligned}$$

$$\text{Misses Per Instruction} = \text{Miss Rate} \times \overbrace{\text{Numele acc per instrucțiun}}^{A.1}$$

$$\text{Misses Per Instruction} = 0.08 + 0.09 \times 0.1 = 0.089$$

$$\text{Miss Rate} = \frac{0.089}{1.1} = 0.0809$$

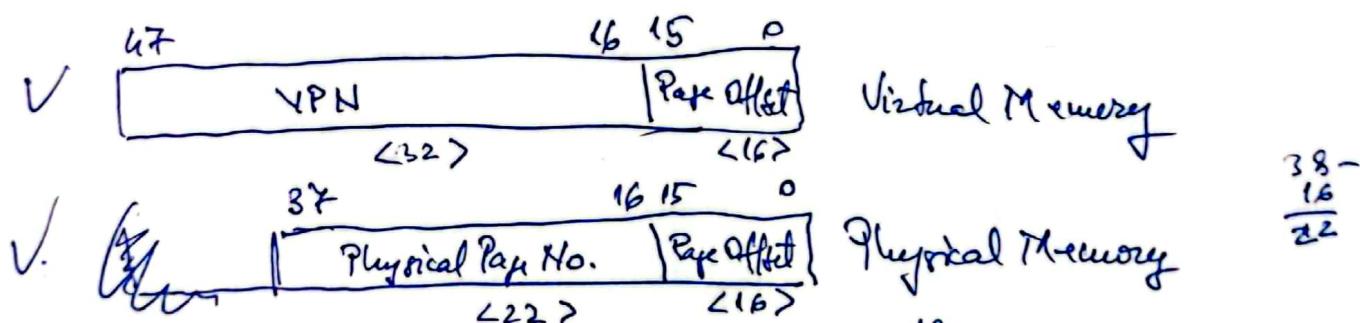
a) ATAT = $0.2 + 0.0809 \times 116 \times 0.2 = \boxed{2.07688 \text{ us}}$

b) CPU time = $1000 \times (3 + 0.089 \times 116) \times 0.2 =$
 $= 2664.8 \text{ us} = \boxed{2.664 \mu\text{s}}$

4) Se dă un sistem de memorie, cu spațiu virtual de dimensiune I
 256 TiB, spațiu fizic de 256 GiB și adresa la nivel de byte. Cadrul este pe 64 de biți, block-ul are 64 de cărți. O pagină de memorie are dimensiunea de 64 KiB, iar dimensiunea de date a cache-ului (SA pe 2 căi) este de 256 KiB. Cache-ul este Write Back, cu Write Allocate în caz de Write Miss și are o politică de replace random. TLB-ul are 64 de intrări și folosește măpare directă.
 Se cer:

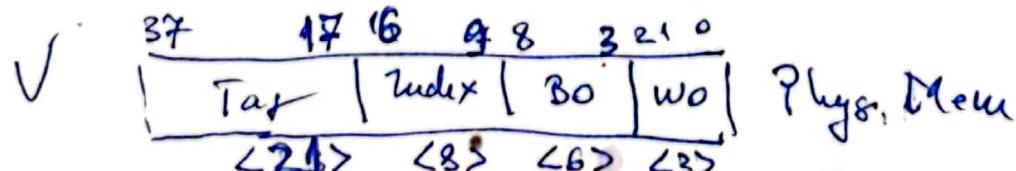
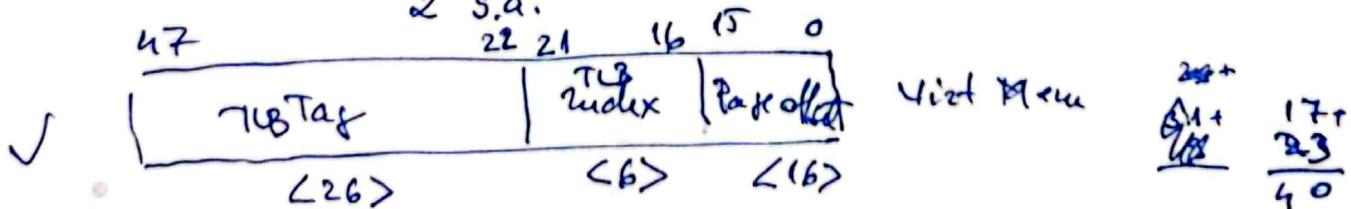
- 1P a) Formatul adresării de memorie virtuală respectiv fizică
- 1P b) Dimensiunea totală a memoriei cache și a TLB-ului.
- 1P c) Schema bloc a mapării adresă fizică-adresă virtuală ce conține memoria cache și TLB-ul (ca în curs).

$$\begin{aligned}
 \text{a)} \quad VMSA &= 2^8 \times 2^{40} B = 2^{48} B & P_{pg} &= 64 KiB = 2^{16} B \\
 PMSize &= 2^{38} B & \text{word offset} & \\
 (\text{word}) &= 2^3 B & \text{block offset} & \\
 1 \text{ block} &= 2^6 \text{ words} = 2^6 \cdot 2^3 B = 2^9 B & \text{TLB entries} &= 2^6 \xrightarrow{\text{TLB index}}
 \end{aligned}$$



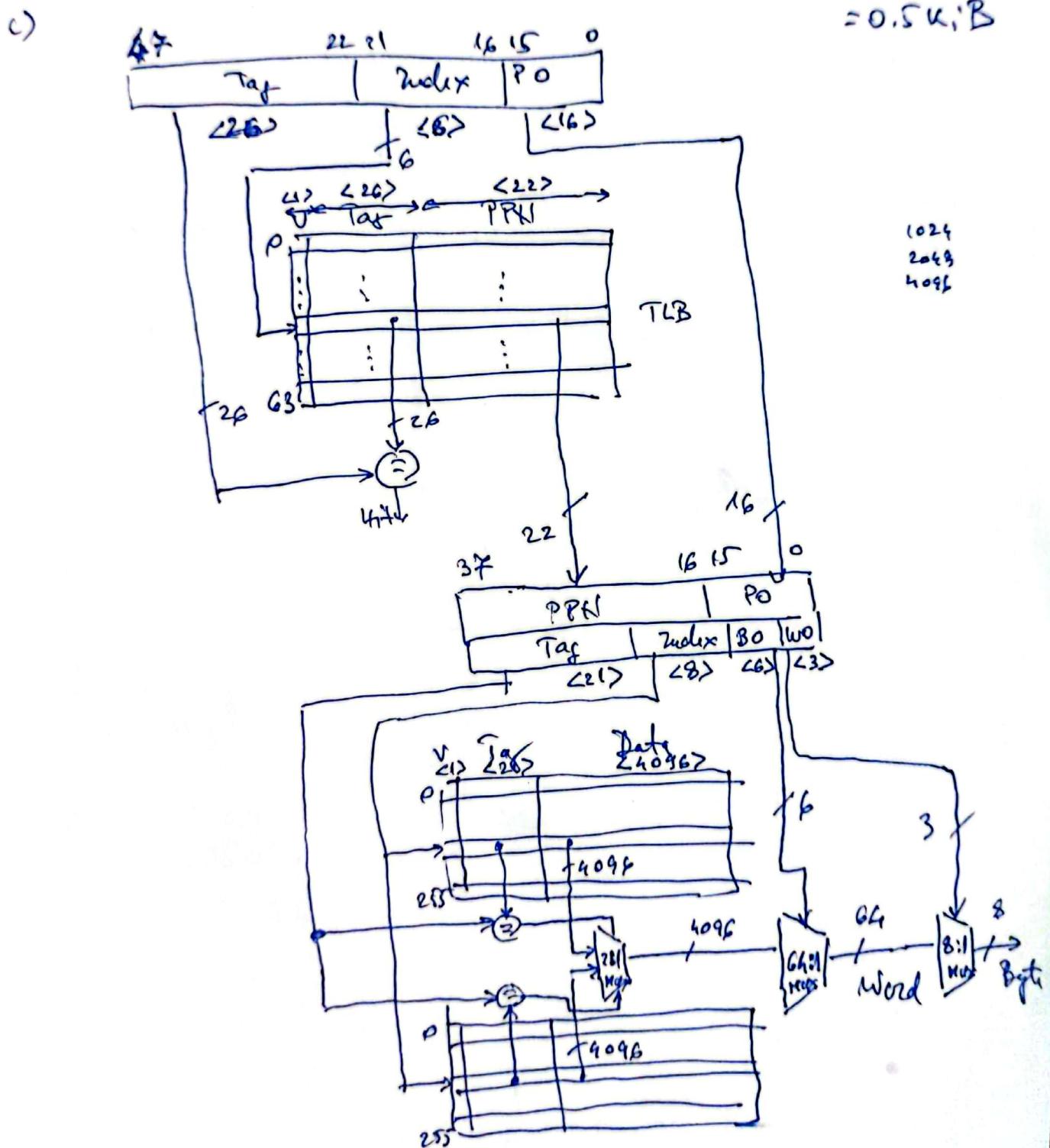
$$\text{Cache data size} = 256 \text{ KiB} = 2^{18} B = \frac{2^{18} B}{2^9 B/\text{block}} = 2^9 \text{ blocks}$$

$$\text{Bank size} = \frac{2^9 \text{ blocks}}{2^1 \text{ S.A.}} = 2^8 \xrightarrow{\text{Index}} \text{blocks/bank.}$$



$$\begin{aligned}
 b) \text{Cache Size} &= \underbrace{2^1}_{\text{s.a}} \times \underbrace{2^8}_{\text{index}} \times \left(\underbrace{2^9 \text{B}}_{\text{data block}} + \underbrace{16 + 216}_{\substack{\text{16 dirty} \\ \text{Tag}}} \right) = 2^9 \times 2^9 \text{B} + 2^9 \cdot 3 \text{B} = \\
 &= 2^{18} \text{B} + 3 \times 0,5 \text{KiB} \\
 &\cong 3 \text{B} \\
 &= 256 \text{KiB} + 1,5 \text{KiB} \\
 &= 257,5 \text{KiB}
 \end{aligned}$$

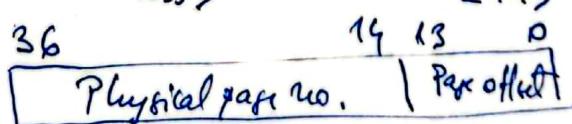
$$\begin{aligned}
 \text{TLB Size} &= 2^6 \times \left(\underbrace{1}_{\text{TLB tag}} + \underbrace{2^6}_{\text{PPN}} + \underbrace{2^2}_{\text{L1}} \right) \text{bits} = 2^6 \times \underbrace{496}_{\cong 2^9} = 2^6 \times 2^3 \text{B} \\
 &\cong 2^3 \text{B} = 2^9 \text{B} \\
 &= 0,5 \text{KiB}
 \end{aligned}$$



- (4) Se dă un sistem de memorie cu spațiu virtual de dimensiunea 2^{48} și 512 TiB, spațiu fizic de 128 GiB și adresa la nivel de byte.
- Curențul este pe 64 de biti, block-ul are 32 de cărți. O pagină de memorie are dimensiunea de 16 KiB, iar dimensiunea de date a cache-ului (stoc 4 căi) este de 128 KiB. Cache-ul este Write Through cu Write No Allocate în caz de Write Miss și are o politică de relaşare de tip LRU cu age register. TLB-ul are 128 de ieșiri și folosește maparea set-associativă pe 2 căi. Se cer:
- (1P) a) Dimensiunea totală a memoriei cache și a TLB-ului
- (1P) b) Formatul adresii de memorie virtuală, respectiv fizică
- (1P) c) Schema bloc a mapării adresa virtuală-adresa fizică a contine memoria cache și TLB-ul (ca în curs pt Fast MMU).

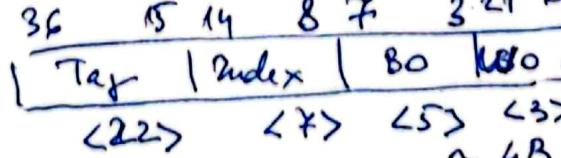
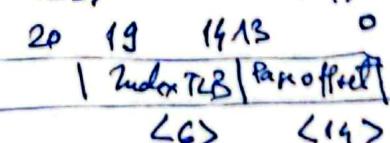
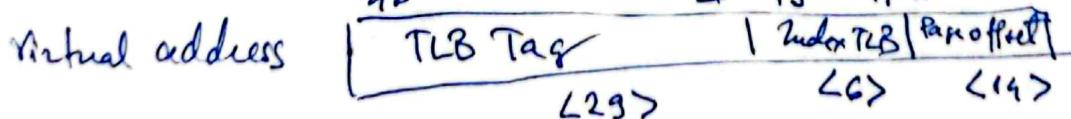
$$\begin{aligned} \text{a) Virtual Space} &\rightarrow 512 \text{ TiB} = 2^{49} \text{ B} \\ \text{Physical Space} &\rightarrow 128 \text{ GiB} = 2^{37} \text{ B} \\ 1 \text{ word} &= 64 \text{ bits} = 2^6 \text{ B} \\ 1 \text{ block} &= 2^5 \text{ words} = 2^5 \cdot 2^6 \text{ B} = 2^{11} \text{ B} \end{aligned}$$

$$\begin{aligned} 1 \text{ page} &= 2^{14} \text{ B} \\ \text{Cache data size} &= 2^{17} \text{ B} = \\ &= 2^{\frac{17}{2}} \text{ B} = 2^8 \text{ blocks} \\ \text{Bank size} &= \frac{2^8 \text{ blocks}}{2^3 \text{ B/block}} = 2^5 \text{ blocks} \end{aligned}$$



128 entries TLB = 2^7 entries

$$\frac{2^7}{2^1 \text{ sa}} = 2^6 \text{ index}$$



✓

✓

$$\begin{aligned} \text{b) Cache total size} &= \underbrace{2^2}_{\text{s.a.}} \times \underbrace{2^7}_{\text{Index}} \times \left(2^8 \text{ B} + \underbrace{\underbrace{1}_{\text{v}} + \underbrace{2}_{\text{age}} + \underbrace{22}_{\text{tag}}}_{\cong 4 \text{ B}} \right) = \\ &= 128 \text{ KiB} + 2^9 \times 2^8 \text{ B} = 128 \text{ KiB} + 2 \text{ KiB} = \boxed{130 \text{ KiB}} \end{aligned}$$

$$TLB_{size} = 2^7 \times (\underbrace{\frac{1}{4}}_{tag} + \underbrace{\frac{23}{4}}_{PPN} + \underbrace{\frac{23}{4}}_{Index}) \cong 2^7 (24 + 32) \text{ bits} = \\ = 2^7 \cdot 7 \text{ B} \cong 2^7 \cdot 8 \text{ B} \cong \boxed{1 \text{ KB}}$$

c)

