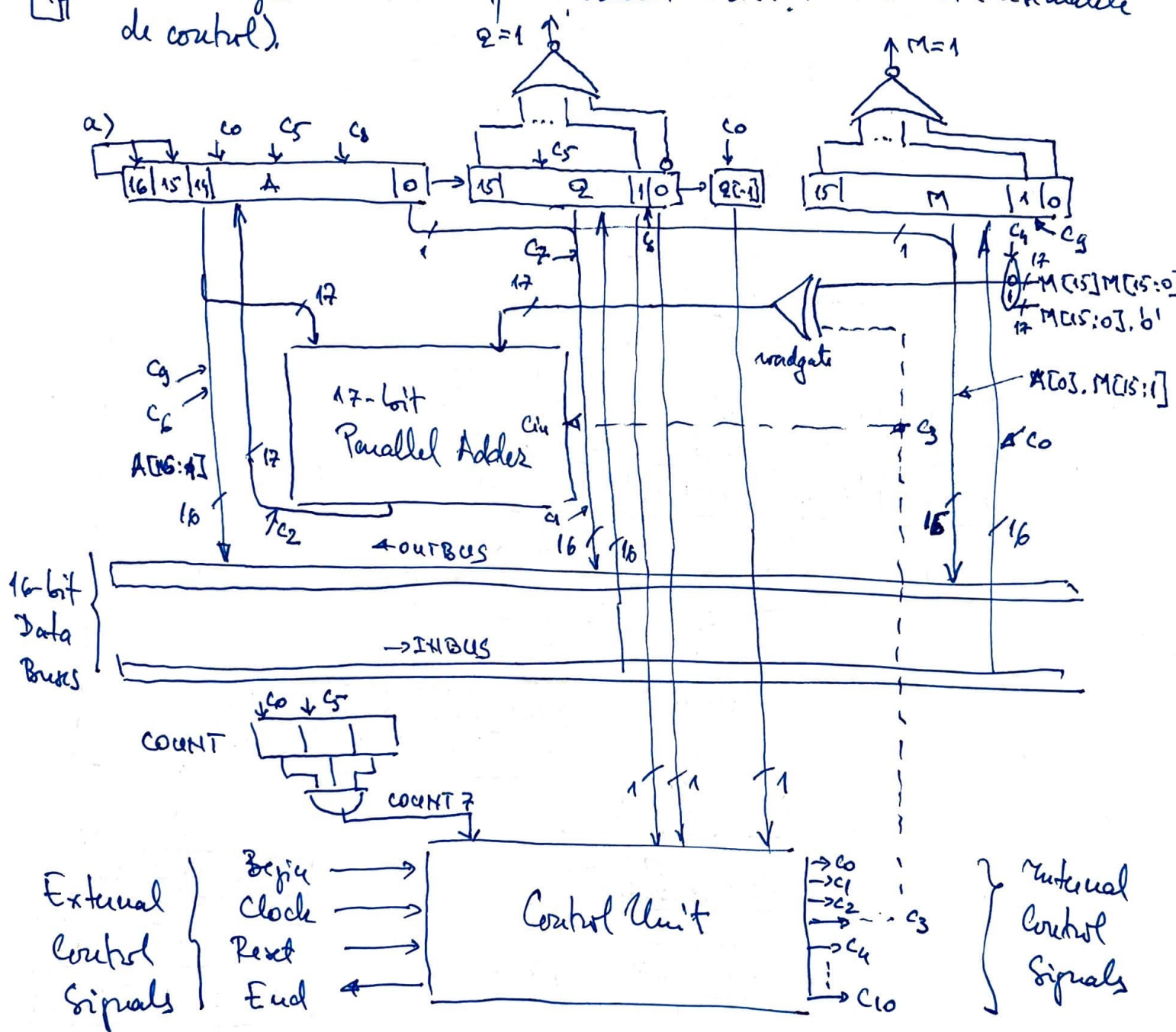
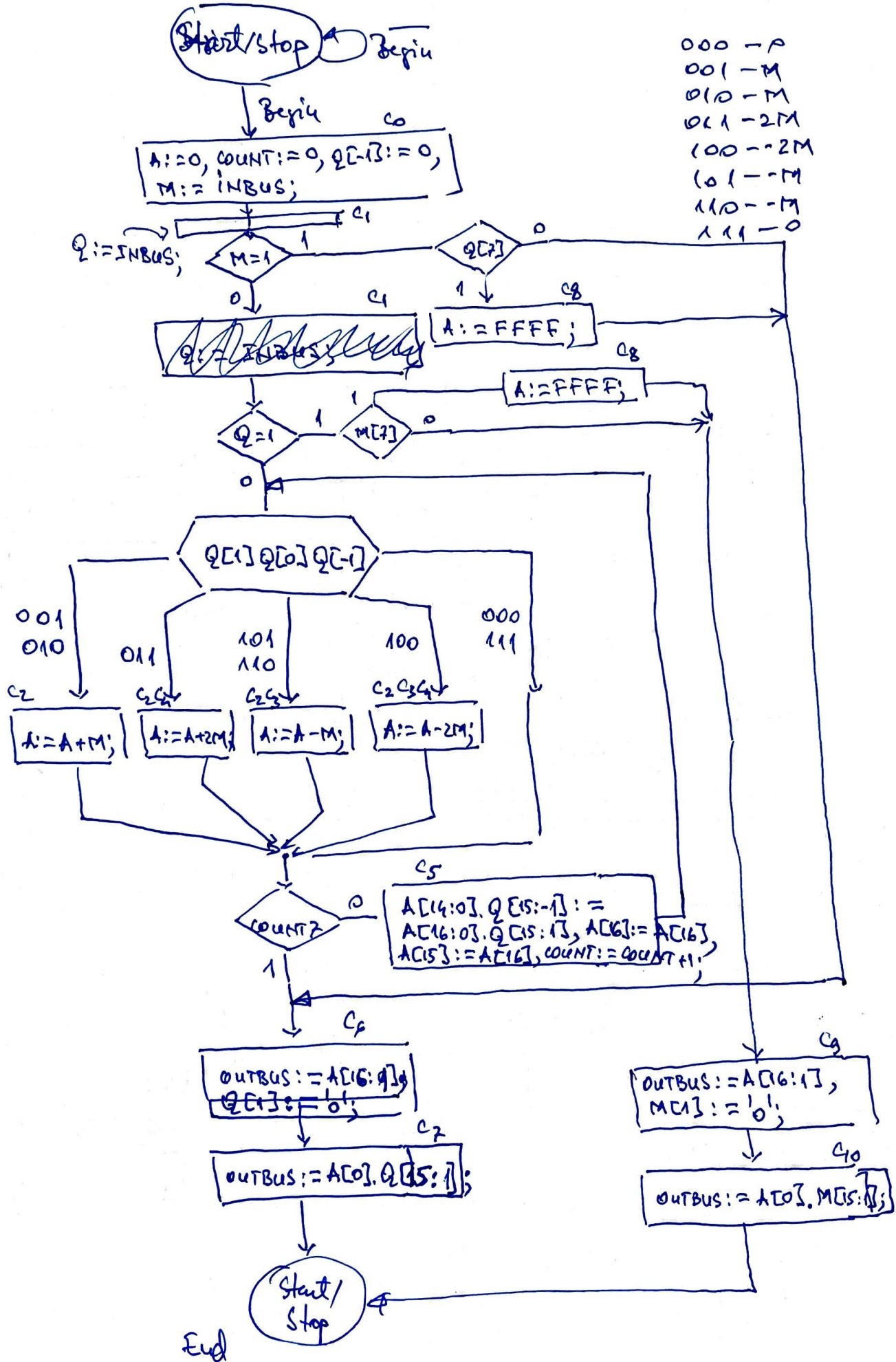


- (2p) ① Să se modifice dispozitivul de înmulțire Booth radix-4, astfel încât acesta să înmulțească numere fracționare și să trateze în mod distinct situațiile de înmulțire cu unu. (Rezultatul este egal cu celălalt operand, exprimat pe 32 biți.) Se cere:

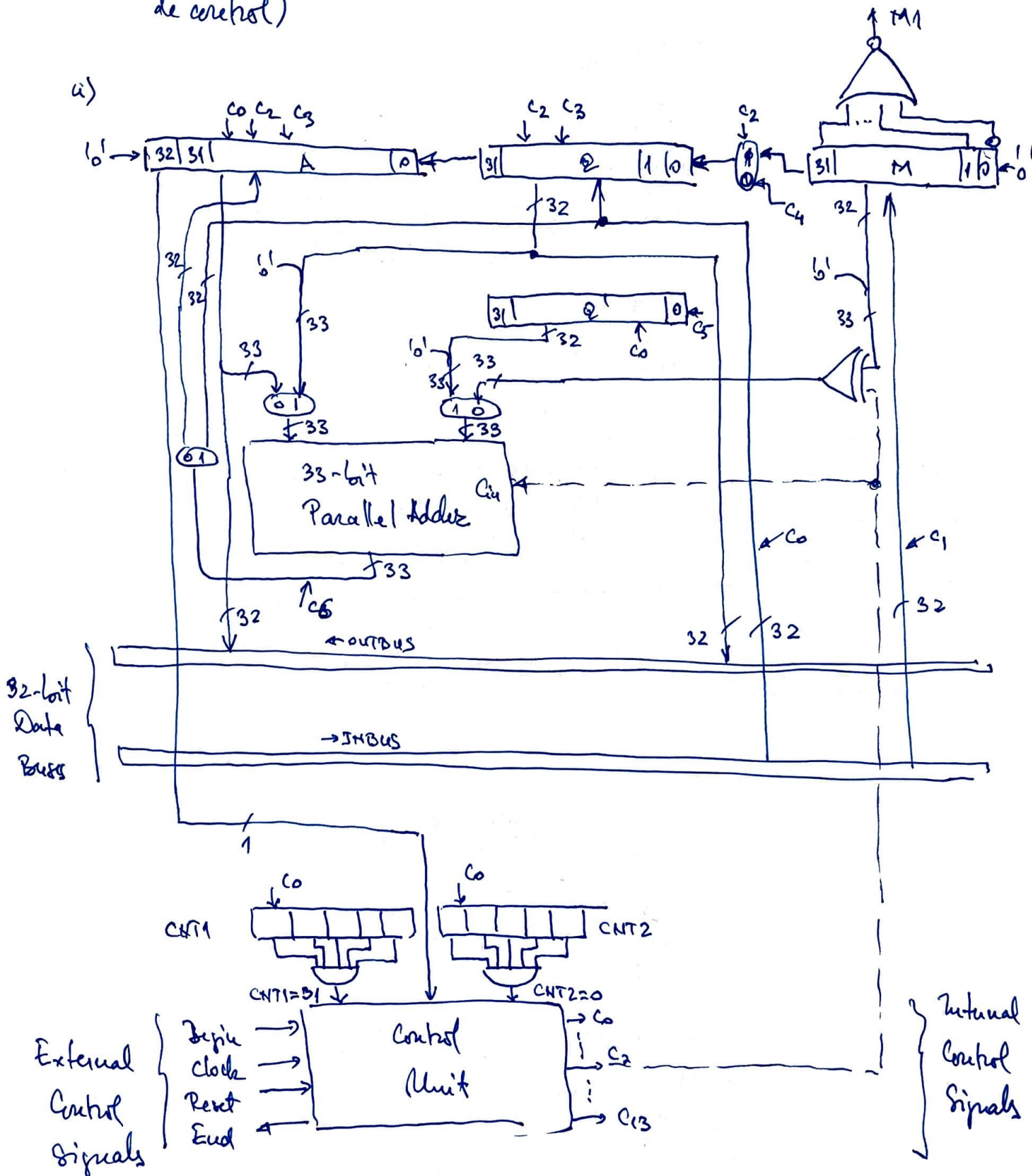
- ①p a) Schema bloc hardware pe platforma ~~hardware~~ Hayes. (cu semnale de chl.)
 ①p b) Ordinograma ce descrie funcționarea unității de control (cu semnalele de control).



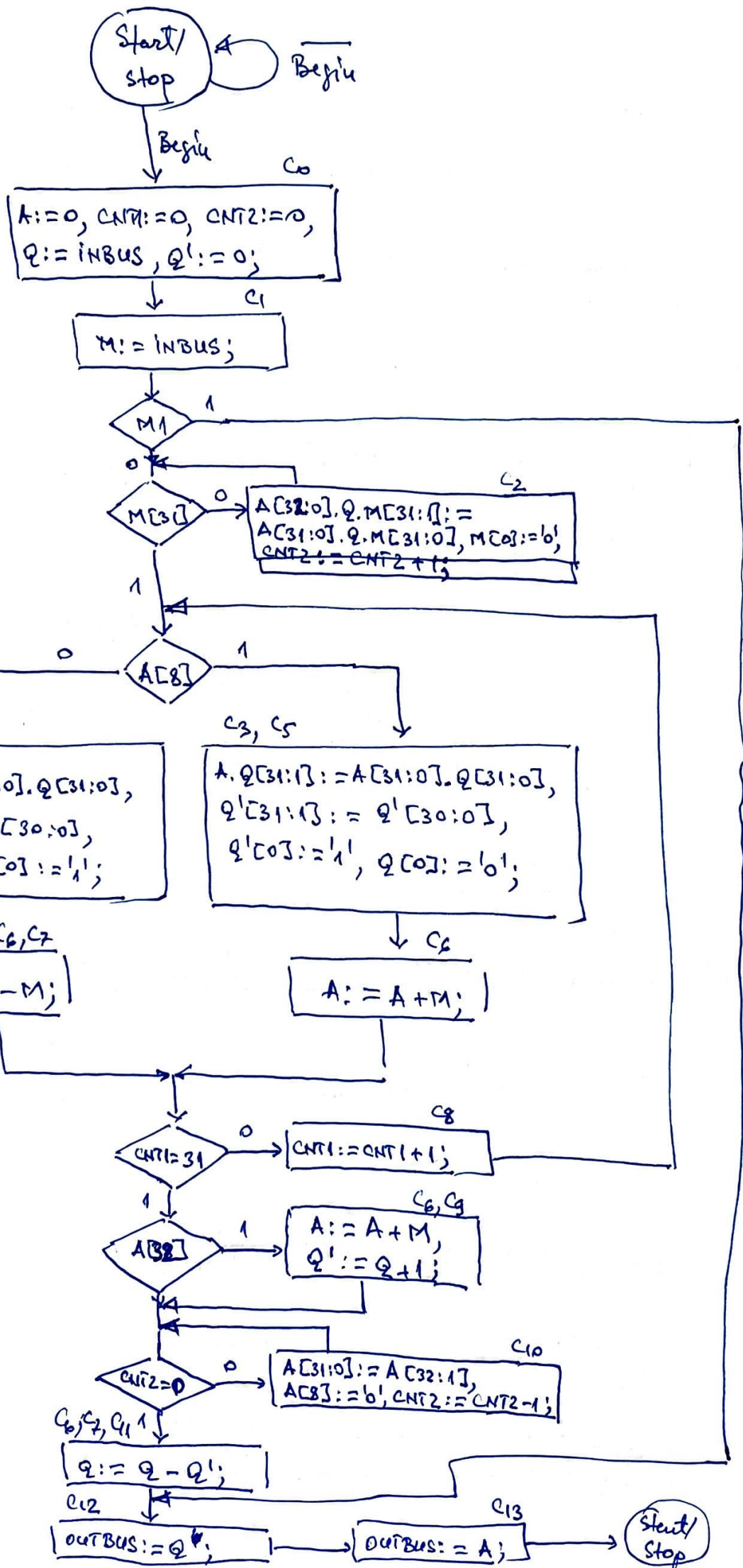


- a) Schema bloc hardware foloșind platforma Hayes + semnalul de control

- b) Ordinograma ce descrie funcționarea unității de control (+ avarale de control)



6)



- ② Considerăm o mașină de calcul x caracterizată printr-un clock frequency f de 5 GHz, care rulează un benchmark suite și 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 rămân
necorectate

Această mașină este optimizată de 2 echipe, una de hardware, alta de software. Echipa hardware optimizează implementarea setului de instrucțiuni, reducând numărul de cicluri de clock necesitate de instrucțiunile AL de la 5 la 3 și de cele L/S de la 3 la 2. (Această optimizare are drept consecință degradarea clock ~~frequency~~ cycle time la nivel de instrucțiuni 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

a) CPU time

b) MIPS

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

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

~~Frequency~~
~~Clock cycle time~~ HW = ~~5 GHz - 0.05 \times 5 GHz = 0.95 GHz~~

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

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

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

~~CPU time SW~~

$$IC_{ALSW} = 5000$$

$$IC_{MOVSW} = 8000$$

$$IC_{LSW} = 3000$$

$$IC_{otherSW} = 1800$$

$$CPU_{timeSW} = (5000 \times 5 + 8000 \times 3 + 3000 \times 3 + 1800 \times 2) \times 0.2 \mu s =$$

$$= 61600 \times 0.2 \mu s = 12320 \mu s$$

$$= \boxed{12.320 \mu s}$$

\Rightarrow HW team is better!

b) $IC_{HW} = 20000$

$$IC_{SW} = 17800$$

$$MIPS_{HW} = \frac{20000}{11550 \mu s \times 10^{-6}} =$$

$$= \frac{20000}{11550 \times 10^{-9} \times 10^6} = \frac{20000}{11550} \times 10^3$$

$$= 17316 \times 10^3 = \boxed{1731.6}$$

$$MIPS_{SW} = \frac{17800}{12320 \times 10^{-9} \times 10^6} = \boxed{1444.8}$$

2) Considerăm o mașină X caracterizată printr-un clock frequency f de 4.5 GHz, care rulează un benchmark suite ce generează următoarea statistică pe tipuri de instrucțiuni

Inst. 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 optimizează implementarea setului de instrucțiuni, reducând numărul de cicluri de clock necesitate de instrucțiunile AL de la 4 la 3 și MOV de la 3 la 2. Restul rămân neschimbate. Această optimizare ~~are~~ a fost dublată de alimentarea CPU-ului la o frecvență crescută și implicit de o îmbunătățire 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
[1P]

b) MIPS
[1P]

$$a) \text{ Clock cycle time}_x = \frac{1}{4.5 \text{ GHz}} = \frac{1}{4.5 \cdot 10^9} \text{ s} = 0.2222 \mu\text{s} = \text{Clock cycle time}_{SW}$$

$$\text{CPU time}_x = (4 \times 10000 + 3 \times 10000 + 2 \times 7000 + 1 \times 3000) \times 0.2222 \\ = 87000 \times 0.2222 = 19331.4 \mu\text{s} = \boxed{19.3314 \mu\text{s}}$$

$$\text{Clock cycle time}_{HW} = 0.95 \times 0.2222 \mu\text{s} = \underline{0.211 \mu\text{s}}$$

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

$$= 14137 \mu s = \boxed{14.137 \mu s}$$

$$IC_{ALSW} = 0.85 \times 10000 = 8500$$

$$IC_{MOVSW} = 0.85 \times 10000 = 8500$$

$$IC_{L3SW} = 0.9 \times 7000 = 6300$$

$$IC_{OPUSW} = 3000 = 3000$$

$$\text{CPU time}_{SW} = (7 \times 8500 + 2 \times 6300 + 3000) \times 0.2222 \mu s$$

$$= 16687.22 \mu s$$

$$= \boxed{16.68722 \mu s}$$

b) $IC_{HW} = 30000$

$$IC_{SW} = 26300$$

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

$$\text{MIPS}_{SW} = \frac{26300}{16.68722 \cdot 10^{-6} \cdot 10^6} = 1576.0564$$

③ Avem un computer load/store cu un sistem cache ce se caracterizează
 prin următoarele valori empirice atunci când rula un anumit
 program: 10% Data Miss Rate, 8% Instruction Miss Rate și
 20% instrucțiuni load/store. În orice moment, 20% dintr-un blocuri
 sunt modificate (i.e., dirty). Programul determină un CPI de
 2 clock cycles dacă cache-ul este perfect și costă din 10000 de
 instrucțiuni în cod sursă. Calculatorul are o frecvență a
 clock-ului de 4.5 GHz. Penalitatea de miss este determinată
 de faptul că un access la BUS ^{inducă} ~~interzicere~~ 10 cicluri de clock,
 lățimea BUS-ului este de 4 cuvinte, block-ul are 16 cuvinte,
 iar cache-ul implementează o politică Write Back cu Write
 Allocate în cazul unui miss. Calculați:

① a) ~~ITMR~~ ITMR, știind că $\lambda_{ac} = 2cc$.

① b) CPI time.

a)

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

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

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

$$\begin{array}{r}
 0.08 \\
 2.02 \\
 \hline
 0.10
 \end{array}$$

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

$$\begin{array}{r}
 0.245 \\
 1.2 \swarrow
 \end{array}$$

$$\text{Misses Per Instruction} = \text{Miss Rate} \times \text{Mem. accesses per instruction}$$

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

$$\text{Miss Rate} = \frac{0.1 \text{ ~~0.08~~}}{1.2} = 0.0833 \quad \Rightarrow \text{MR} \text{ ~~0.08~~ } = 0.1$$

$$AMAT = 0.4444 \mu s + 0.0833 \times 48 \text{ ce} \times 0.2222 =$$

$$= \boxed{1.3328 \mu s}$$

6.8

$$b) CPU_{time} = 10000 (2 + 0.1 \times 48) \times 0.2222 =$$

$$= 15109.6 \mu s = \boxed{15.1096 \mu s}$$

(3) Avem un computer L/S cu un sistem cache cu n caractere 2^{n-1}
 (2p) prin următoarele valori empirice atunci când rulăm un 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 CPI de 3 c.c. dacă cache-ul este perfect și conține din 1000 de instrucțiuni în cod mașină. Calculatoarele au o frecvență a clock-ului de 5 GHz. Penalitatea de miss e determinată de faptul că un access la BUS necesită 8 cicluri de clock, lățimea BUS-ului este de 2 cuvinte, block-ul are 32 de cuvinte iar cache-ul implementează o politică Write Through cu Write No Allocate în cazul unui miss.
 (10% ~~sunt~~ din accese sunt scrieri, restul sunt citiri).

Calculați:

[1p] a) $ATAT$, știind că $t_{acc} = 1cc$.

[1p] b) CPI_{time}

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

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

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

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

$$\begin{aligned} \text{Misses Per Instruction} &= \text{Miss Rate} \times \overbrace{\text{Mem acc per instruction}}^{4.1} \\ \text{Misses Per Inst} &= 0.08 + 0.09 \times 0.1 = 0.089 \end{aligned}$$

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

$$a) \text{ AMAT} = 0.2 + 0.0809 \times 116 \times 0.2 = \boxed{2.07688 \text{ } \mu\text{s}}$$

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