① $X = -111$; $Y = -52$ $X \to Q$ Booth modificat.

$M = 1100\ 1100$

$-M = 0011\ 0100$

52 - 111 -
64
32 47 -
20 - 32
16 15
=4

$X_{SM} = 1110\ 1111$

$= 1001\ 0001_{C2}$

$Y_{SM} = 1011\ 0100_{SM}$

$= 1100\ 1100_{C2}$

$1\ 1001\ 0061$

01 - 0 00 - 5
00 - 1 10 - 6
00 - 2 11 - 7
10 - 3
01 - 4

COUNT	Q[1:0]	A	Q[-1]	Q	B	M
000	0	0000 0000	1	1001 0001	0	1100 1100
	+	1100 1100			0	
	0	1100 1100			0	
		1110 0110	0	1100 1000	0	
001	0	1111 0011	0	0 1110 0100	0	
010	0	1111 1001	1	0011 0010	0	
011	0	1111 1100	1	1001 1001	0	
100	+	1100 1100			0	
	0	1100 1000			0	
		1110 0100	0	1100 1100	0	
101	0	1111 0010	0	0110 0110	0	
110	0	1111 1001	0	0011 0011	0	
111	+	0011 0100			1	
	0	0010 1101				
		0001 0110	1	0001 1001	1	

-111×-52

222

555

5772

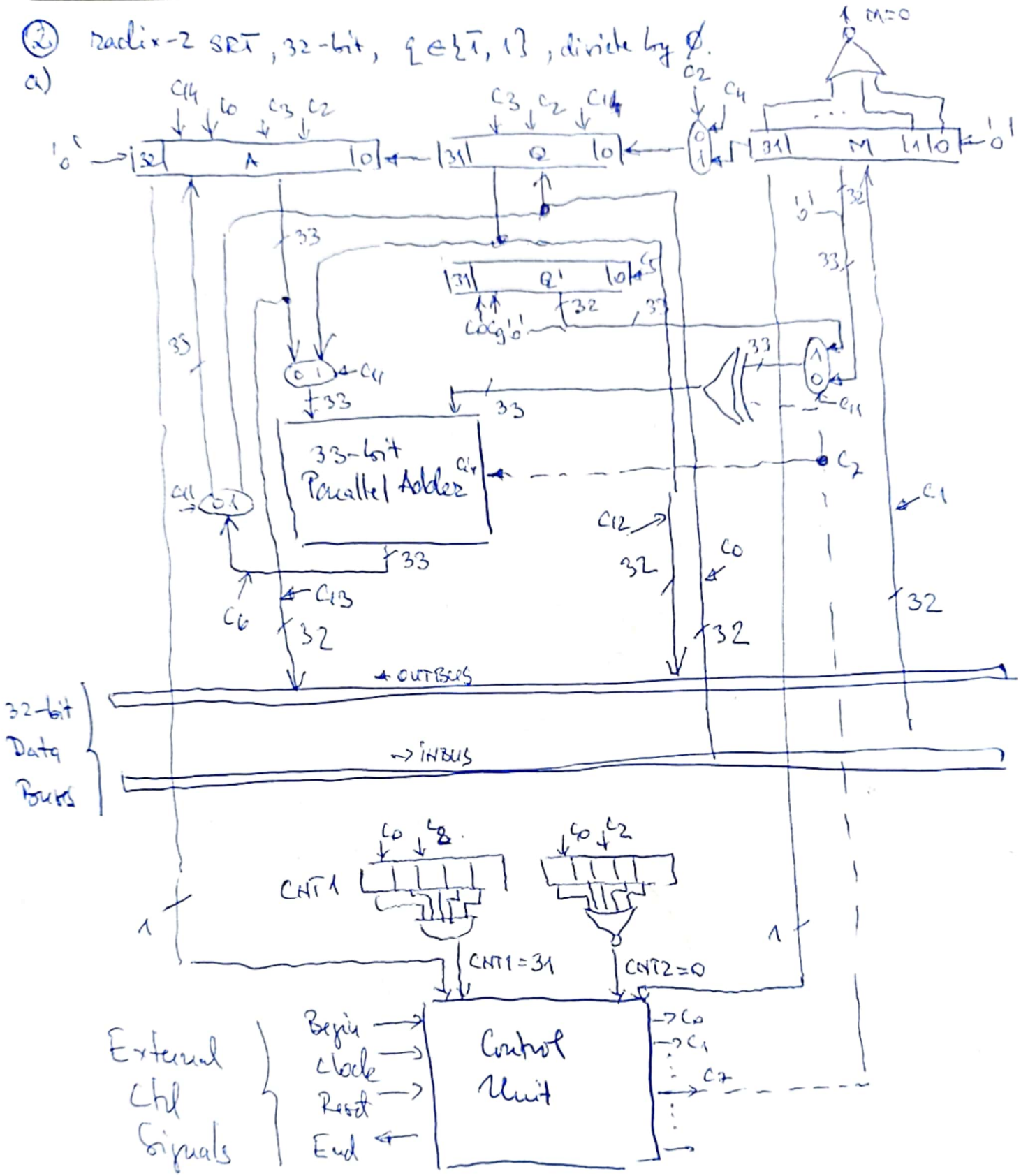
5778 -
4096
1682 -
1024
= 658 -
512
146 -
128
= 18

2048
40

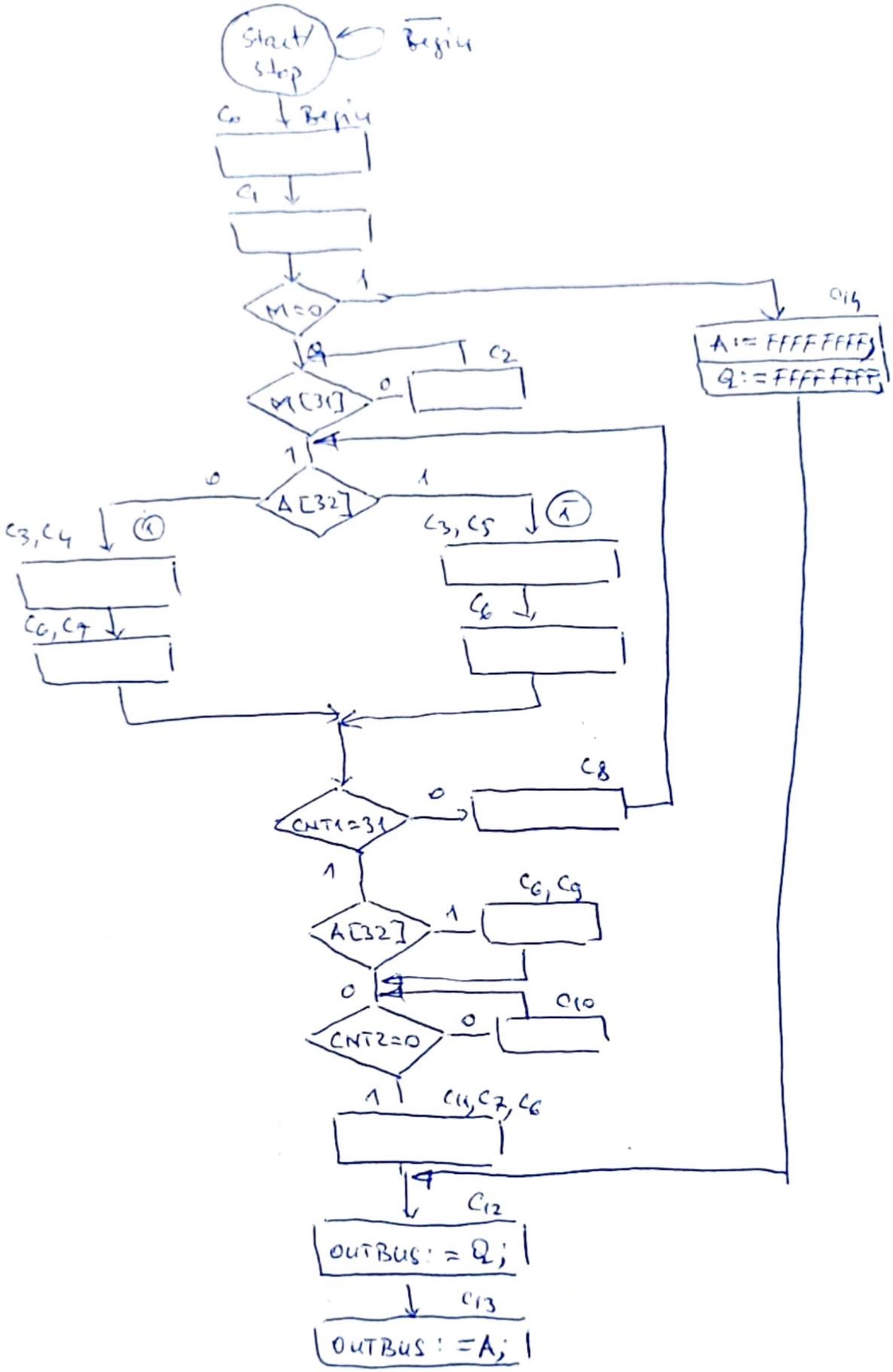
$? =$ 0001011010001100

(2) radix-2 SRT, 32-bit, $q \in \{\bar{1}, 1\}$, divide by 0.

a)



2.b)



(3) a) $CPU\ time_x = IC \times (0.4 \times 3 + 0.3 \times 2 + 0.1 \times 2 + 0.2 \times 1) \times 0.2\,ns$

$= IC \times 0.44\,ns$ $\quad CPI_x = CPI_y$

$CPU\ time_y = IC \times 0.55\,ns = IC \times 2.2 \times 0.25\,ns$

$Clock\ rate_x = \frac{1}{0.2\,ns} = 5\,GHz$

$Clock\ rate_y = \frac{1}{0.25\,ns} = 4\,GHz$

a) r - fracția de instr. LS rămasă după reducerea la ceas y

$$CPU\ time_y^{new} = IC\left[1 - 0.3(1-r)\right] \times \frac{0.4 \times 3 + 0.3 \times r \times 2 + 0.4}{1 - 0.3(1-r)} \times 0.25\,ns$$

$$= IC(1.6 + 0.6 \times r) \times 0.25\,ns$$

Condiția este ca $CPU\ time_y < IC \times 0.44\,ns$

$$IC(1.6 + 0.6 \times r) \times 0.25\,ns < IC \times 0.44\,ns$$

$$1.6 + 0.6 \times r < 1.76 \Rightarrow r < \frac{0.16}{0.6} = 0.2666$$

$\Rightarrow$ fracția ce rămâne < 0.2666

fracția ce se elimină $> (1 - 2.666) = 0.7334$

$\Rightarrow$ procentajul minim de instr LS ce trebuie eliminat $\boxed{73.3\%}$

b) $MIPS = \frac{Clock\ rate}{CPI \times 10^6} \Rightarrow MIPS_x = \frac{5 \cdot 10^9}{2.2 \cdot 10^6} = 2272.727$

$$MIPS_y = \frac{4 \cdot 10^9}{\frac{1.6 + 0.6 \times r}{1 - 0.3(1-r)}\ 10^6} = \frac{4000(1 - 0.3(1-r))}{1.6 + 0.6 \times r} > 2272.72$$

$$4000(0.7 + 0.3r) > 3636.3632 + 1363.6362r$$

$$2800 + 1200r > 3636.3632 + 1363.6362r$$

$$163.6362r < -836.3632$$

$r < -5.111 \Rightarrow$ fracție negativă $\Rightarrow$ imposibil

Era de așteptat, deoarece reducerea de instrucțiuni instrucțiuni certe CPU time dar degradează MIPS!

(4) 64TiB - VM => $2^6 \times 2^{40} = 2^{46}$ B

a) 8GiB - PM => $2^3 \times 2^{30} = 2^{33}$ B

byte addressing

16KiB pages => $2^4 \cdot 2^{10} = 2^{14}$ B

32-entry TLB (2-way SA)

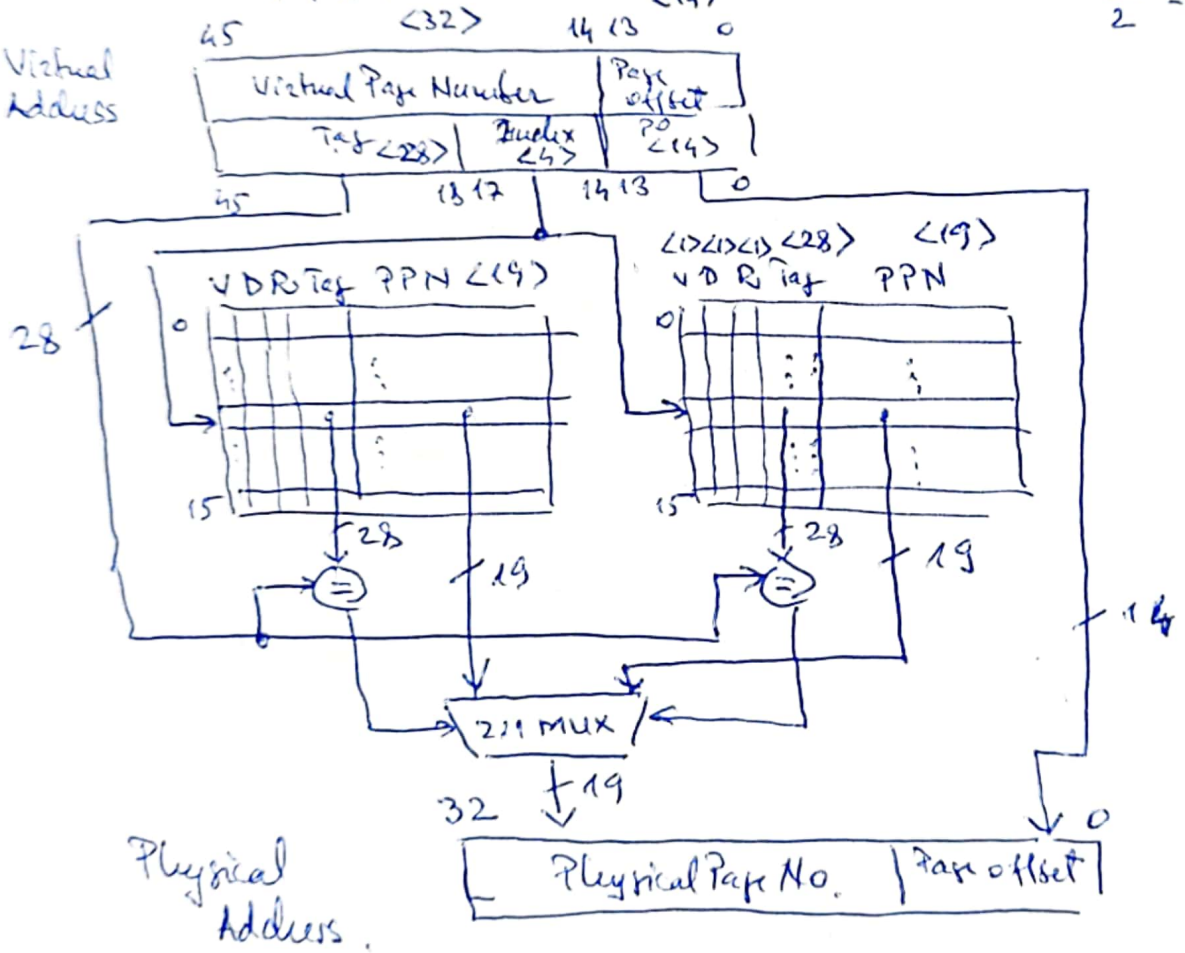
256KiB Data Cache (2-way SA) = 2^{18} B = $\frac{2^{18}}{2^7} = 2^{11}$ blocks

1 block = 16 words = 2^4 W = $2^4 \cdot 2^3$ B = 2^7 B

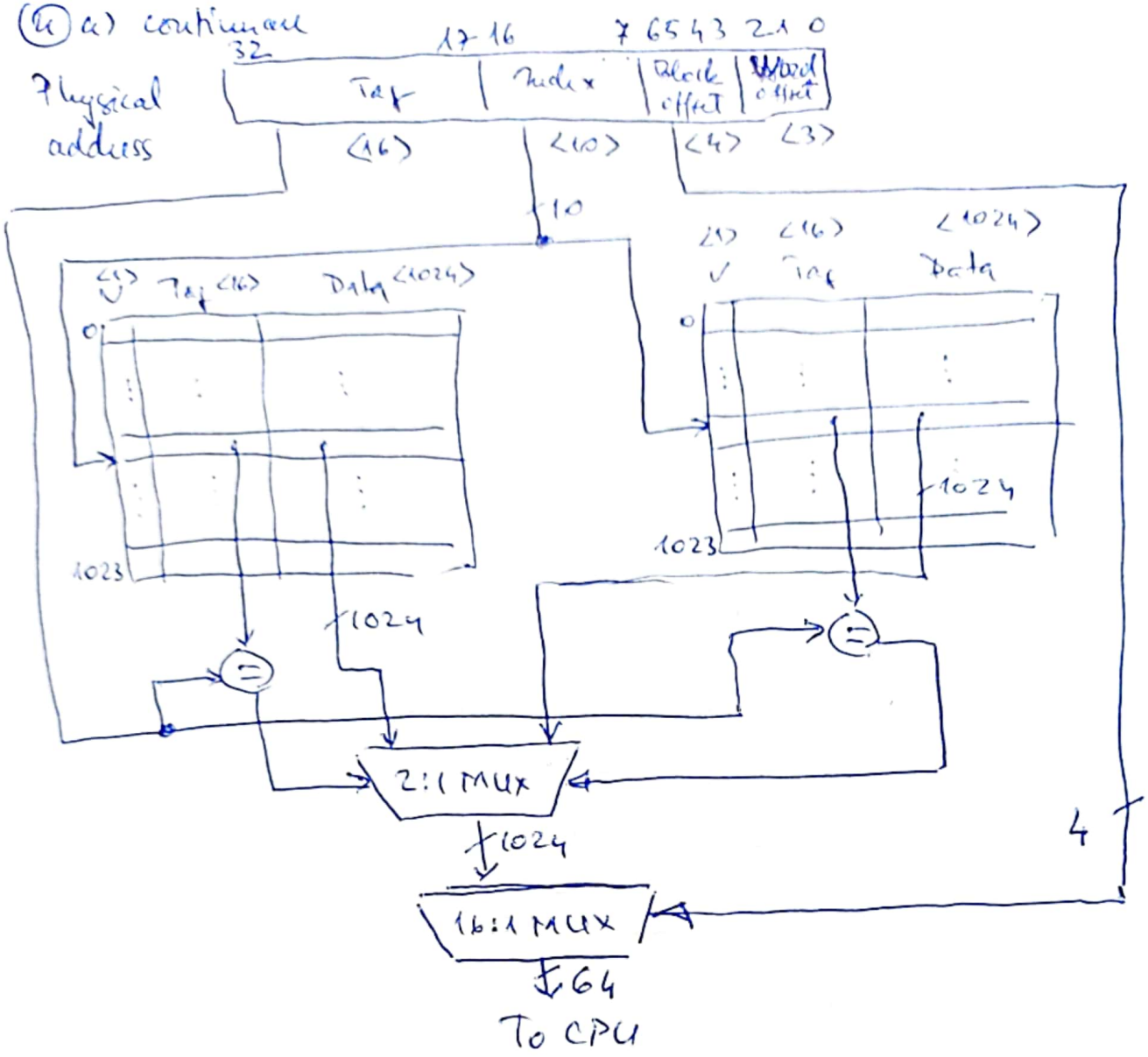
1 word = 64 bits = 2^6 bits = 2^3 B

$\frac{2^{11}}{2} = 2^{10}$ blocks / bank

32 - 17 = 15



(4) a) continuare



b) TLB size = $\underbrace{2^5}_{\text{total entries}} \times (\underbrace{31 \text{ bits}}_{\text{Tag, V, D, R}} + \underbrace{19 \text{ bits}}_{\text{PPN}}) = 2^5 \times 50 \text{ bits}$

$\cong 2^5 \times 2^6 \text{ bits}$

$= 2^5 \times 2^3 B = 2^8 B$

$= 0.25 \text{ KiB}$

Cache size total = $\underbrace{2}_{\text{s.a}} \times \underbrace{2^{10}}_{\text{index}} \times (\underbrace{17 \text{ bits}}_{\text{ctrl bits}} + 2^7 B)$

$= 2^{11} \times 17 \text{ bits} + 2^{18} B$

$\cong 2^{11} \times 2^5 \text{ bits} + \underbrace{256 \text{ KiB}}_{\text{data}} \cong 264 \text{ KiB}$

$2^{16} \text{ bits} = 2^{13} B = 8 \text{ KiB}$

(5) 0.2 L/S iush.

Clock rate = 3.6 GHz

CPI ideal = 3 c.c.

IMR = 8%

DMR = 5%

Miss penalty = 150 ns = $\lceil \frac{150\,ns}{0.277\,ns} \rceil$ = 542 c.c.

Miss rate = 6%

t_{acc} = 2 c.c.

optimizat →

a) $CPU\ time_{orig.} = IC \times (CPI_{ideal} + Misses\ per\ iush. \times Miss\ Penalty) \times Clock\ cycle\ time\ original$

Clock cycle time = $\frac{1}{3.6 \times 10^9\ s^{-1}} = \frac{1}{3.6}\ ns = 0.277\ ns$

Misses per instr original = 0.08 + 0.2 × 0.05 = 0.09

Memory accesses per iush = 1.2

$CPU\ time_{orig} = IC \times (3 + 0.09 \times 540) \times 0.277\ ns$

$= \boxed{IC \times 14.2932\ ns}$

$CPU\ time_{optim} = IC \times (CPI_{ideal} + Mem\ acc\ per\ iush \times Miss\ rate \times Miss\ Penalty) \times CCT$

$= IC(3 + 1.2 \times 0.06 \times 542) \times 0.277\ ns$

$= \boxed{IC \times 11.640\ ns}$ ✓ ⇒ CPU time optim mai bun!

b) $AMAT_{orig} = 0.277\ ns + \frac{0.09}{1.2} \times 542 \times 0.277\ ns$

$= 11.537\ ns$

$AMAT_{optim} = 2 \times 0.277\ ns + 0.06 \times 542 \times 0.277\ ns$

$= \boxed{9.562\ ns}$ ✓ AMAT optim mai bun!

(6)

```
        MOV R0, #0  ; initializez i
        MOV R1, #0  ; initializez m
        ADR R2, max ; aduc adresa variabilei max
        LDR R3, [R2] ; aduc valoarea lui max
loop    ADD R1, R1, R0 ; m = m+i
        MUL R4, R1, #2 ; calculam 2*max
        CMP R1, R4  ; comparam m cu 2*max
        BLE skip    ; sau la skip daca m e mai mic sau egal cu 2max
        MOV R1, R3  ; m = max
skip    ADD R0, R0, 1 ; updatez indexi
        CMP R0, R3  ; compare i with max
        BLT loop    ; if i < max continue loop
loopend ...
```