CURS 4

1.4.1. Restoring Division

5771 | 135
101 | 42

semnul scăderii

nr. 5771

COUNT	S	A	Q	M
000	0	00101101	00010110	10000111
	−	10000111		
	1	10100110	00010110	
	+	10000111		
	0	00101101		← restaurare
	0	01011010	00101100	← shiftare la stânga
001	−	10000111		
	1	11010011	00101100	
	+	10000111		
	0	01011010		
	0	10110100	01011000	
010	−	10000111		
	0	00101101	01011001	
	0	01011010	10110010	
011	−	10000111		
	1	11010011	10110010	
	+	10000111		
	0	01011010		
	0	10110101	01100100	
100	−	10000111		
	0	00101110	01100101	
	0	01011100	11001010	
101	−	10000111		
	1	11010101	11001010	
	+	10000111		
	0	01011100		
	0	10111001	10010100	
110	−	10000111		
	0	00110010	10010101	
	0	01100101	00101010	
111	−	10000111		
	1	11011110	$\underbrace{00101010}_{(42)_{10}}$	
	+	10000111		
	0	01100101		

Remainder $=(101)_{10}$ => 13 op. aritmetice

ex:

Quotient $=(22)_{10}$

```
1011 0110 | 1000
-1000     | [10110]
 00110
 -1000
 11101
+1000
 01101
 -1000
  01011
  -1000
   00110
   -1000
    1110
   +1000
   [0110]
```

↳ Remainder $= 6_{10}$

Divizor → n-bit

Quotient → n-bit

Remainder → n-bit

Divident → (2n−1) bit

1.4.2. Non-Restoring Division

$r'_{i-1} = r_{i-1} - M$

r'_{i-1} → restul parțial; $r_{i-1} - M$ → cel care trebuia să dea

$r'_i = 2r'_{i-1} = 2(r_{i-1} - M)$ → shiftare

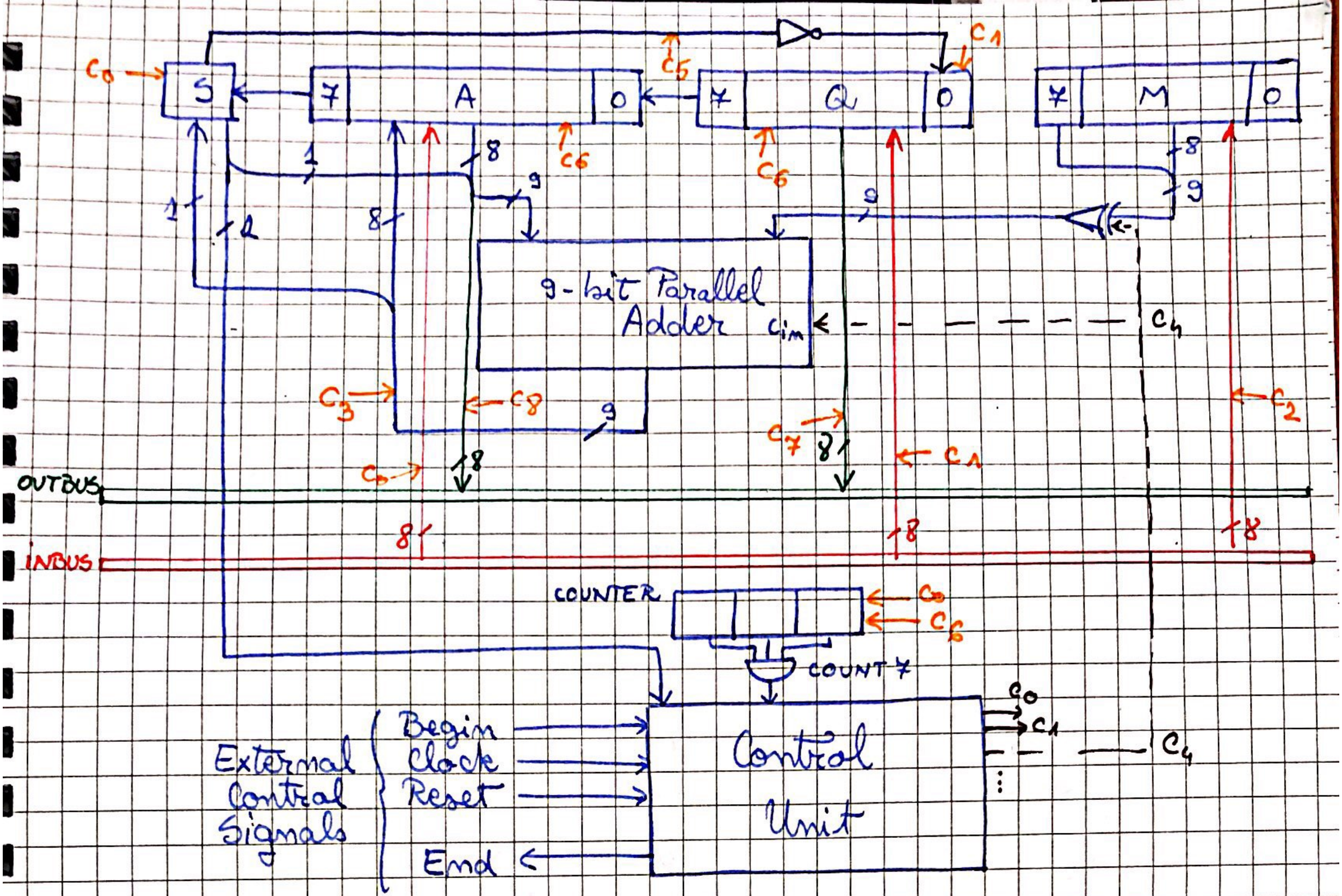
Dacă $q_i = 0$ => M nu intră în A => $r_i = 2r_{i-1} - M = r'_i + M$

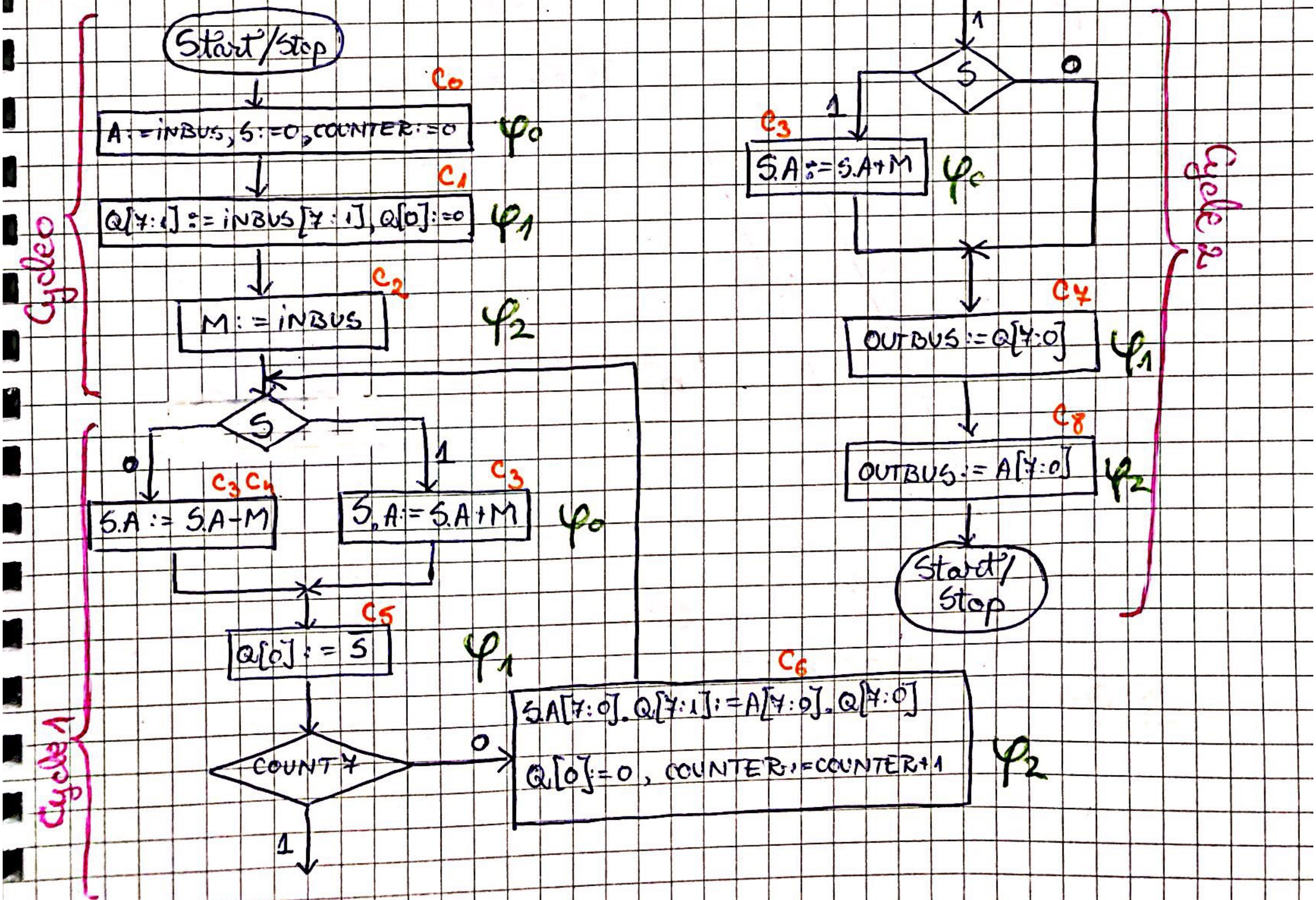
$q_i = 1$ => M intră în A => $r_i = 2(r_{i-1} - M) - M = r'_i - M$

COUNT	S	A	Q	M
000	0	00101101	00010110	100000111
	−	10000111		
	1	10100110	00010110	
	1	01001100	00101100	
001	+	10000111		
	1	11010011	00101100	
	1	10100110	01011000	
010	+	10000111		
	0	00101101	01011001	
	0	01011010	10110010	
011	−	10000111		
	1	11010011	10110010	
	1	10100111	01100100	
100	+	10000111		
	0	00101110	01100101	
	0	01011100	11001010	
101	−	10000111		
	1	1101		
	1			
110	+	10000111		
	0	00110010		
	0	01100101	00101010	
111	−	10000111		
	1	11011110	$\underbrace{00101010}_{\text{Quotient} = (42)_{10}}$	
PC	+	10000111		
	0	$\underbrace{01100101}$ Remainder = $(101)_{10}$		

PC → pas de corecție pt. că ultimul S = 1.

=> 9 op. aritmetice


S
7
A
0
7
Q
0
7
M
0
9-bit Parallel Adder
c_{in}
OUTBUS
INBUS
COUNTER
COUNT 7
External Control Signals
Begin
Clock
Reset
End
Control Unit


Cycle 0
Start/Stop
A:=INBUS, S:=0, COUNTER:=0
φ_0
Q[7:1]:=INBUS[7:1], Q[0]:=0
φ_1
M:=INBUS
φ_2
Cycle 1
S.A:=S.A-M
S.A:=S.A+M
φ_0
Q[0]:=S
φ_1
COUNT 7
S.A[7:0].Q[7:1]:=A[7:0].Q[7:0]
Q[0]:=0, COUNTER:=COUNTER+1
φ_2
Cycle 2
S.A:=S.A+M
φ_0
OUTBUS:=Q[7:0]
φ_1
OUTBUS:=A[7:0]
φ_2
Start/Stop

1.4.3. Algoritmul S.R.T.

Reprezentare pentru:

- radix 4: $\{\bar{2}, \bar{1}, 0, 1, 2\}$
- radix 8: $\{\bar{4}, \bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3, 4\}$
- radix 16: $\{\bar{8}, \bar{7}, \bar{6}, \ldots, 6, 7, 8\}$
- radix 2: $\{\bar{1}, 0, 1\}$

$$10\bar{1}0\bar{1} = -1\cdot 2^0 + 0\cdot 2^1 - 1\cdot 2^2 + 0\cdot 2^3 + 1\cdot 2^4 = 11$$

Codificare: $1 \to \begin{smallmatrix}1\\0\end{smallmatrix}$; $0 \to \begin{smallmatrix}0\\0\end{smallmatrix}$; $\bar{1} \to \begin{smallmatrix}0\\1\end{smallmatrix}$ =>

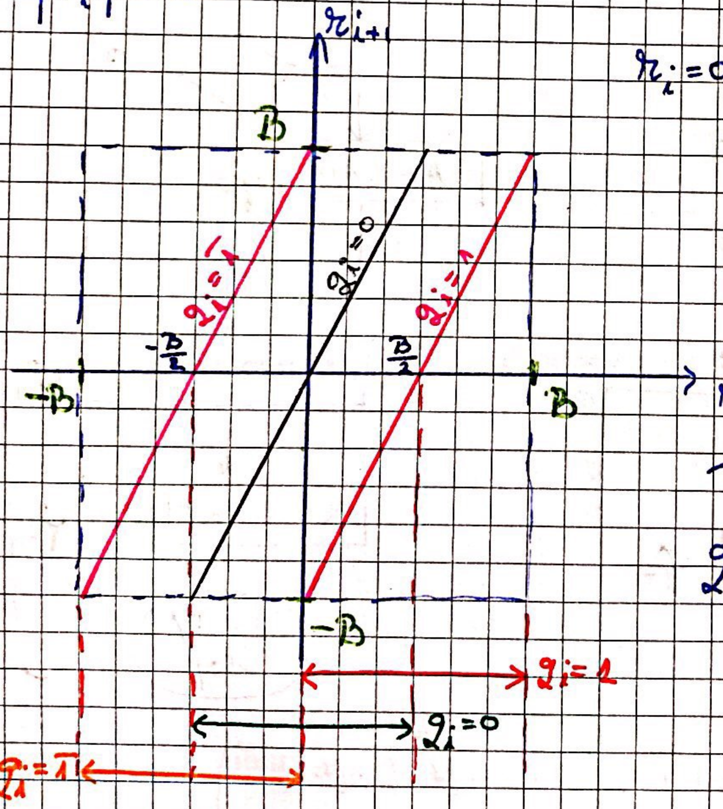
$$\Rightarrow \begin{array}{r} 10000 - \\ 00101 \\ \hline 01011 \end{array} = (11)_{10}$$

$r_{i+1} \leftarrow 2r_i - q_i \cdot B$

$q_i \in \{\bar{1}, 0, 1\}$

$|r_i| < B$

Aici: $\begin{cases} A \to P \\ Q \to A \\ M \to B \end{cases}$



$r_i = 0 \Rightarrow r_{i+1} = 2B - B = B \quad q_i = 1$

$r_{i+1} = 2r_i - B = 0 \Rightarrow$

$\Rightarrow r_i = \frac{B}{2}$

$r_{i+1} = -B = 2r_i - B \Rightarrow$

$\Rightarrow r_i = 0$

$q_i = -1$:

$M_{i+1} = B \Rightarrow M_i = 0$

$M_{i+1} = 0 \Rightarrow M_i = -B/2$

$M_{i+1} = -B \Rightarrow M_i = -B$

Algoritmul lui SRT: (radix 2) 2^k Divident 2^k Divisor

(1.)

P	A	B
$\longleftarrow n+1 \longrightarrow$	$\longleftarrow n \longrightarrow$	$\longleftarrow n \longrightarrow$

If B has k leading 0's (zeros), shift P.A.B k positions left

(2.) For $i=0$ to $n-1$

If top 3 bits of register P are equal then $q_i=0$; Shift P.A left

If top 3 bits of P are not equal and $P \geq 0$ then:

$q_i = 1$; Substract B from P; Shift P.A left

If top 3 bits of P are not equal and $P < 0$ then:

$q_i = \bar{1}$; Add P to B; Shift P.A left

(3.) If $P < 0$ Add B to P; Substract 1 from A

(4.) Shift P right k positions