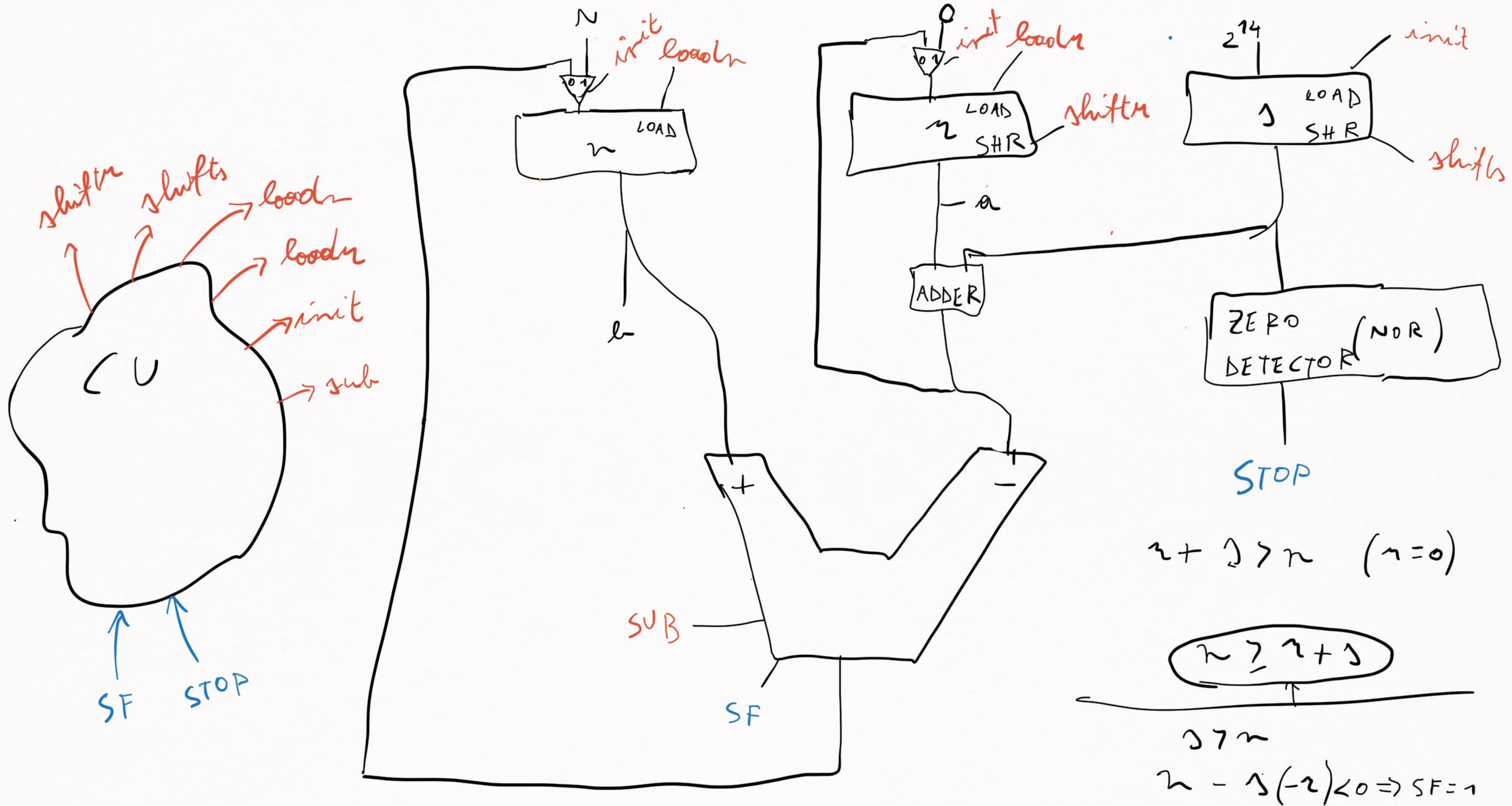
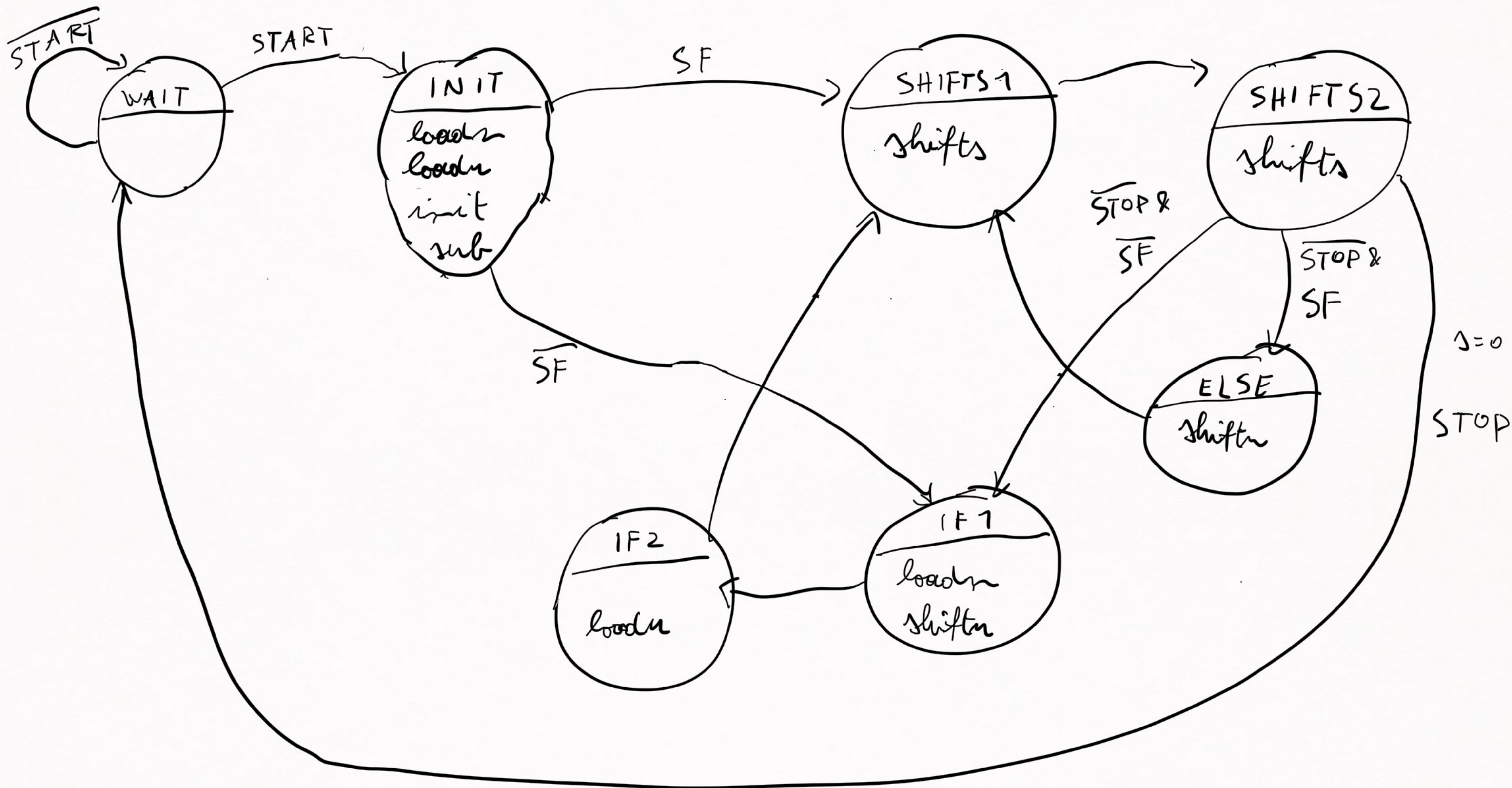
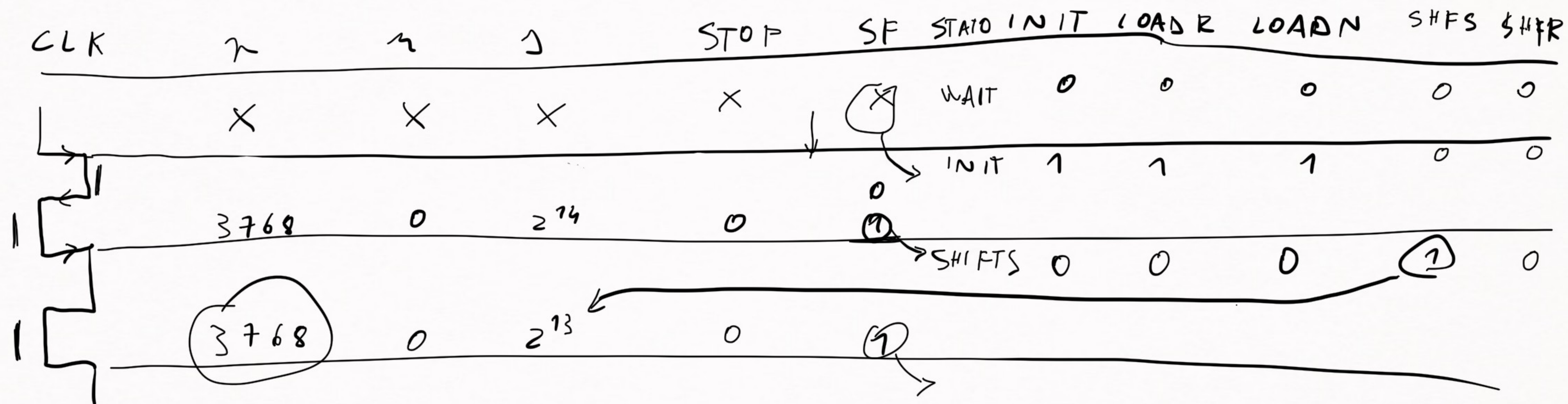


COMPITO 17/06/2019







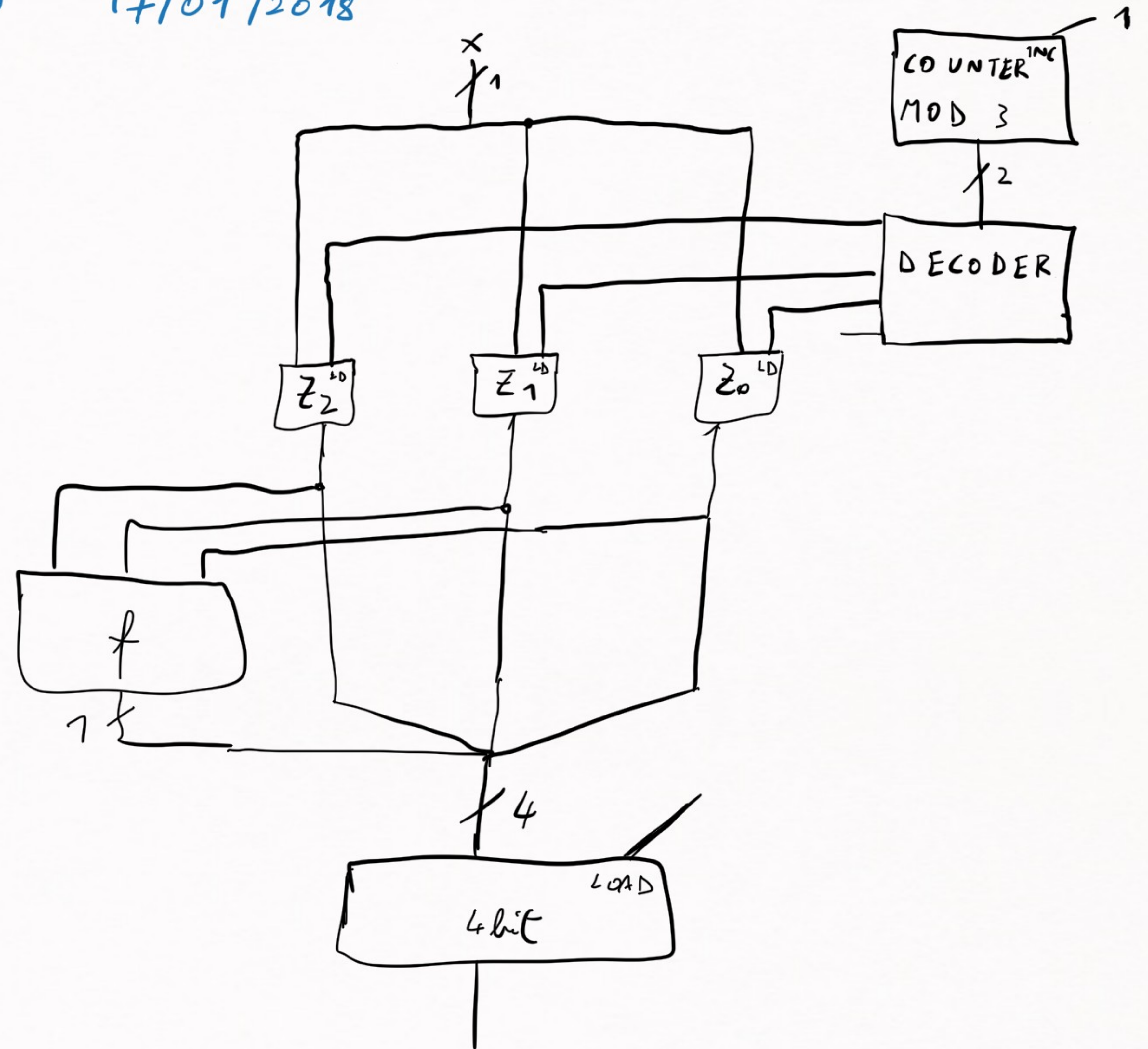
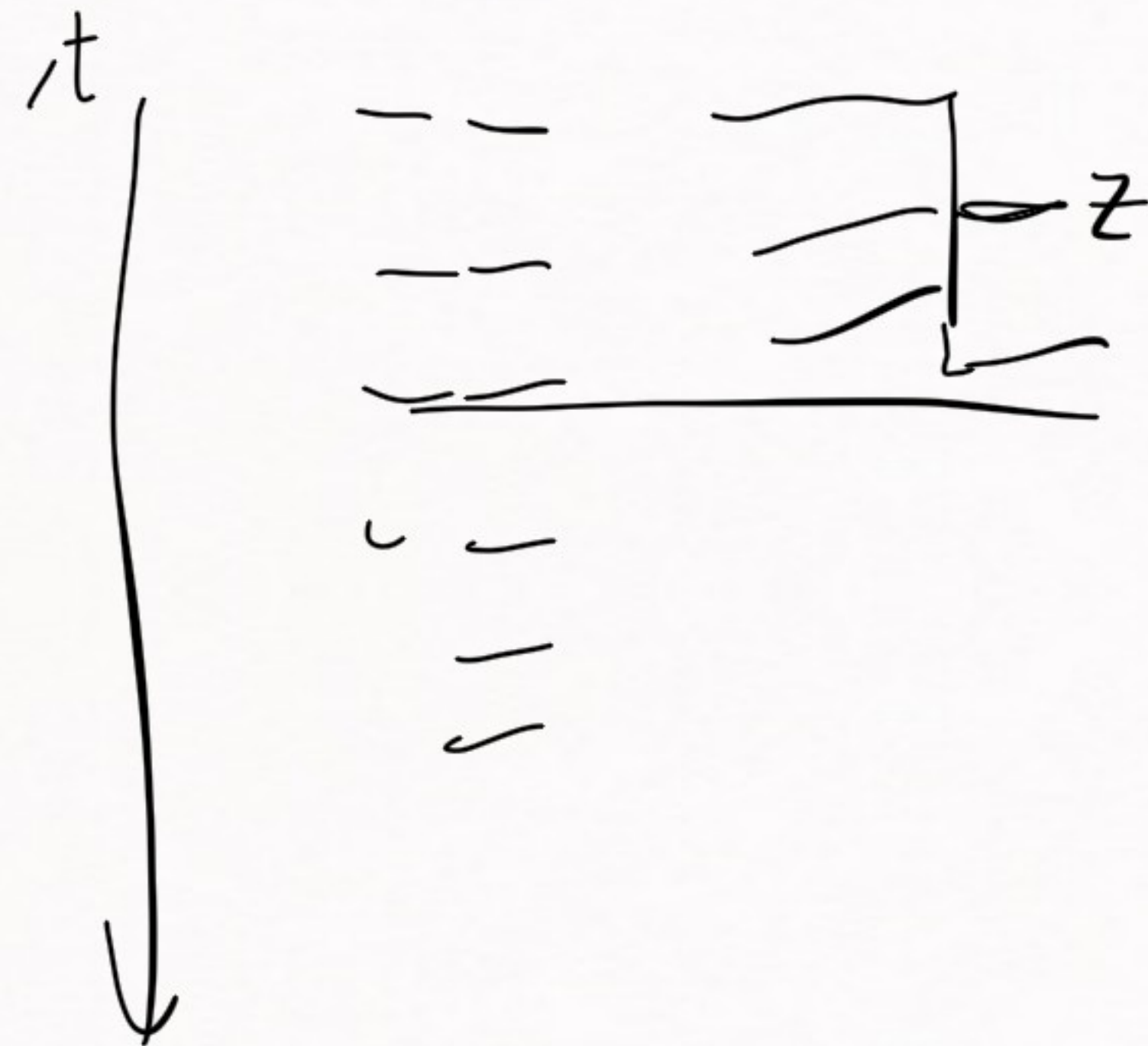
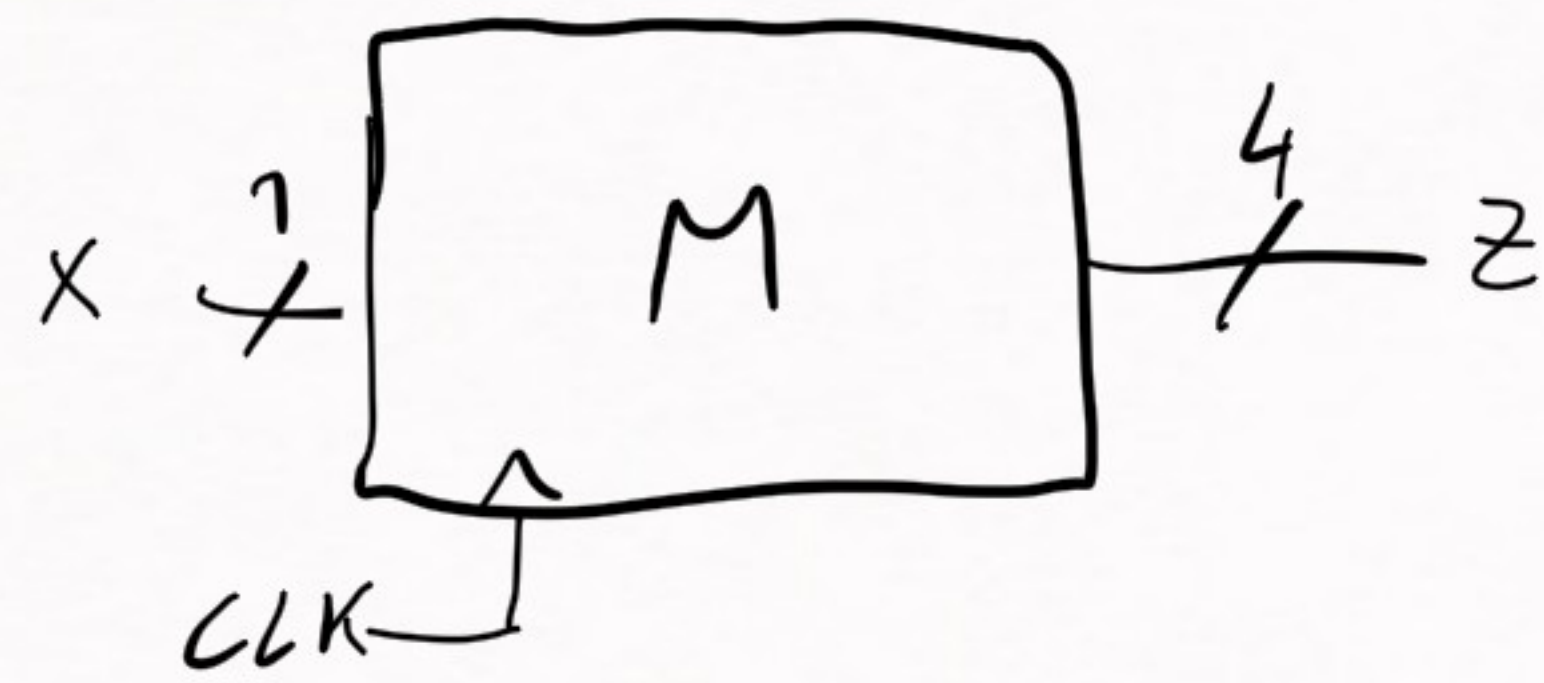
2^{12}

:

2^{10}

0

COMPITO 17/01/2018



BANCO MEMORIA

$$S = 512M \times 32$$

$$\rightarrow 128M \times 32 = 2$$

$$\rightarrow 128M \times 16 = 4$$

$$2^{29}$$

00	1	
01	2	
10	3	4
11	5	6

$$32$$

$$A_{28} A_{27} \dots A_1 A_0$$

$$CS_1 = \overline{A_{28}} \overline{A_{27}}$$

$$CS_2 = \overline{A_{28}} A_{27}$$

$$CS_3 = CS_4 = A_{28} \overline{A_{27}}$$

$$\frac{17}{28} = \frac{17}{28} \quad \begin{array}{c} \text{34-28} \\ \downarrow \end{array} \quad \begin{array}{c} 17 \quad 6 \quad 12 \quad 24 \quad 20 \quad 12 \\ \hline \end{array} \quad \begin{array}{c} h=2 \\ k=3 \end{array}$$

A	1
B	2
C	1
D	1

```

visti db 0 dup(0);
mov cl, 1 // t
mov bl, p ←
xor bh, bh
mov visti[BX], cl
cycle: inc cl
shl BX, 1
cmp BX, 9
jl update
sub BX, 9
update: mov dx, visti[BX]
cmp dx, 0
jre end

```

```

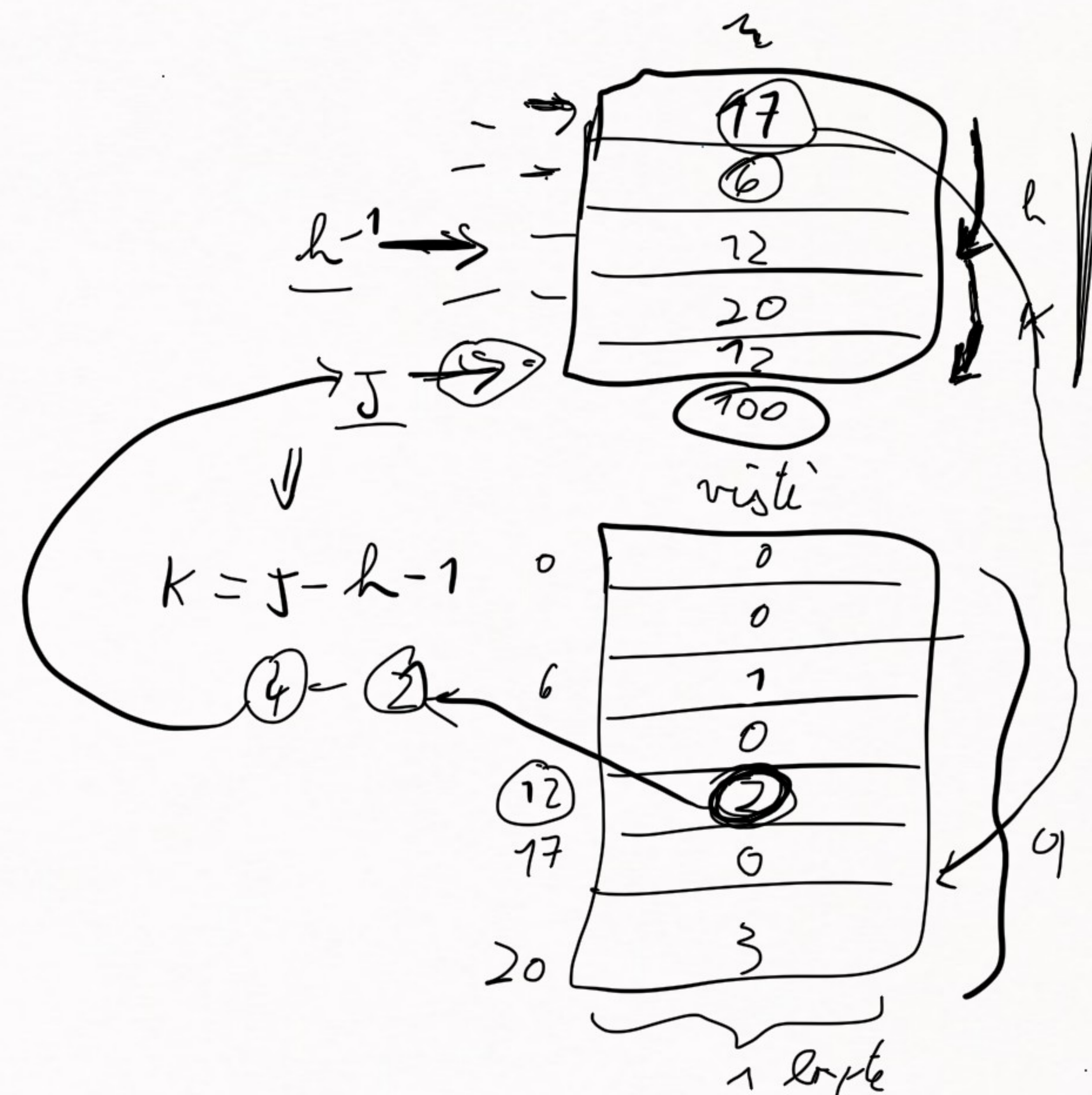
mov visti[BX], cl
jmp cycle

```

```

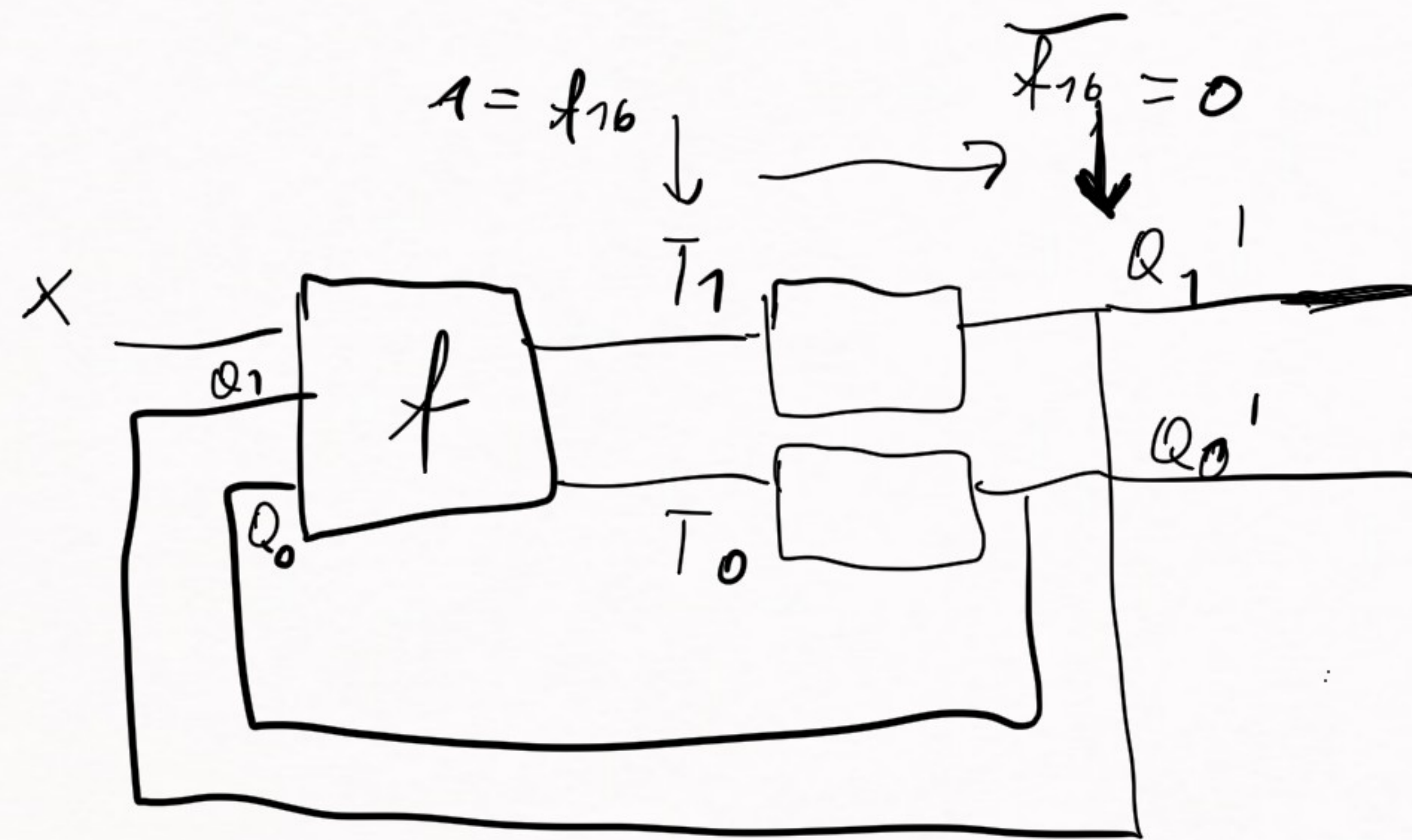
end: sub dx, 1
mov ax, dx
sub cl, ax
sub cl, 1
mov ah, cl

```



SINTESI

VINCOLATA



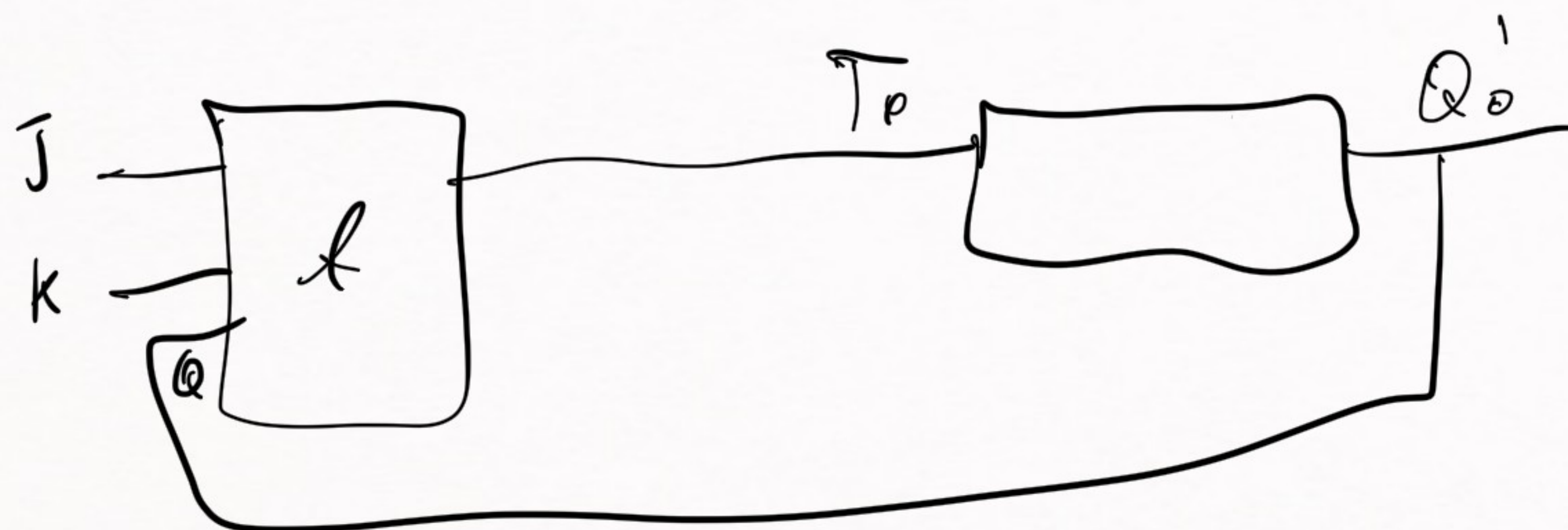
obiettivo $\begin{cases} Q_1' = X Q_0 + \bar{X} Q_1 \leftarrow \\ Q_0' = \bar{X} + \bar{Q}_0 \leftarrow \end{cases}$

vincolo $\begin{cases} Q_1' = T_1 \oplus Q_1 = T_1 \bar{Q}_1 + \bar{T}_1 Q_1 \\ Q_0' = T_0 \oplus Q_0 \end{cases}$

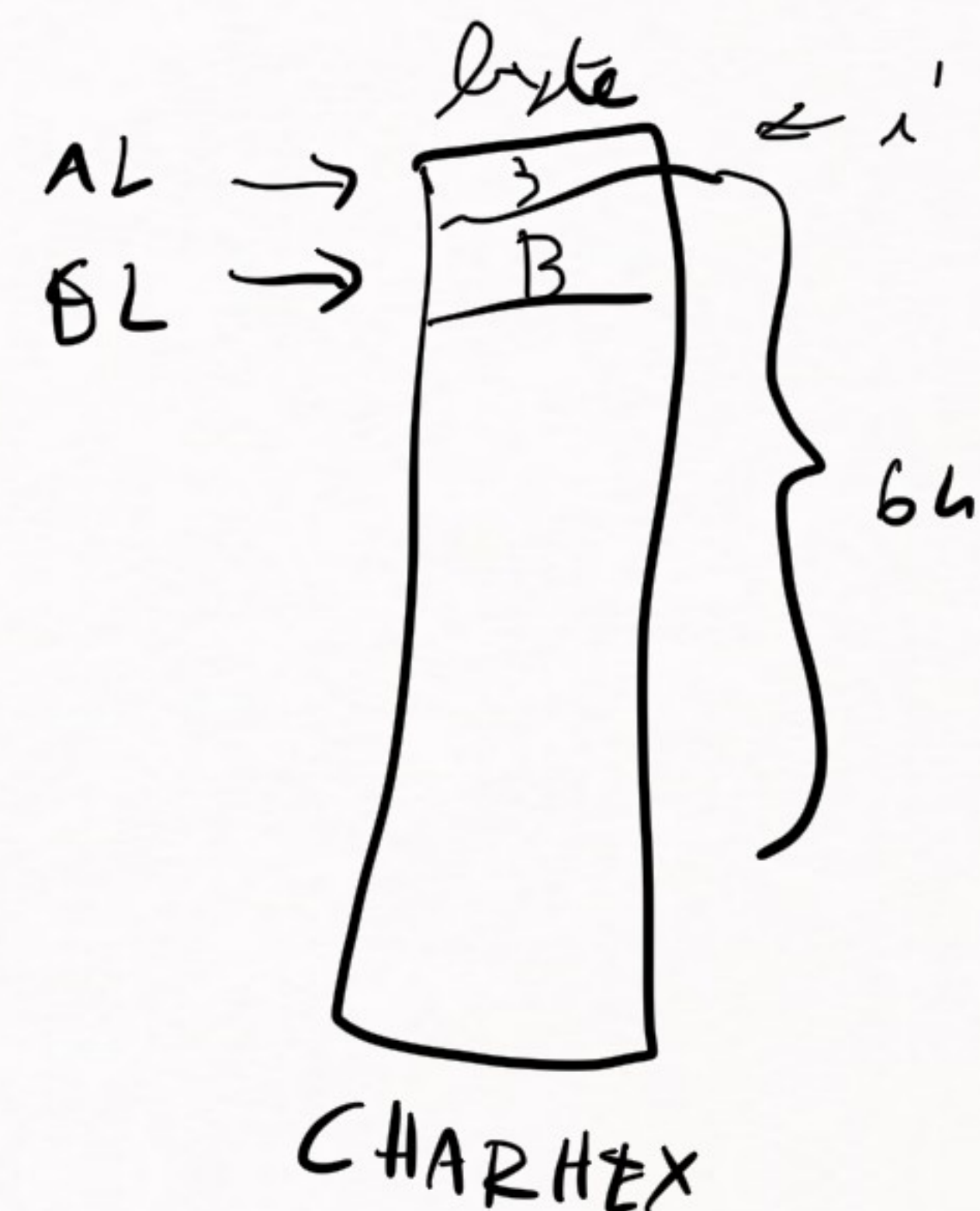
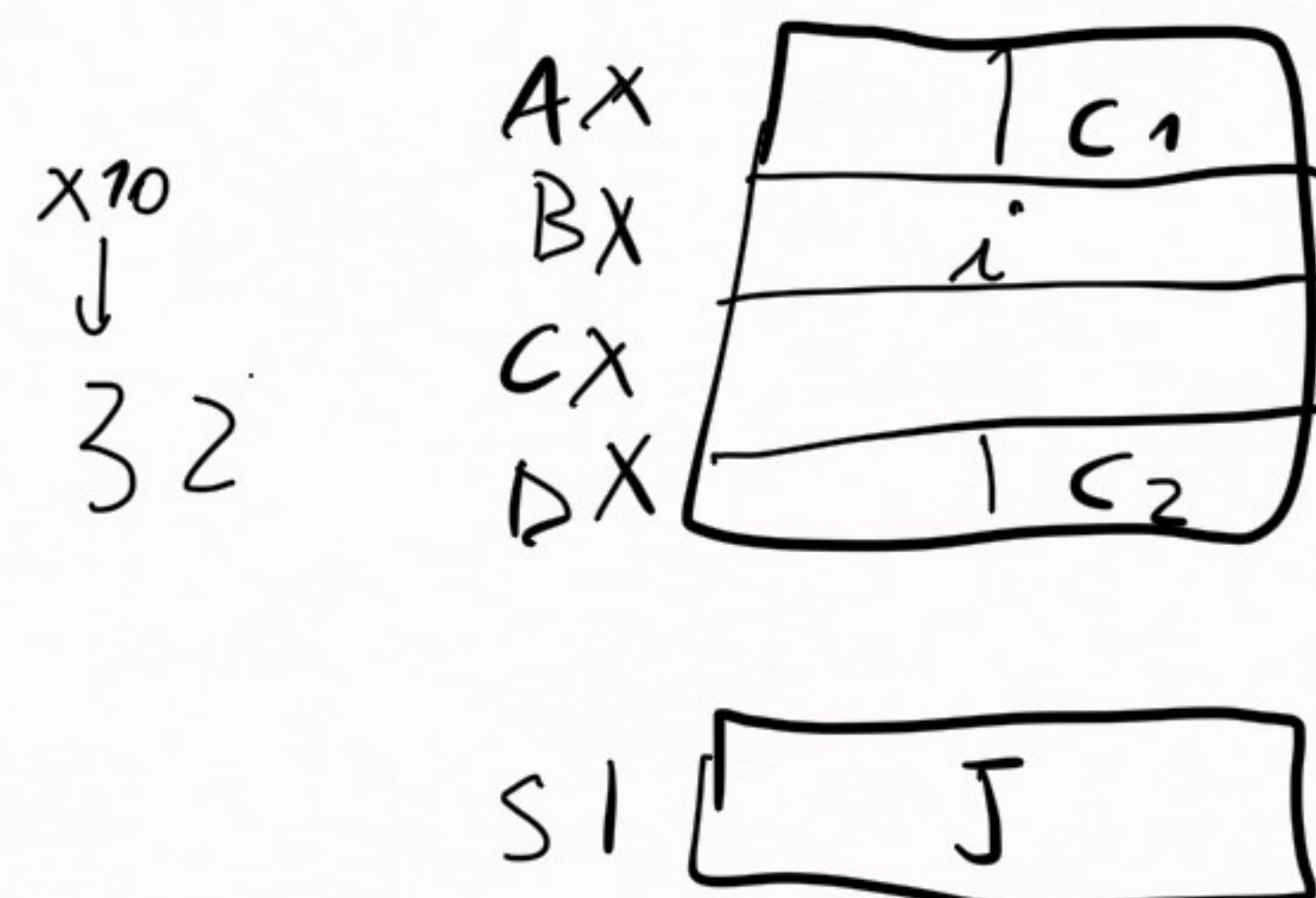
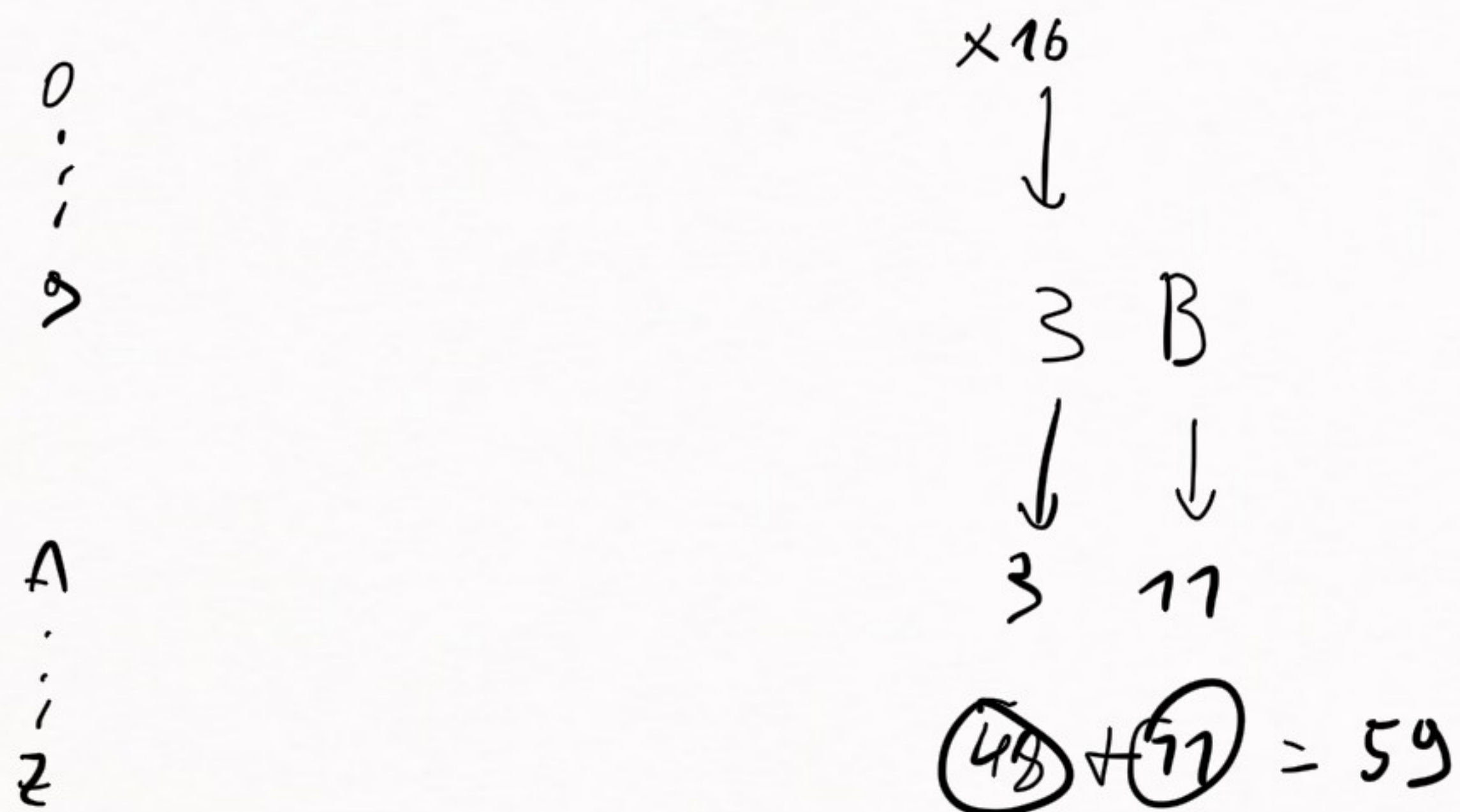
$f \begin{cases} f_1 \Rightarrow T_1 = f_{10} \bar{X} \bar{Q}_1 \bar{Q}_0 + f_{11} \bar{X} \bar{Q}_1 Q_0 + \dots \\ f_0 \Rightarrow T_0 = f_{00} \bar{X} \bar{Q}_1 \bar{Q}_0 + f_{01} \bar{X} \bar{Q}_1 Q_0 + \dots \end{cases}$

$X Q_1 Q_0$	$X Q_0 + \bar{X} Q_1$	T_1	Q_1'	f_1
0 0 0	0	f_{10}	f_{10}	0 = f_{10}
0 0 1	0	f_{11}	f_{11}	0
0 1 0	1	f_{12}	$\bar{f}_{12}$	0
0 1 1	1	f_{13}	$\bar{f}_{13}$	0
1 0 0	0	f_{14}	f_{14}	0
1 0 1	1	f_{15}	f_{15}	1
1 1 0	0	f_{16}	$\bar{f}_{16}$	1
1 1 1	1	f_{17}	$\bar{f}_{17}$	0

$f_1 = 0 \cdot \bar{X} \bar{Q}_1 \bar{Q}_0 + \dots =$
 $= (f_{15} X \bar{Q}_1 Q_0 + f_{16} X Q_1 \bar{Q}_0)$



$$Q_0^1 = J \bar{Q} + \bar{K} Q$$



Assembly code:

```

xor    BX, BX
xor    SI, SI

cycle: xor    AL, CHAR HEX[BX]
inc    BX
xor    DL, CHAR HEX[BX]

cmp    AL, 'A'
jl     number
sub    AL, 'A'
add    AL, 10
jmp    mult1
number: sub    AL, '0'

mult1: shl    AL, 4 // if BX dispari

```

~~DL~~

add AL, DL
xor INT HEX[SI], AL
inc SI
jmp cycle
else
shl AL, 4

