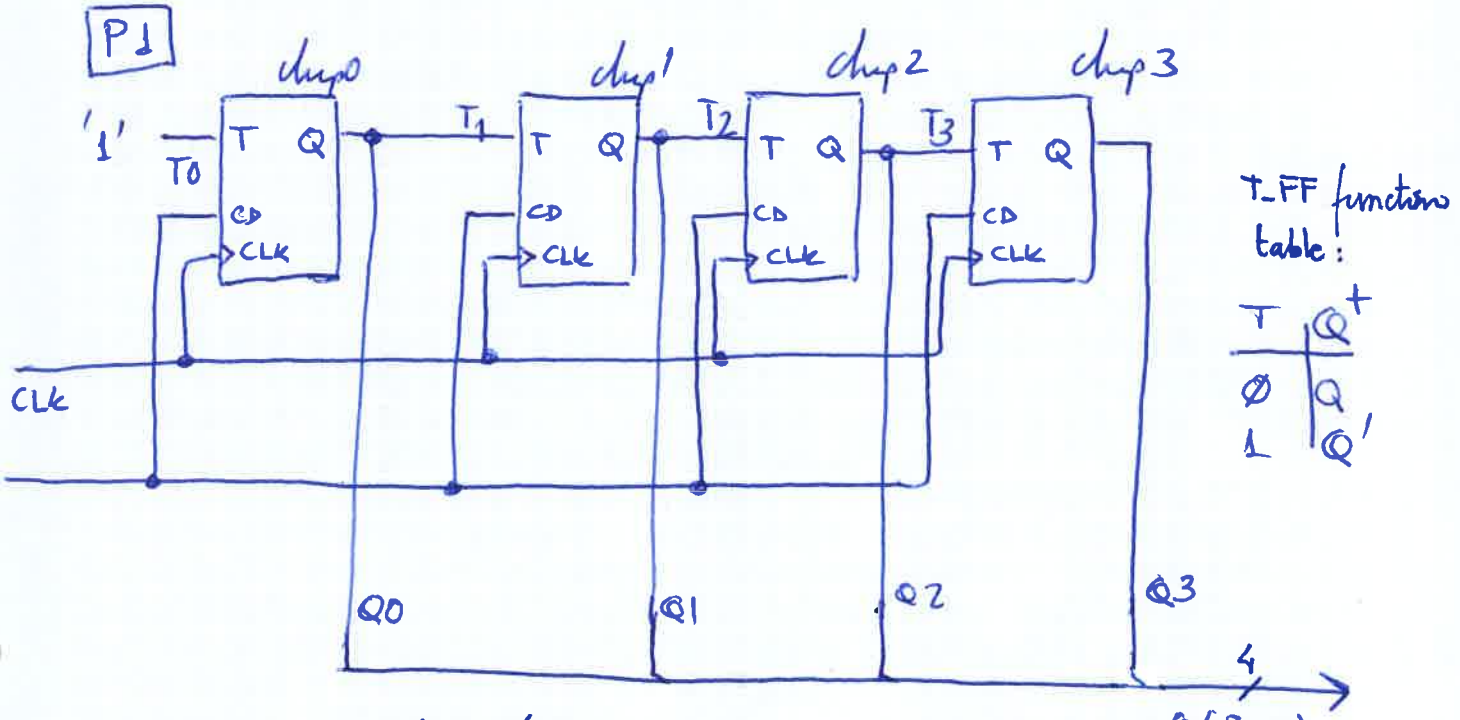
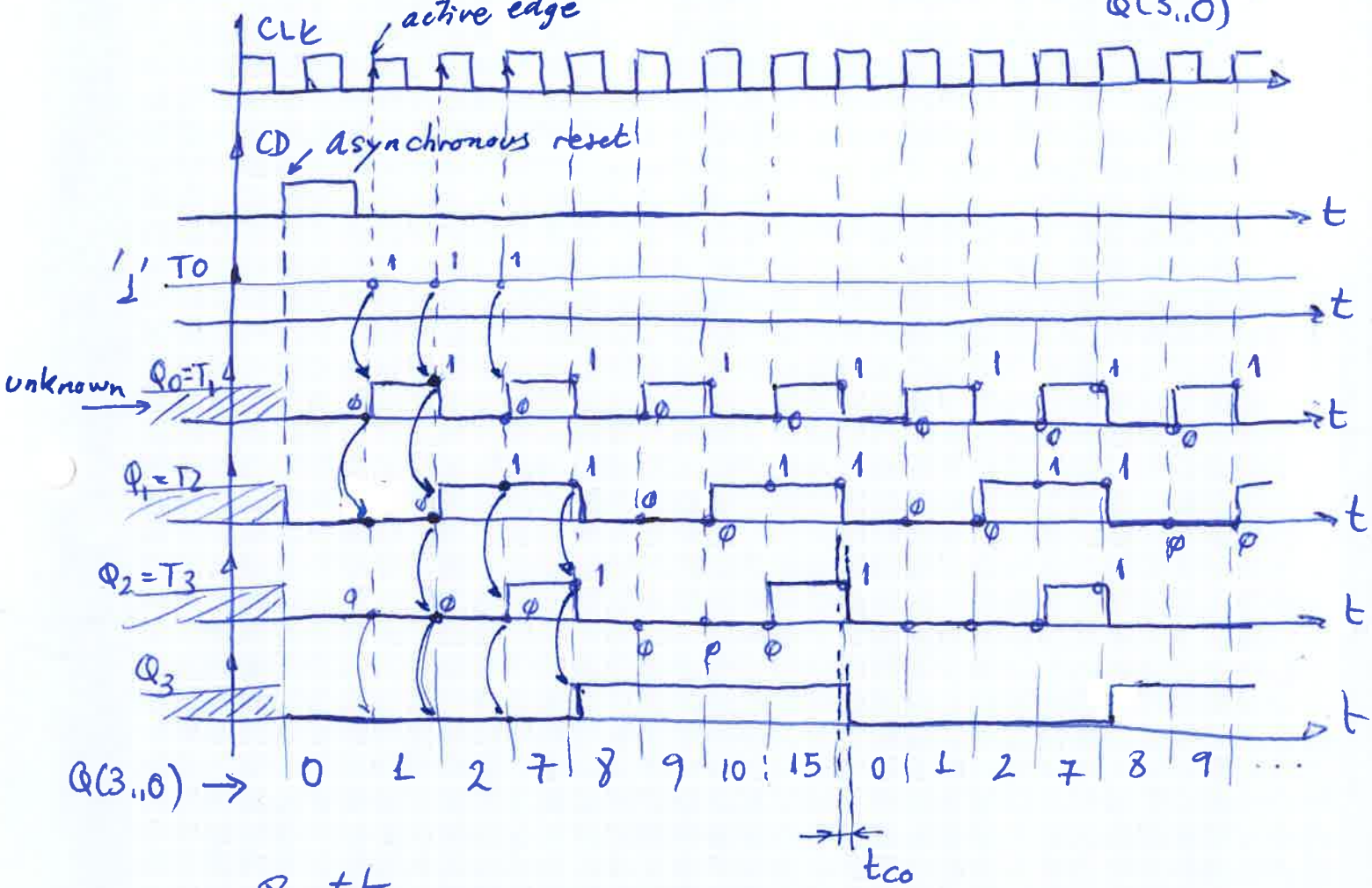
P1



T_FF function table:

T	Q^+
0	Q
1	Q'

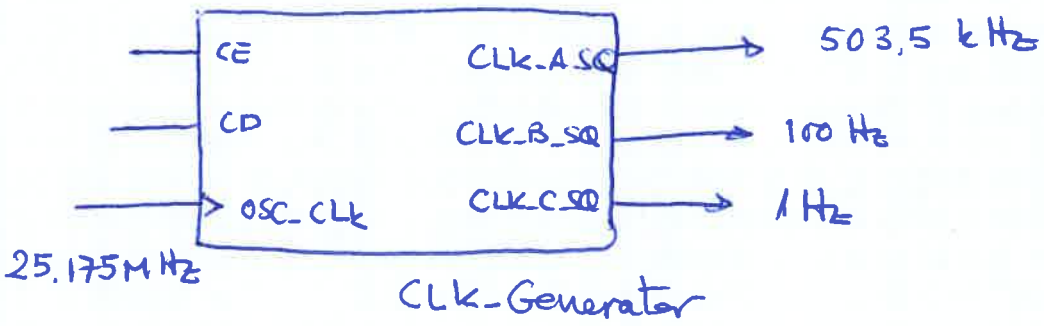


b) → 8 states

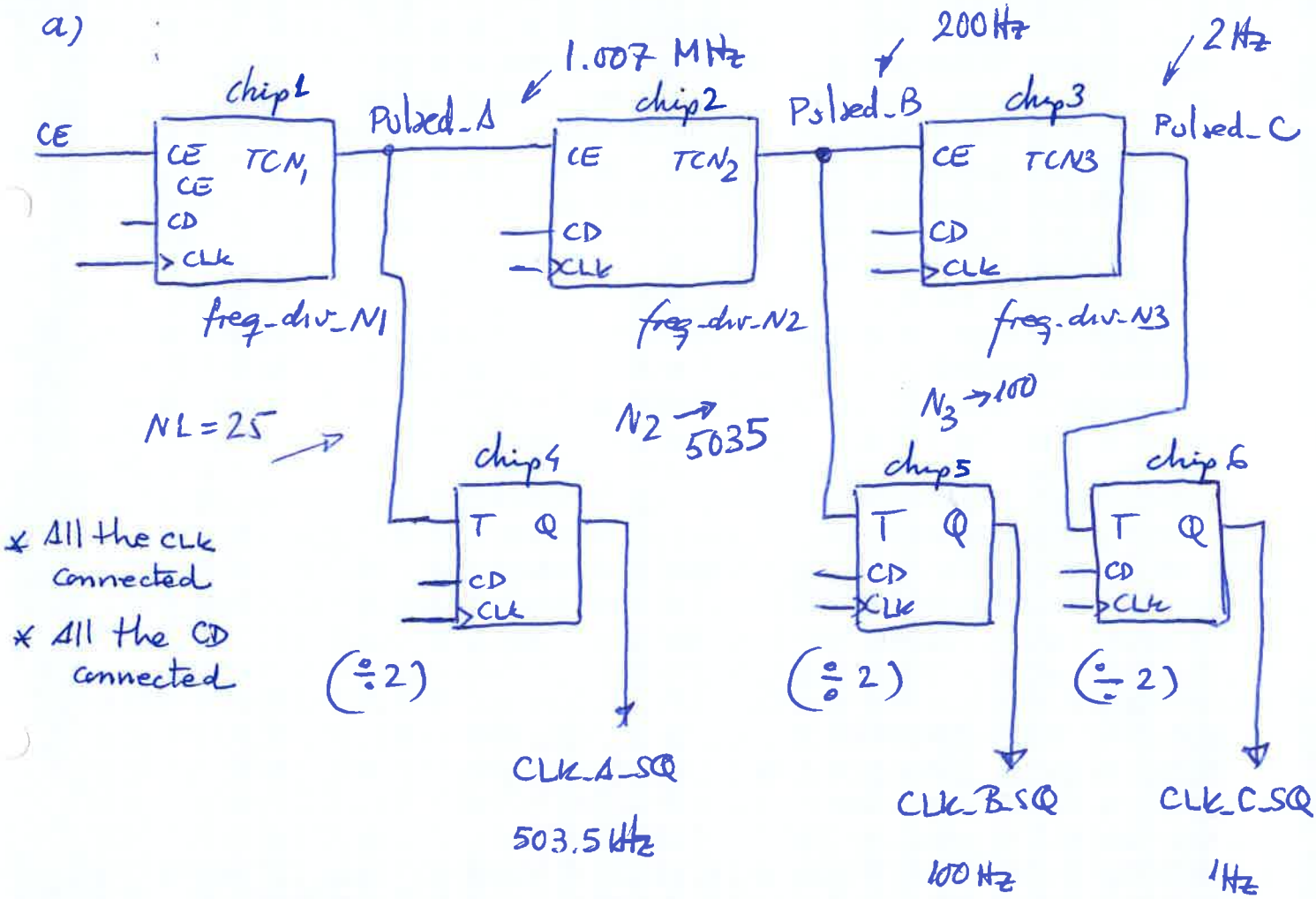
→ It is a synchronous counter that generates the numbers 0, 1, 2, 7, 8, 9, 10, 15, 0, ... running continuously

→ $Q_0 \rightarrow \frac{f_{CLK}}{2}$, $Q_1, Q_2 \rightarrow \frac{f_{CLK}}{4}$, $Q_3 \rightarrow \frac{f_{CLK}}{8}$; $Q_2 \rightarrow$ DC = 25%

P2



a)



$$2 \cdot N_1 = \frac{25.175\ MHz}{503.5\ kHz} = 50 \rightarrow N_1 = 25$$

$$2N_2 = \frac{1.007\ MHz}{100\ Hz} = 10070 \rightarrow N_2 = 5035$$

$$2N_3 = \frac{200\ Hz}{1\ Hz} = 200 \rightarrow N_3 = 100$$

NOTEBLOC®

b) Clk_Generator.vhd (top)

freq_div_25.vhd
freq_div_5035.vhd
freq_div_100.vhd
T_FF.vhd
} components

T_FF requires 1 D_FF (2 states) → 3

N = divider modulo (number of states) $\leq 2^n$ ← number of bits of the counter

$$n \geq \left(\frac{1}{\log 2}\right) \log N$$

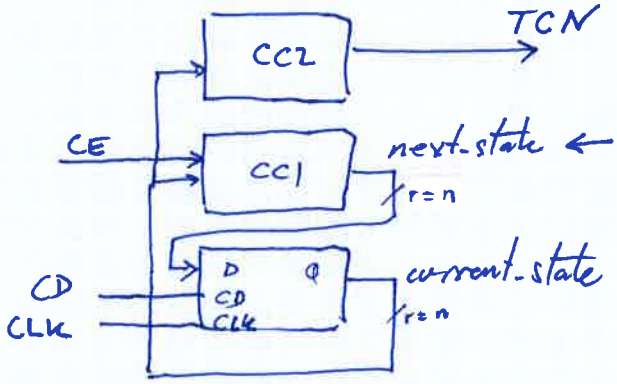
$N_3 = 100$ $n = 7$ → 7

$N_2 = 5035$ $n = 13$ → 13

$N_1 = 25$ $n = 5$ → 5

Number of D_FF → 28

c) freq_divN is a FSM solved using the plan Y (STD_LOGIC_VECTO



Arithmetic library in VHDL

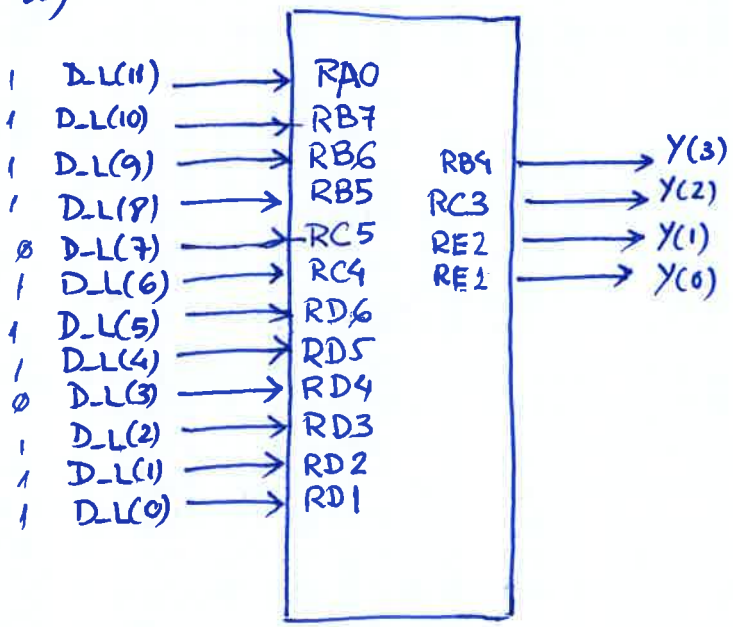
next_state ← current_state + 1; (Adder)

TCN = '1' when current_state is Max_Count and CE = '1' else '0'

NOTEBLOC®

P3 Example of read and write in the ENC_124

a)

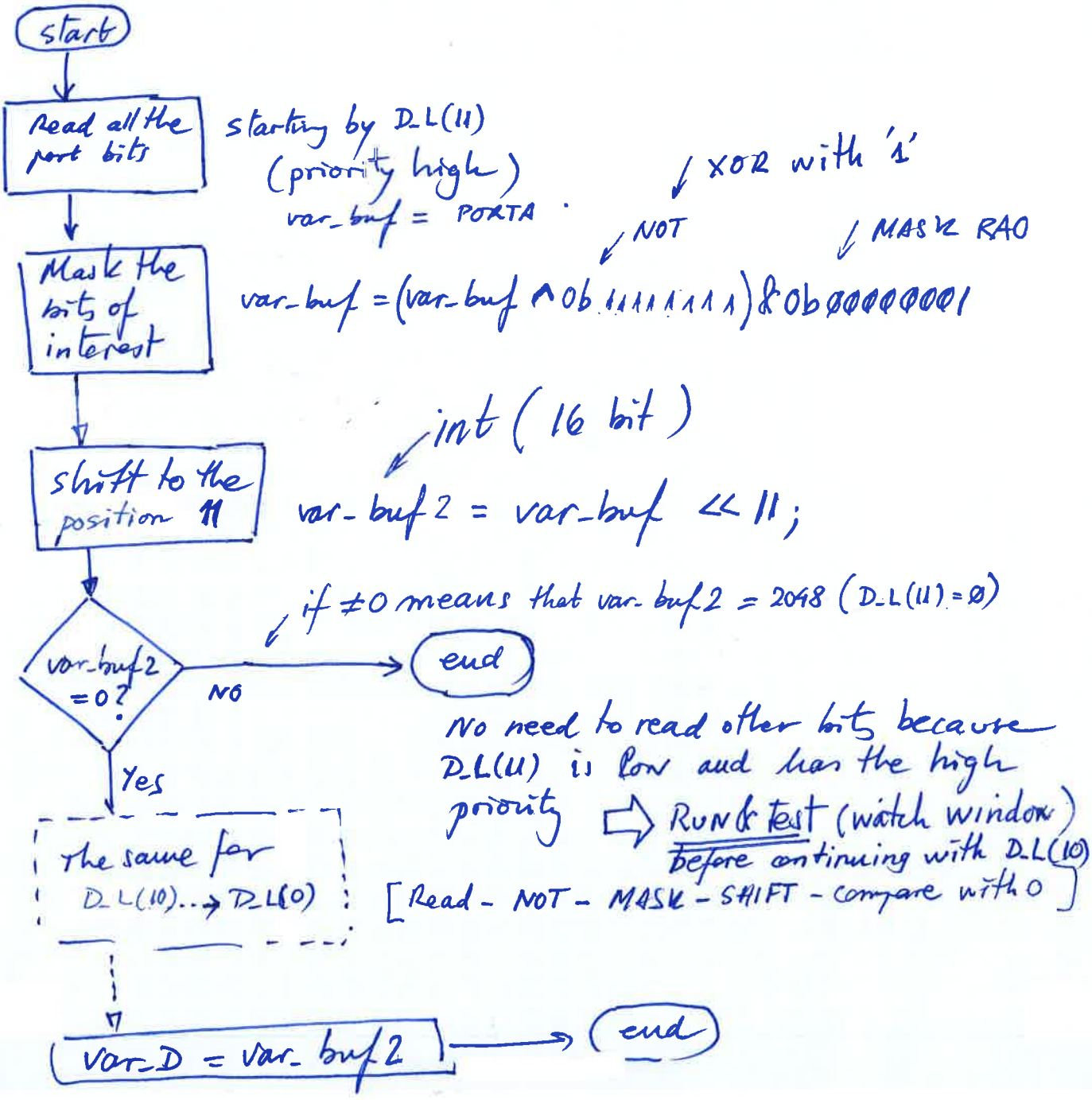


- Objective when reading

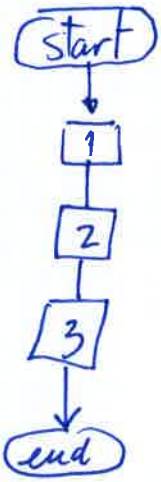
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

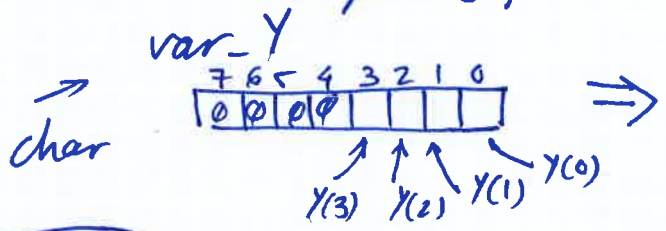
var_D active high

→ When clicking a key a '1' is set in the corresponding bit

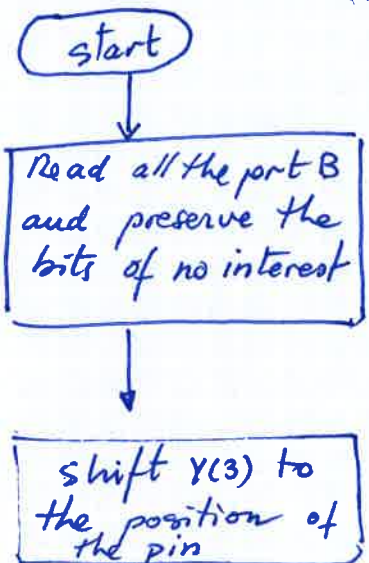


b) write objective is to send each variable bit to the corresponding pin



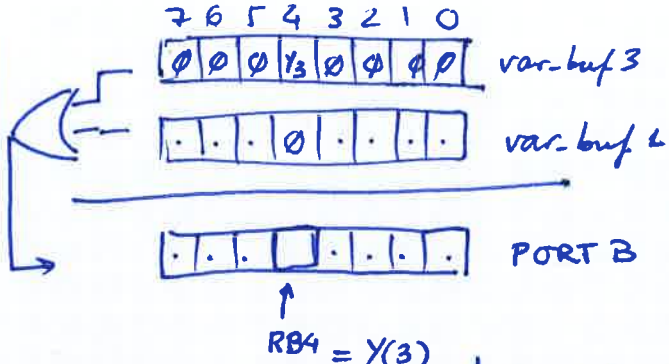


Y(3) → RB4
Y(2) → RC3
Y(1) → RE2
Y(0) → RE1




```
var_buf = PORTB;
var_buf = var_buf & 0b11101111;
```

```
var_buf3 = (var_Y & 0b00001000) << 1;
```




```
PORT B = var_buf2 | var_buf3;
```

OR

Run & test (watch window and LED) before continuing with Y(2), etc.

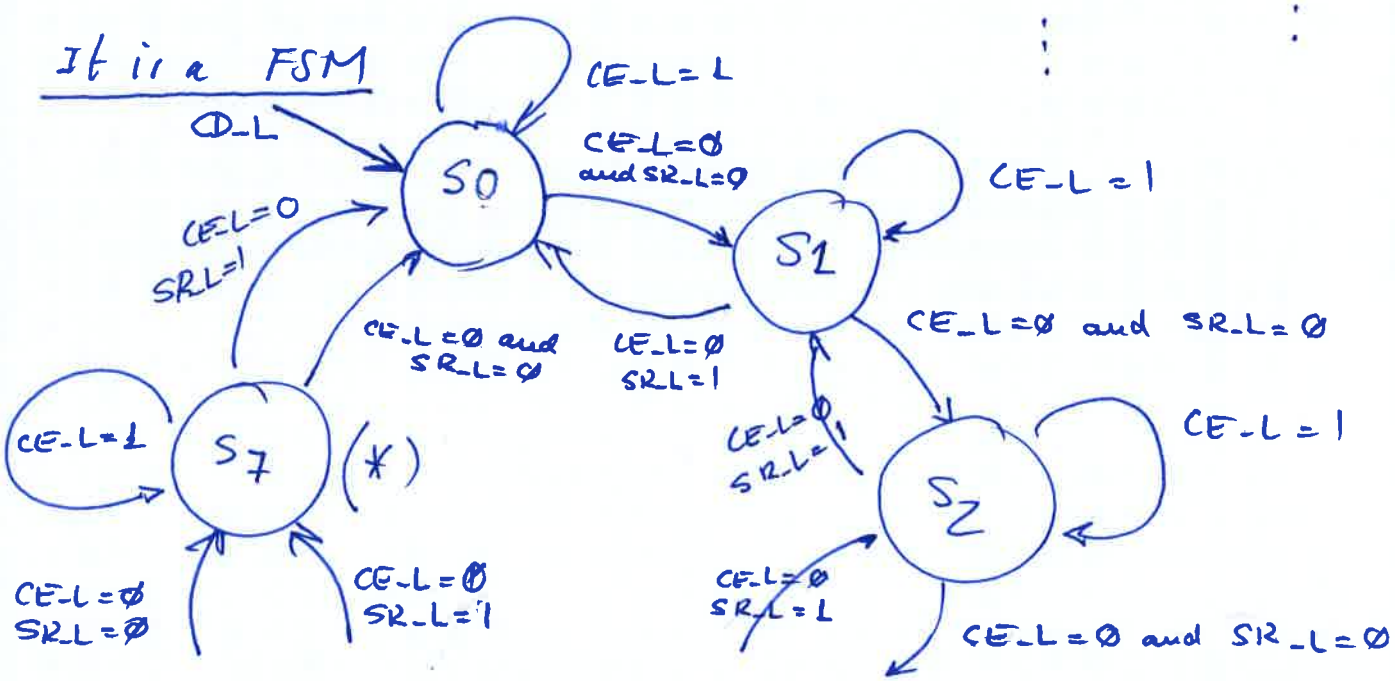
P4

a)

CE_L	SR_L	LED+
1	X	LED do nothing
0	0	rotate → (right)
0	1	rotate ← (left)

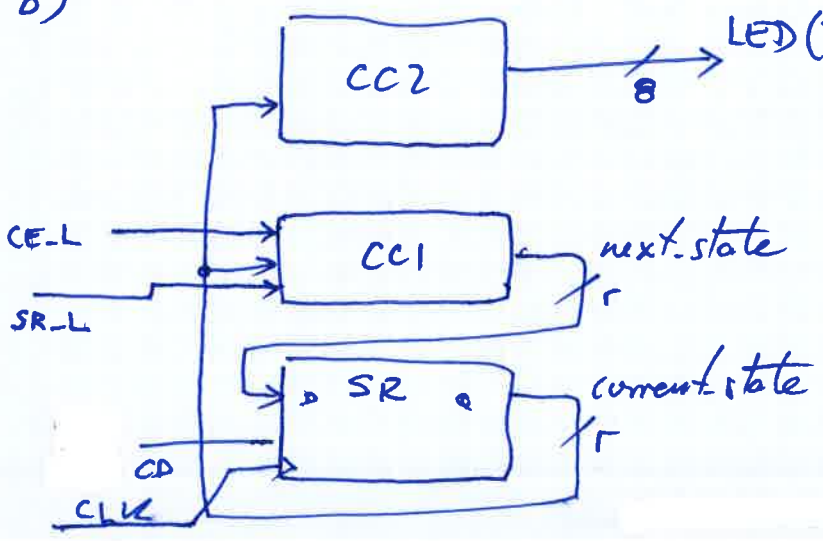
1 0 0 0 0 0 0 0 0 S_0
0 1 0 0 0 0 0 0 0 S_1
0 0 1 0 0 0 0 0 0 S_2
⋮ ↓
0 0 0 0 0 0 0 1 S_7
1 0 0 0 0 0 0 0 0 S_0
⋮ ⋮

It is a FSM



(*) outputs are indicated above. It is a one-hot code where the '1' is shifted to the right or the left and then starts again

b)



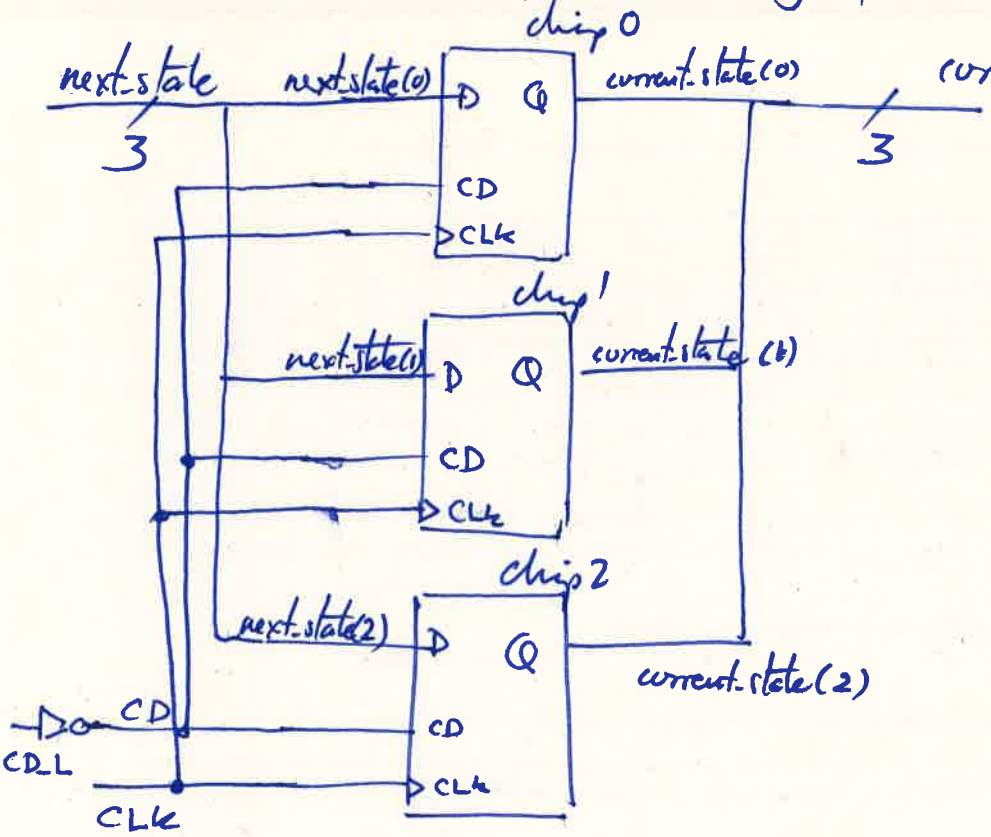
D_FF=3

8 states → r=3 in binary

CC2 will be a code converter from binary to one-hot

CC1 will count in binary up and down

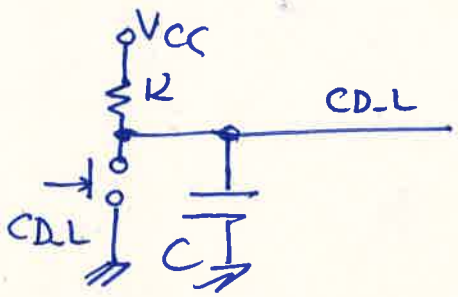
Example drawing of the state register (hardware) structural description



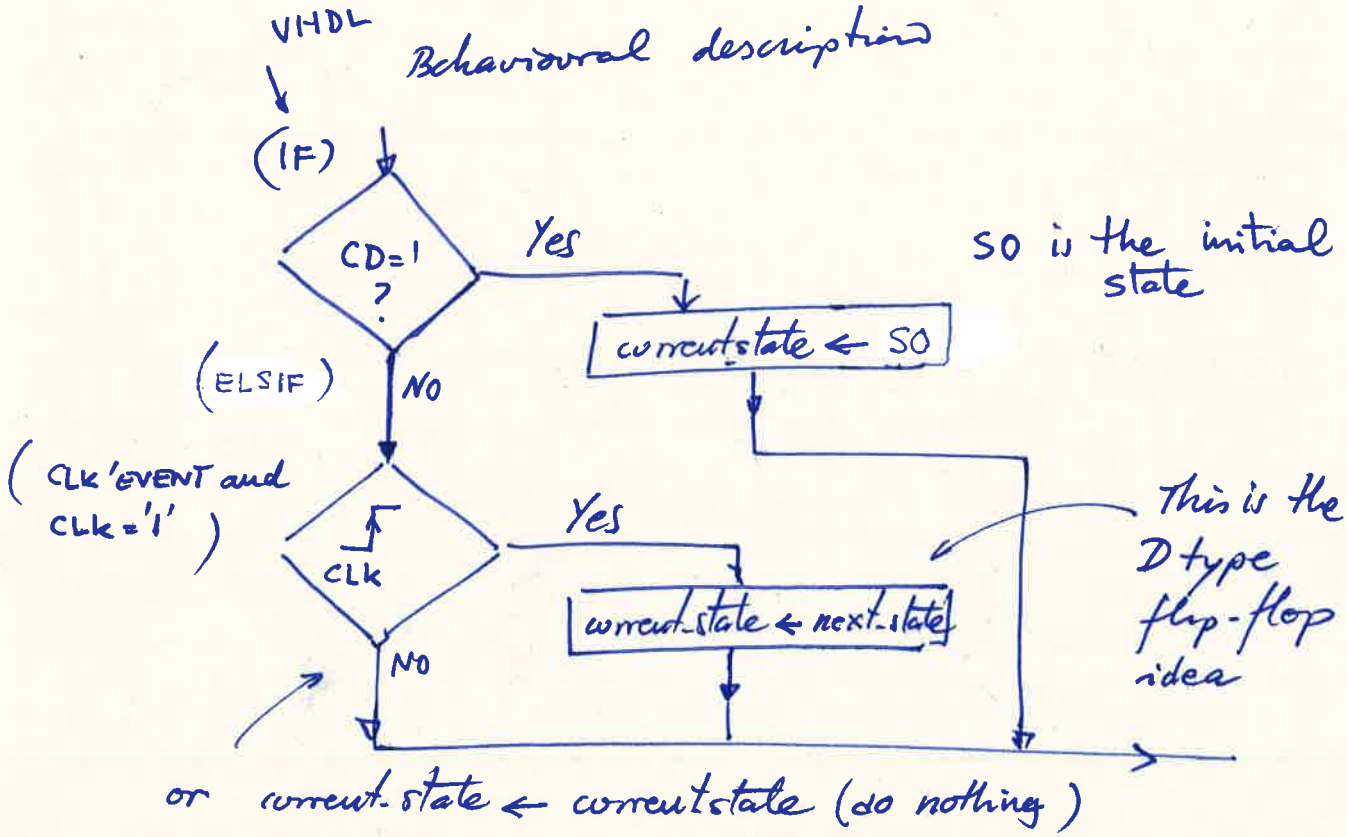
→ This circuit is written as a process in VHDL

↑ Behavioural description

when CD_L = '0' → CD = 1 → current_state = "000"



which is converted by the CC2 truth table to "10000000" LED(7..0)



c) + d)

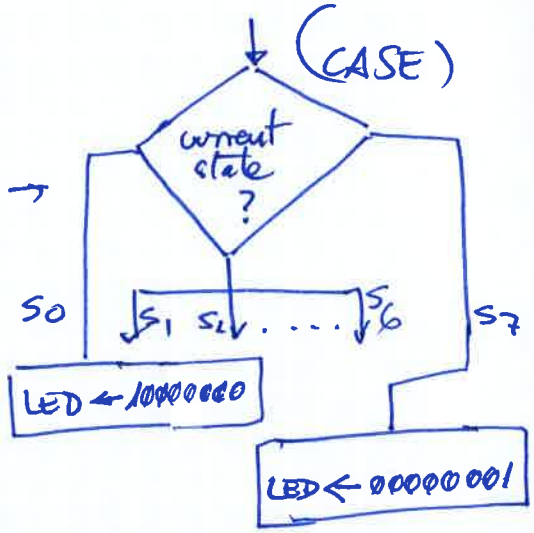
CC1 truth table

CE_L	SRL	current_state	next_state
1	X	current_state	current_state
0	0	S_0	S_1
0	0	S_1	S_2
⋮	⋮	⋮	⋮
0	0	S_7	S_0
0	1	S_0	S_7
0	1	S_1	S_0
⋮	⋮	⋮	⋮
0	1	S_7	S_6

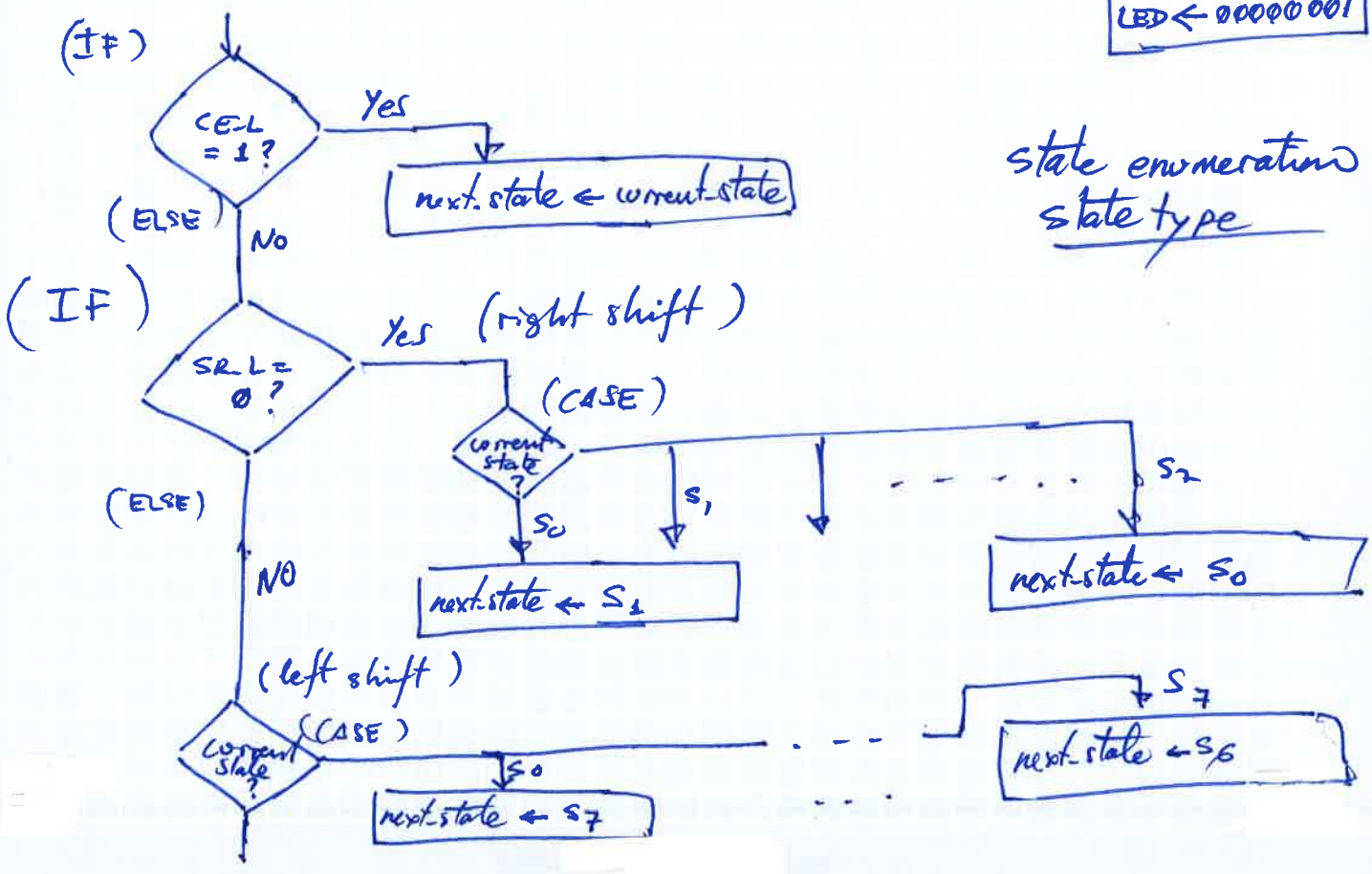
→ all the loops disabling the shifting

CC2 truth table

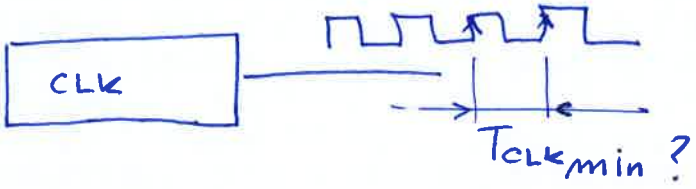
current_state	LED
S_0	10000000
S_1	01000000
⋮	⋮
S_7	00000001



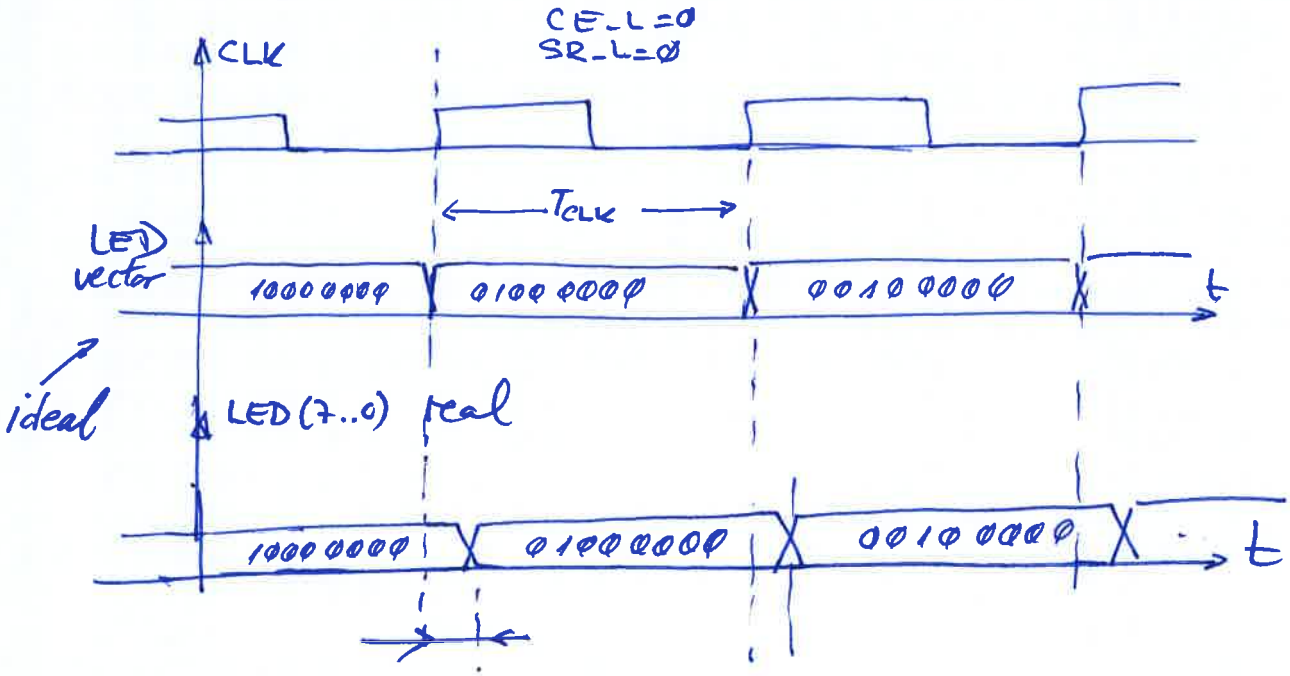
state enumeration
state type



e)



Synchronous FSM as in b)



$$t_p = t_{co} + t_{p_{CC2}}$$

t_{co}: D_FF

$t_{p_{CC2}}$: For instance, a 3-level-of-gate combinational circuit

$$t_p = T_{CLK_{min}} = t_{co} + 3\,t_{p_{1gate}}$$

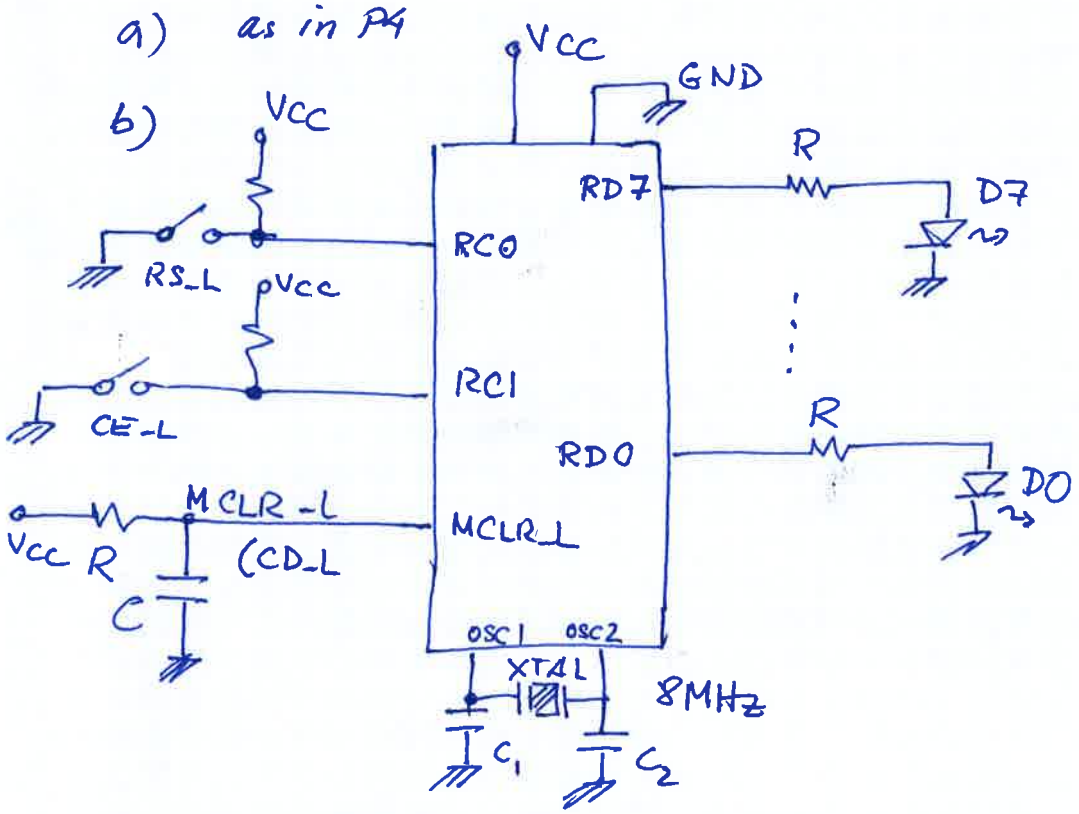
$$= 4.5\,ns + 3 \cdot 3.3\,ns = 14.4\,ns$$

$$f_{CLK_{max}} \leq 69.45\,MHz$$

P5

a) as in P4

b)



(*) CLK will be replaced by the TMR0, so, there's no need for an external input associated to an external interrupt like INT0 (RB0) (in P4)

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	1	1	TRIS C

inputs

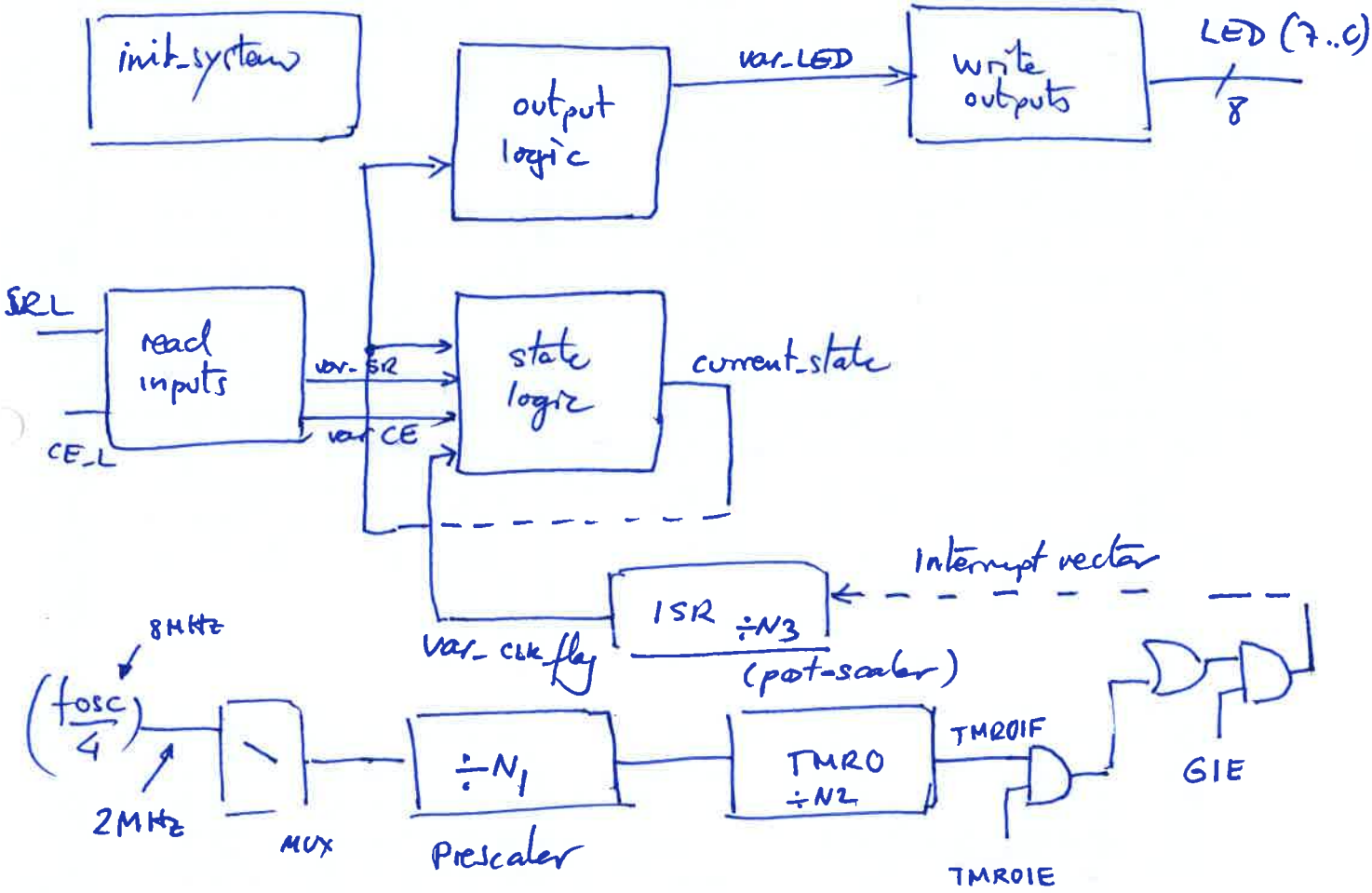
0	0	0	0	0	0	0	0	TRIS D

all outputs

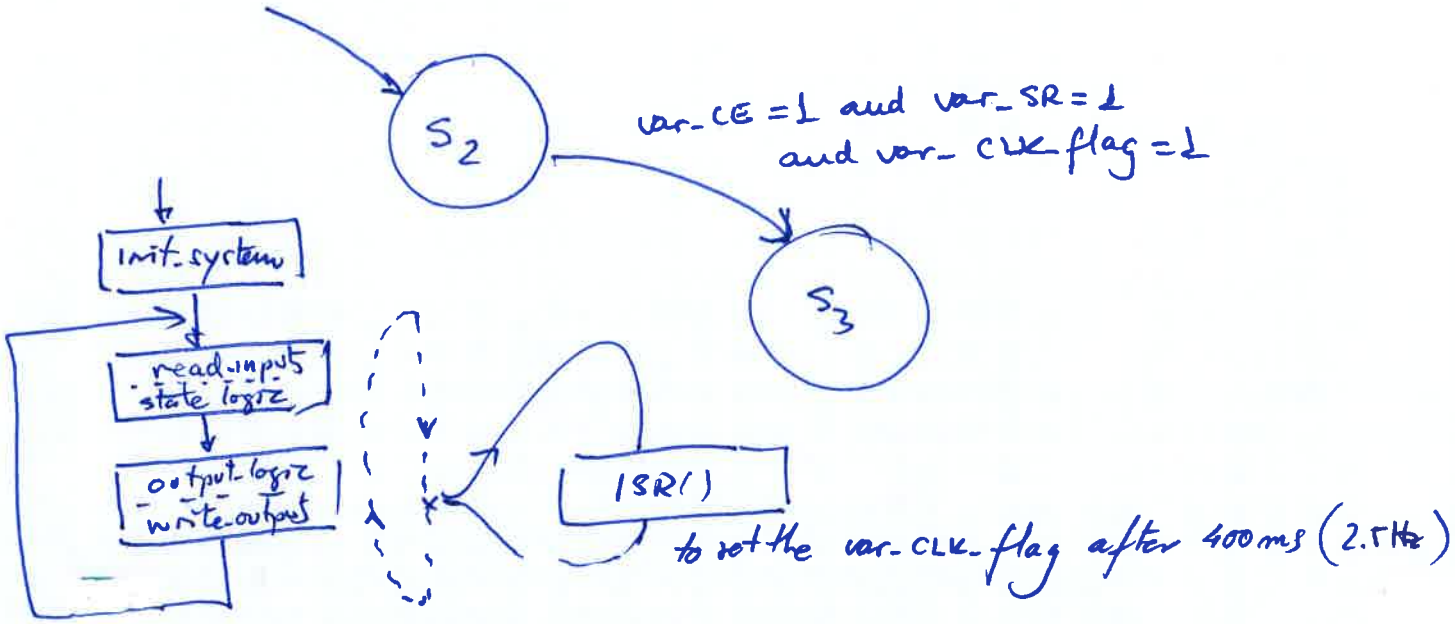
c)

d) → Similar to P4 c)

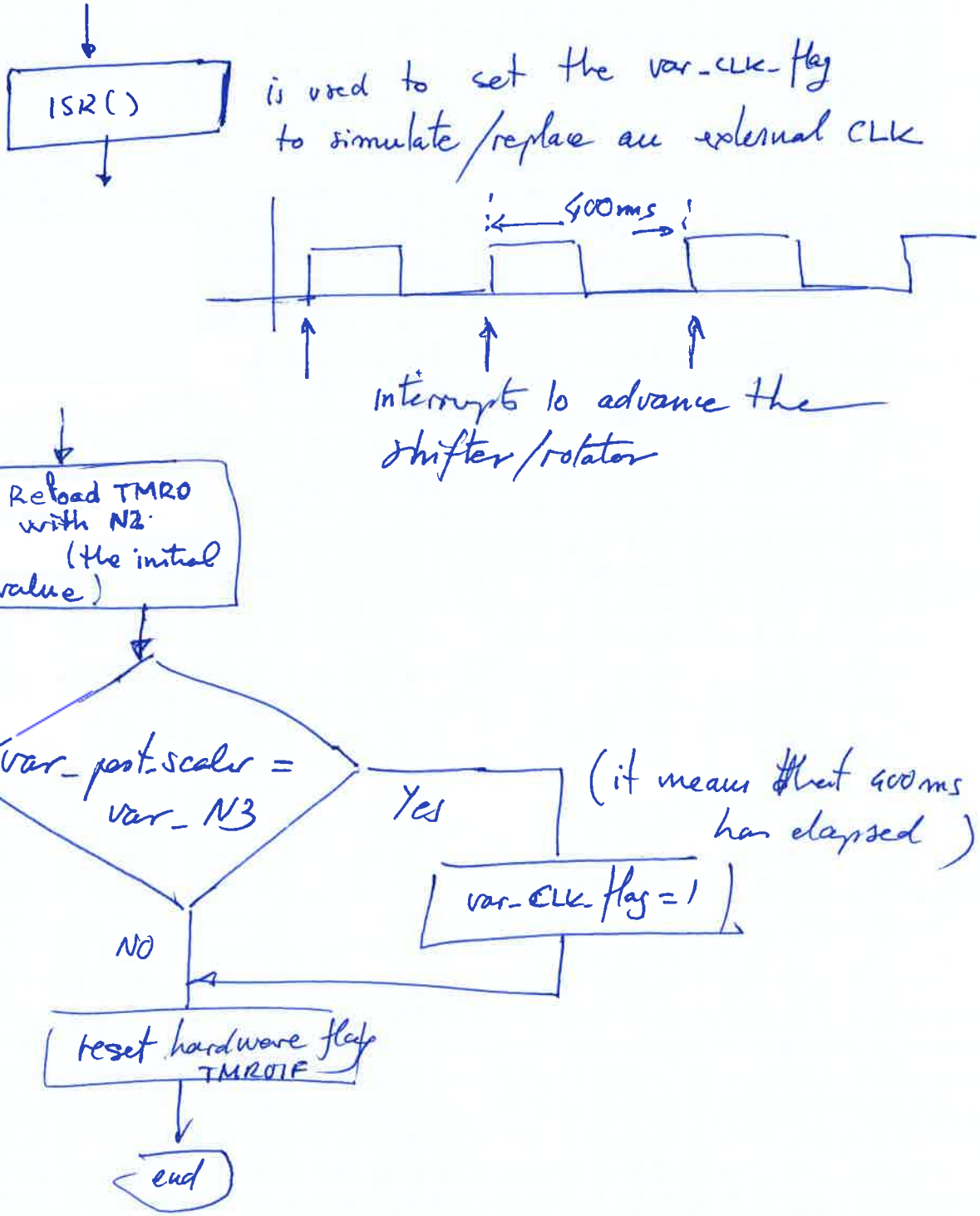
CC2 / CC1 { The truth table → flowchart → translated now to C code



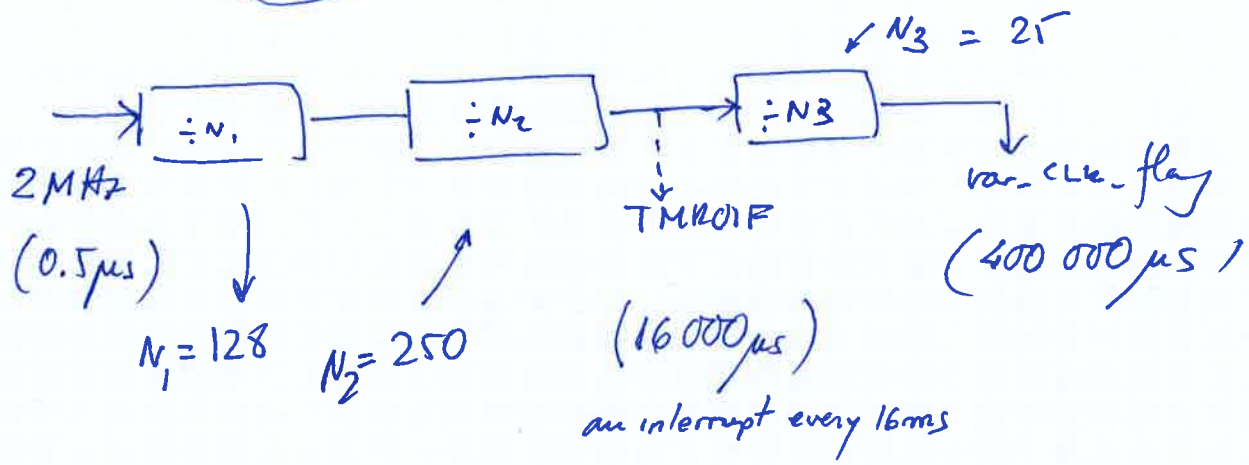
var_N1
var_N2
var_N3
} to set the timers (hardware and software)



e)


ISR()
is used to set the var_CLK_flag
to simulate/replace an external CLK
500ms
interrupt to advance the
shifter/rotator
Reload TMR0
with N2
(the initial
value)
var_postscaler =
var_N3
Yes
(it means that 400ms
has elapsed)
var_CLK_flag = 1
NO
reset hardware flag
TMR0IF
end

f)


N3 = 25
÷N1
÷N2
÷N3
2MHz
(0.5μs)
N1 = 128
N2 = 250
TMR0IF
(16 000μs)
an interrupt every 16ms
var_CLK_flag
(400 000μs)