

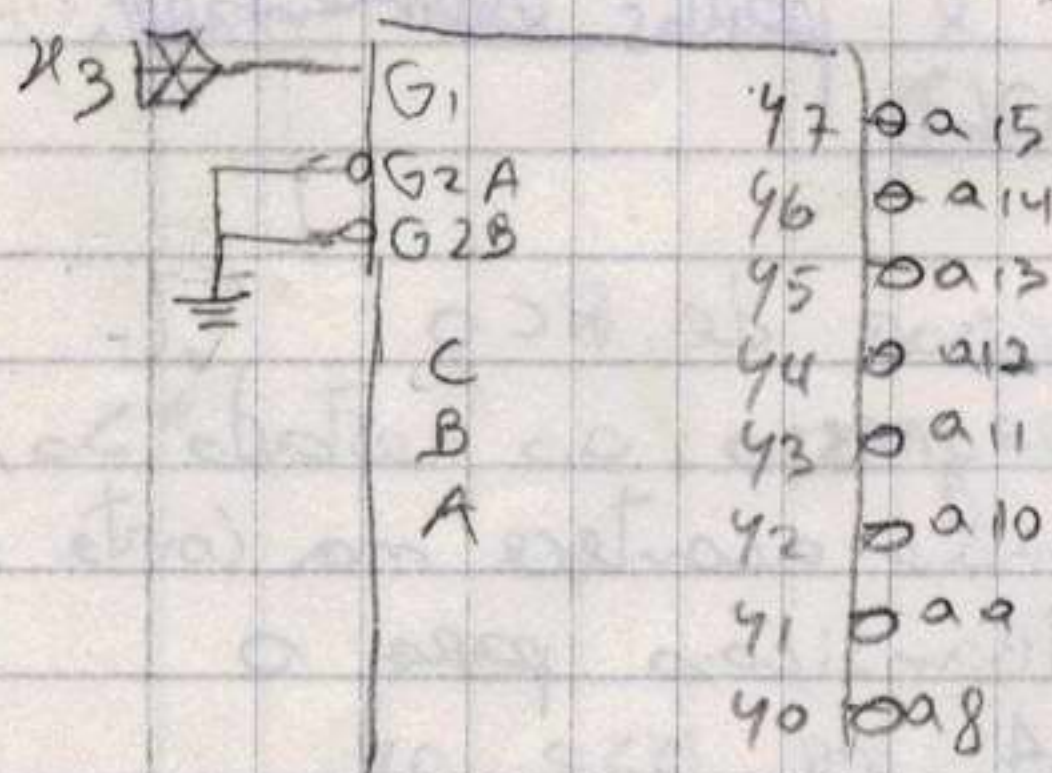
Exame Recorrência 2005

Precisamos de 3 saídas pois para uma palavra de 4 bits temos no máximo 4 bits a "0", como $(4)_{10} = (100)_2$

1 a)

i	$x_3 x_2 x_1 x_0$	$y_2 y_1 y_0$
0	0 0 0 0	1 0 0
1	0 0 0 1	0 1 1
2	0 0 1 0	0 1 1
3	0 0 1 1	0 1 0
4	0 1 0 0	0 1 1
5	0 1 0 1	0 1 0
6	0 1 1 0	0 1 0
7	0 1 1 1	0 0 1
8	1 0 0 0	0 1 1
9	1 0 0 1	0 1 0
10	1 0 1 0	0 1 0
11	1 0 1 1	0 0 1
12	1 1 0 0	0 1 0
13	1 1 0 1	0 0 1
14	1 1 1 0	0 0 1
15	1 1 1 1	0 0 0

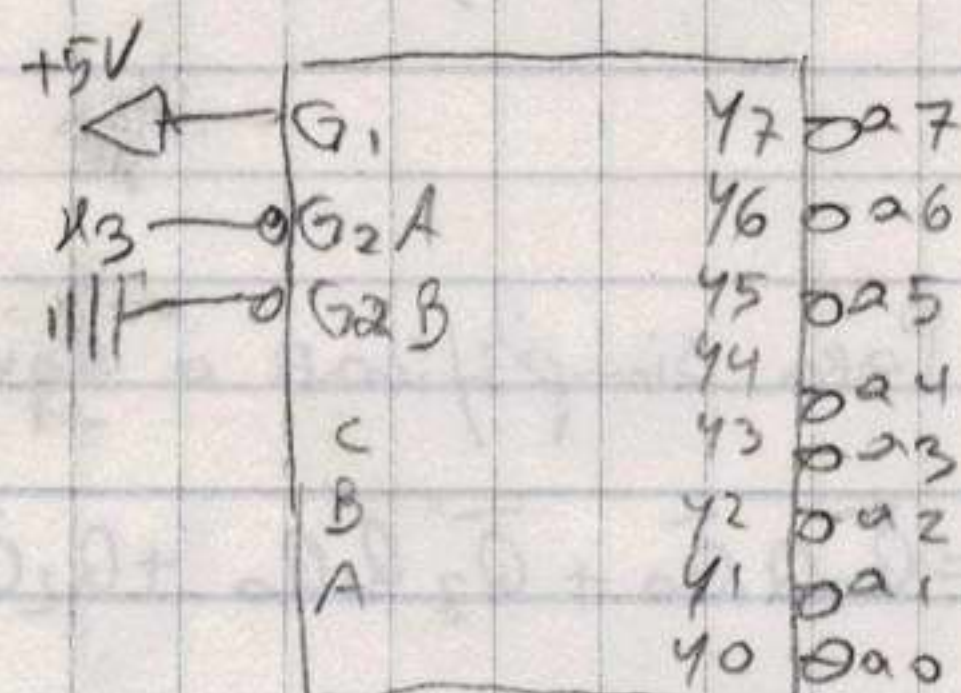
b) Precisamos de 2 decoders 3:8 pois temos 4 entradas e por isso 16 mintermos, sendo cada um dos enables o bit de entrada mais significativo activo a "1" e a "0" num e outro decodificador.



a_0 → y_2

a_1 — 0
 a_2 — 0
 a_3 — 0
 a_4 — 0
 a_5 — 0
 a_6 — 0
 a_7 — 0
 a_8 — 0
 a_9 — 0
 a_{10} — 0
 a_{12} — 0

y_1



a_1 — 0
 a_2 — 0
 a_4 — 0
 a_7 — 0
 a_8 — 0
 a_{11} — 0
 a_{13} — 0
 a_{14} — 0

y_0

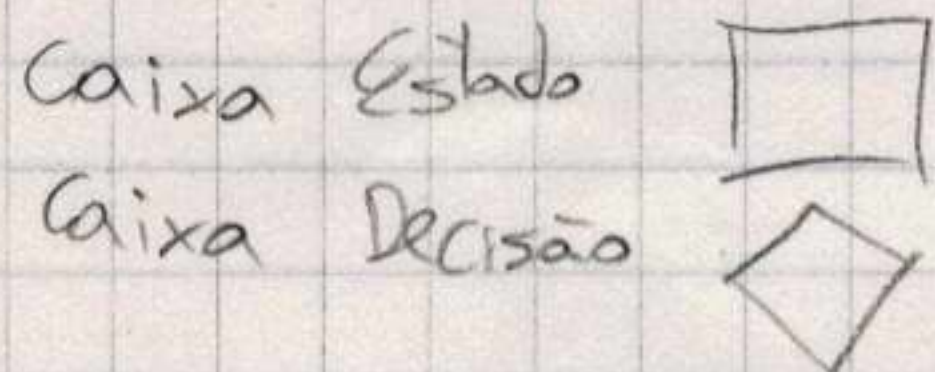
Como as saídas dos decoders estão negadas

$$\overline{A+B} = \overline{A} \cdot \overline{B} \Rightarrow \sum m_i = \sum \overline{m_i}$$

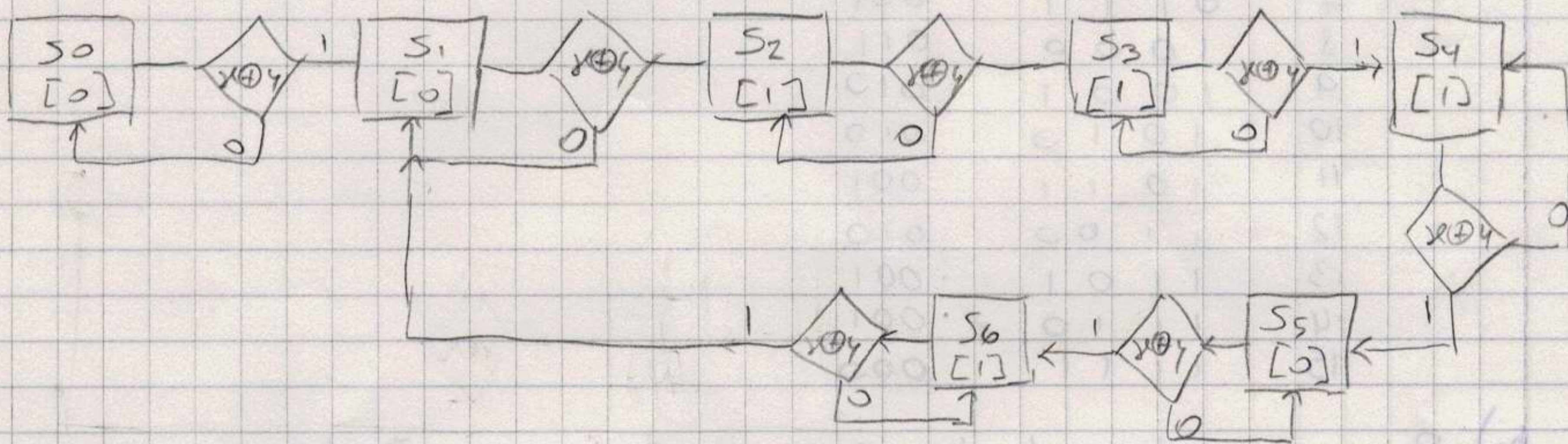
c) Impossível

d) Carta ASM

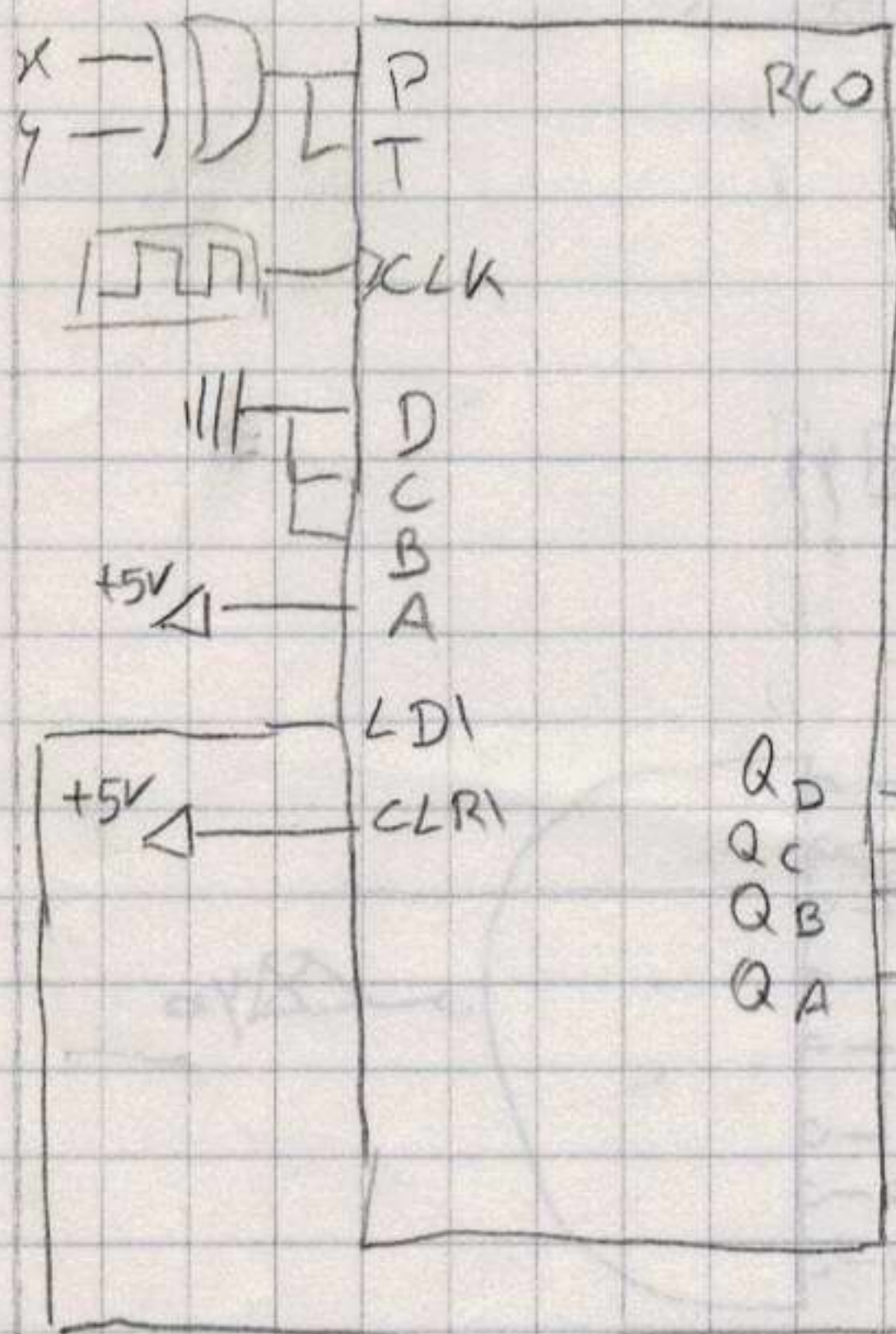
Modelo de Moore \Rightarrow saídas associadas às entradas



$z=1$ qd distância Hamming entre x e y múltipla 2 ou 3.



e) Implementar o SSS com contador binário e portas elementares.
Se P ou T forem a 0 o contador vai a HOLD



- ✓ Neste caso não vamos precisar de RCO
- ✓ O sinal CLR força o regresso ao estado S_0 , não vamos precisar pois tal não acontece na Carta ASM
- ✓ O sinal LDI força uma transição para o estado indicado por DCBA por isso qd chegarmos ao estado S_6 temos de forçar o estado S_1 .

Para simplificar a equação de z

$$z = \bar{Q}_2 Q_1 \bar{Q}_0 + \bar{Q}_2 Q_1 Q_0 + Q_2 \bar{Q}_1 \bar{Q}_0 + Q_2 Q_1 \bar{Q}_0$$

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0			1	1
1	1			

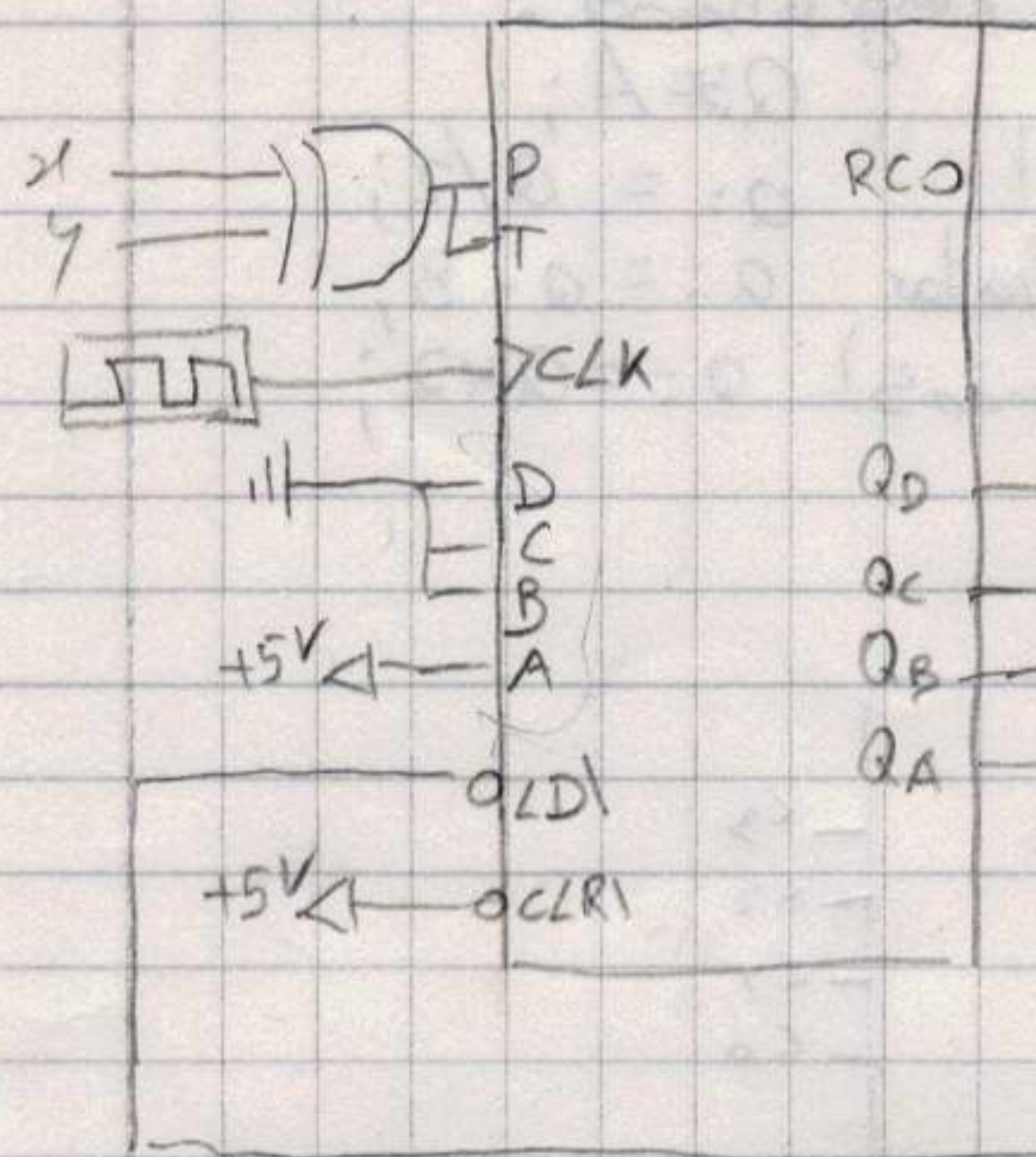
$$z = \bar{Q}_2 Q_1 + Q_2 \bar{Q}_0$$

i	$Q_2 Q_1 Q_0$	z
0	0 0 0	
1	0 0 1	
2	0 1 0	1
3	0 1 1	1
4	1 0 0	1
5	1 0 1	
6	1 1 0	1

O estado S_6 é o primeiro com $Q_2=1, Q_1=1$ logo podemos usar isso para a detecção do estado S_6

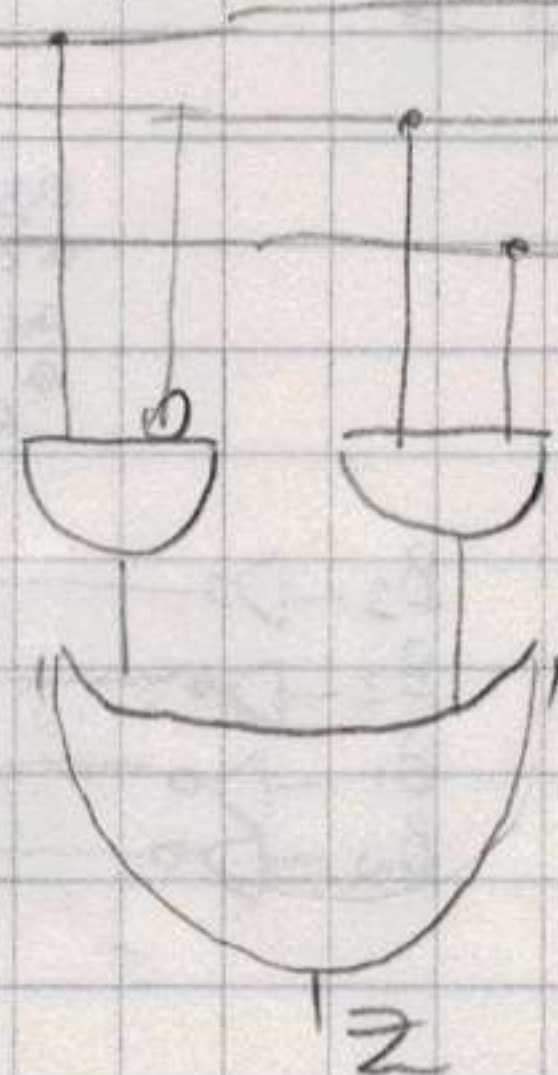
RECORRÊNCIA SD 2005

1e)



Neste 555 o sinal clear não é utilizado pois não precisamos de voltar ao estado S_0 .

O sinal load é activo ao chegarmos ao último estado



	$Q_2 Q_1 Q_0$	Z
S_0	0 0 0	
S_1	0 0 1	
S_2	0 1 0	1
S_3	0 1 1	1
S_4	1 0 0	1
S_5	1 0 1	1
S_6	1 1 0	1

$$Z = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + \bar{Q}_2 \bar{Q}_1 Q_0 + \bar{Q}_2 Q_1 \bar{Q}_0 + \bar{Q}_2 Q_1 Q_0$$

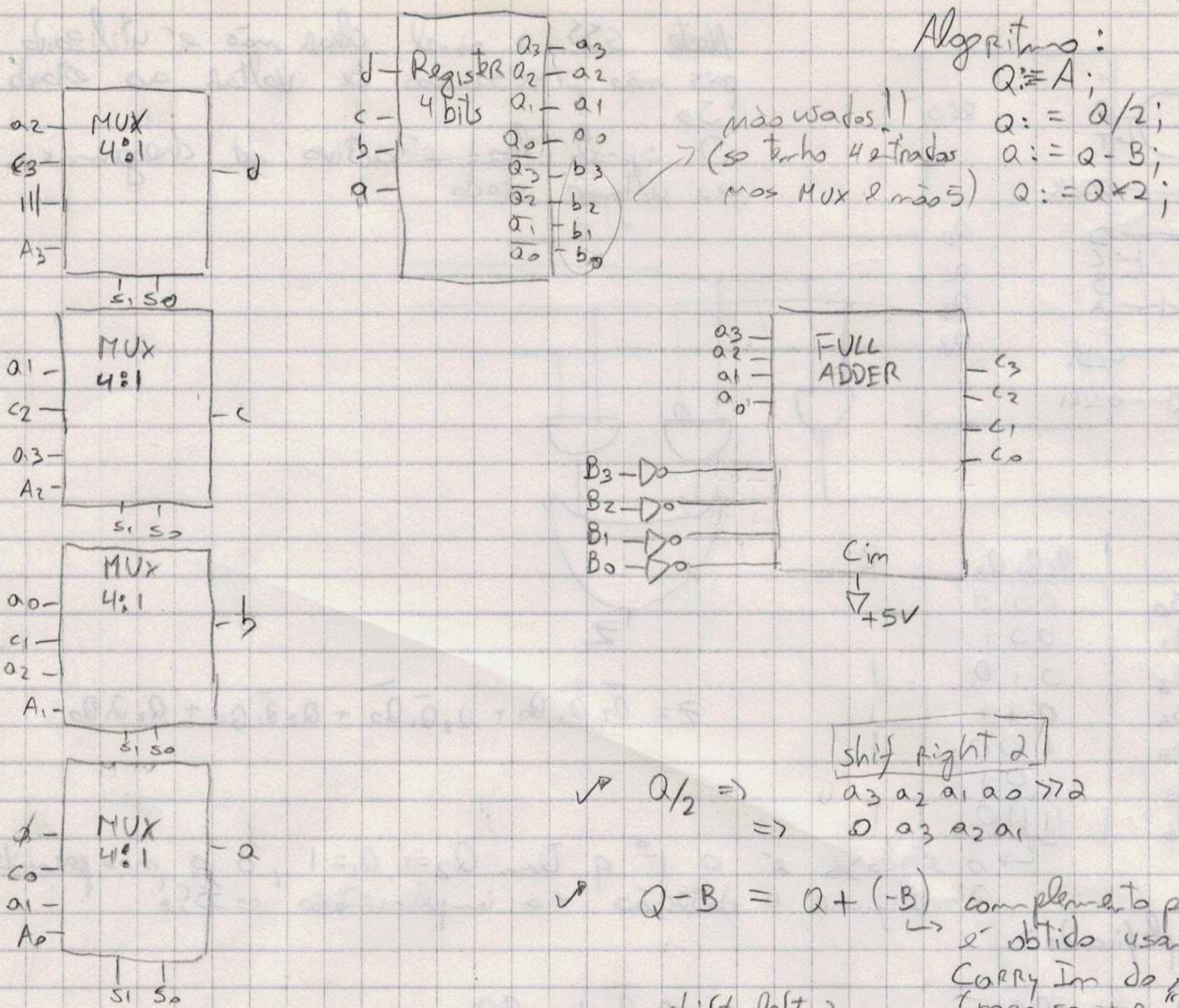
↳ o estado S_6 é o 1º q tem $Q_2=1, Q_1=1, Q_0=0$ mas permite simplificar a detecção e a implementação do 555!

Simplificando Z

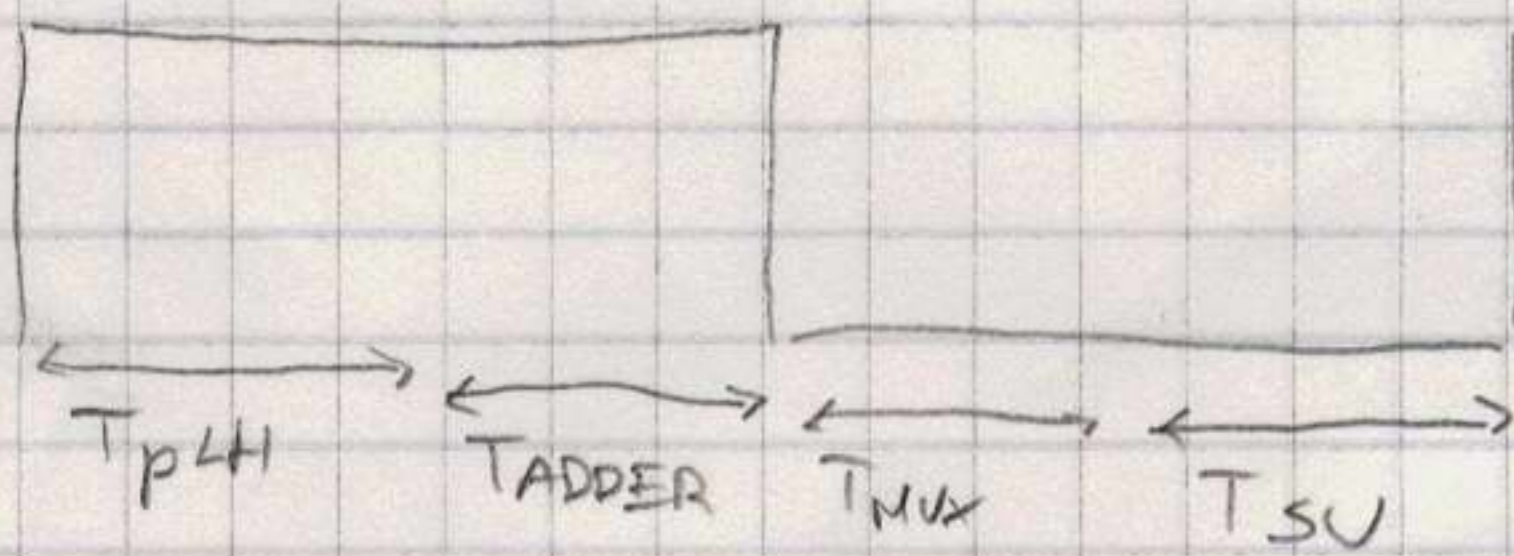
Q_2	$Q_1 Q_0$	
0	00 01 11 10	0 0 1 0
1	00 01 11 10	0 0 1 1

$$Z = Q_2 \bar{Q}_1 + Q_1 Q_0$$

2) operandos de 4 bits com representação em complemento para 2



b) Considerando o período do sinal CLK e snmp o pior dos casos, temos um atraso de propagação à saída do register ($T_{PLH} > T_{PHL}$), dps as saídas do register vão ser as entradas do MUX e do ADDER o que implica que consideremos o pior caso ($T_{ADDER} > T_{MUX}$), as saídas do ADDER vão ser as entradas dos MUX (os MUX estão em paralelo logo consideramos apenas T_{MUX}). Finalmente temos de garantir um tempo mínimo T_{su} para configuração dos FF.



$$T_{min} = T_{PLH} + T_{ADDER} + T_{MUX} + T_{su}$$

$$= 18 + 10 + 4 + 8$$

$$= 40 \text{ ns}$$

$$\Rightarrow f_{max} = \frac{1}{T_{min}} = \frac{1}{40 \times 10^{-9}} =$$