

SD

2005 junho

1)

a)

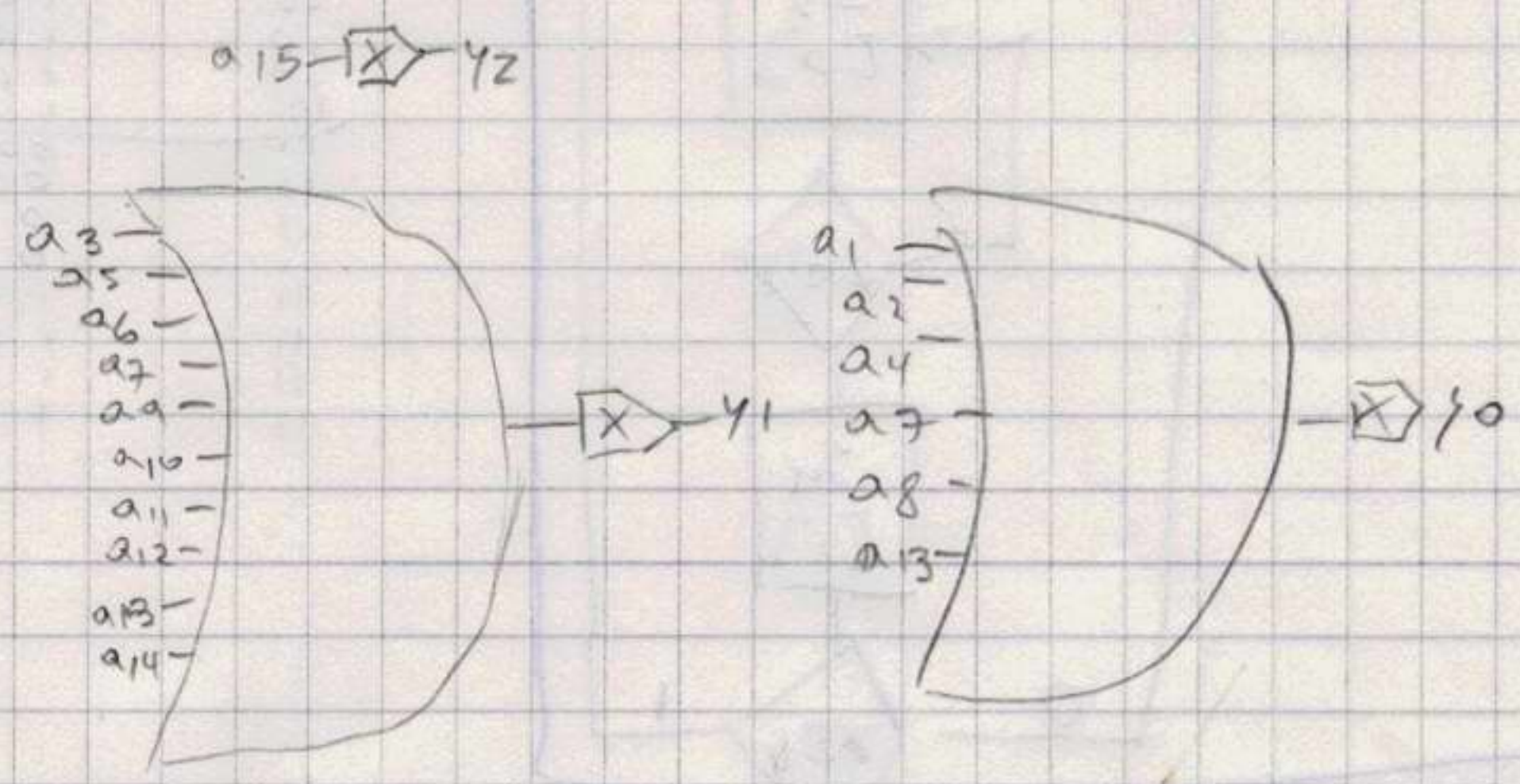
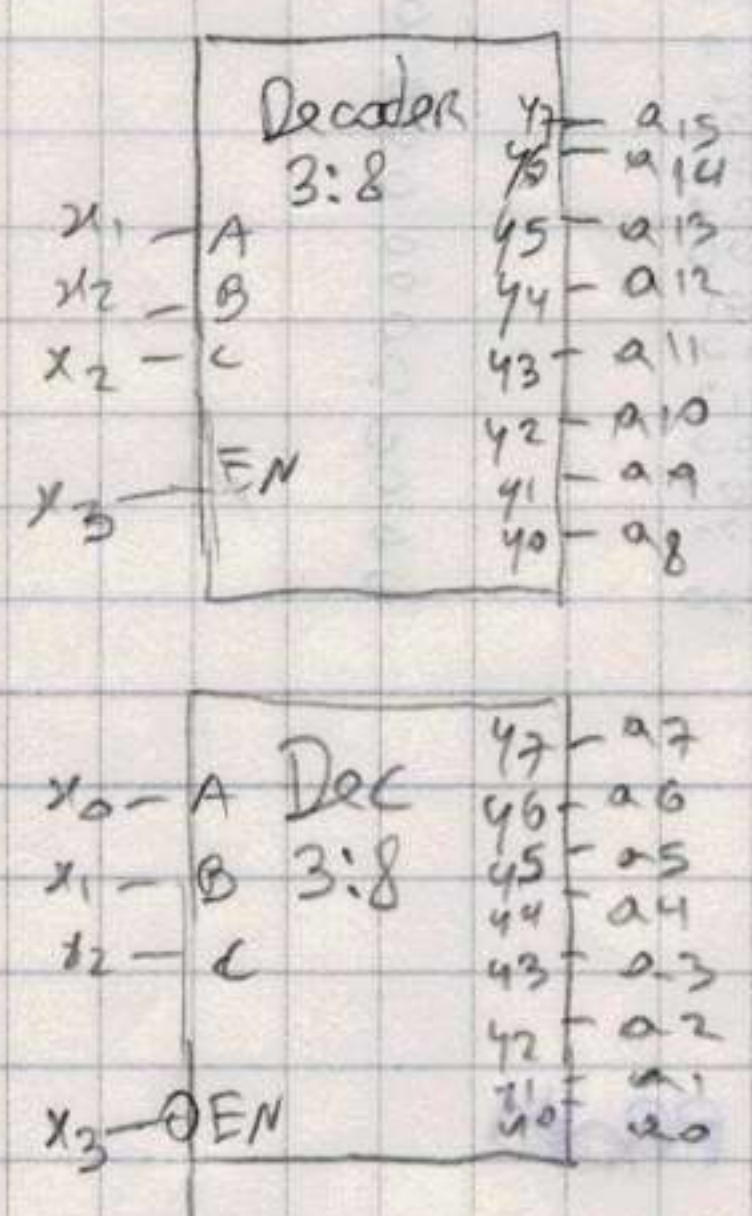
i	x_3	x_2	x_1	x_0	$y_2 y_1 y_0$
0	0	0	0	0	0 0 0
1	0	0	0	1	0 0 1
2	0	0	1	0	0 0 1
3	0	0	1	1	0 1 0
4	0	1	0	0	0 0 1
5	0	1	0	1	0 1 0
6	0	1	1	0	0 1 0
7	0	1	1	1	0 1 1
8	1	0	0	0	0 0 1
9	1	0	0	1	0 1 0
10	1	0	1	0	0 1 0
11	1	0	1	1	0 1 1
12	1	1	0	0	0 1 0
13	1	1	0	1	0 1 1
14	1	1	1	0	0 1 1
15	1	1	1	1	1 0 0

precisamos de 3 bits para o n° máximo de bits a "i" numa palavra de 4 bits
 $2^4 \Rightarrow (4)_{10} = (100)_2$

b)

Decodificadores 3:8

Como temos 4 variáveis de entrada \Rightarrow 16 termos precisamos de 2 decodificadores 3:8, sendo cada um dos enable o bit de entrada mais significativo activo a 1 e a 0 num circuito decodificador.

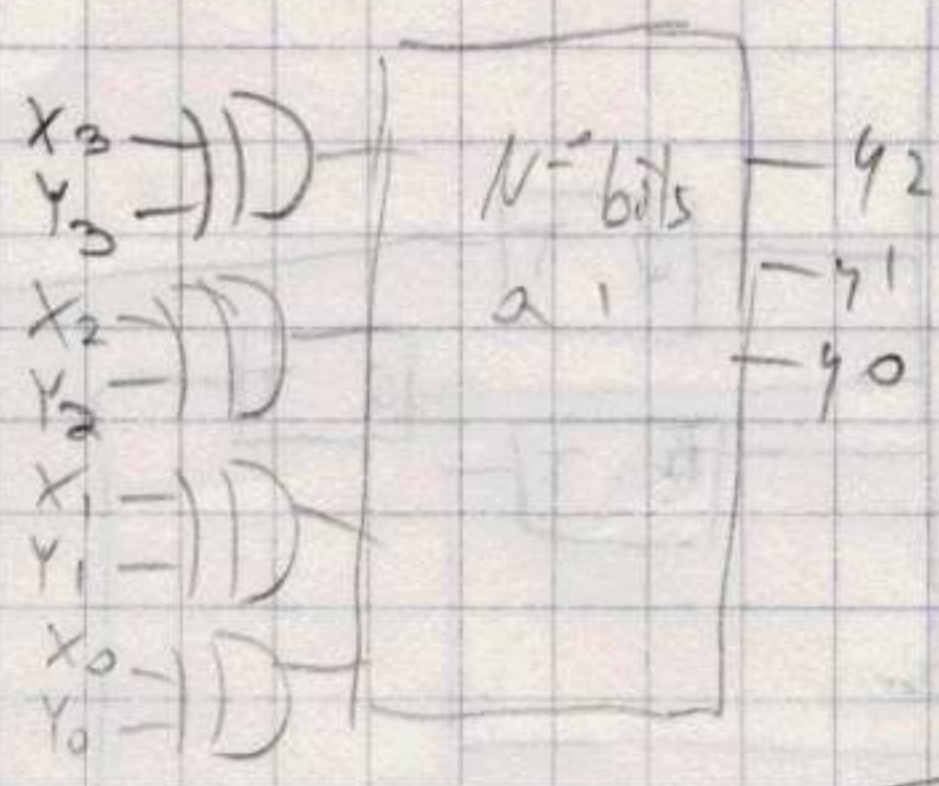
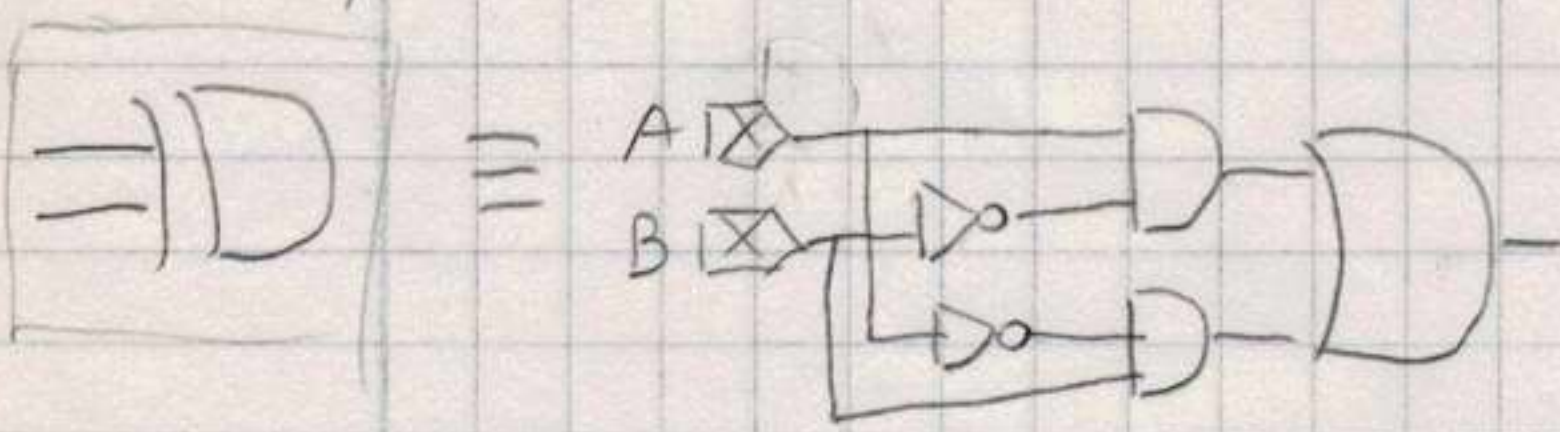


c) Detectar distância Hamming entre 2 palavras de 4 bits

Ex: $X: 0110$
 $Y: 1010$
 $d: 1100$

$\Rightarrow d_i = X_i \oplus Y_i$

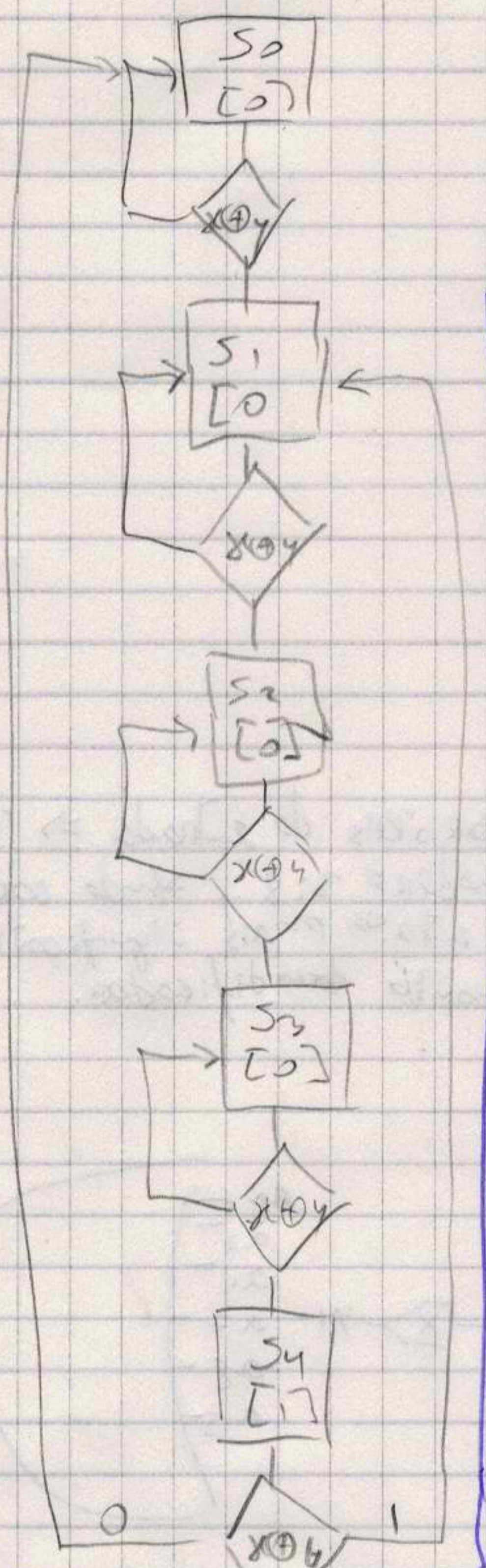
Sendo,





detector distância Hamming

e) Implementação SSS em notação ASM modelo Moore

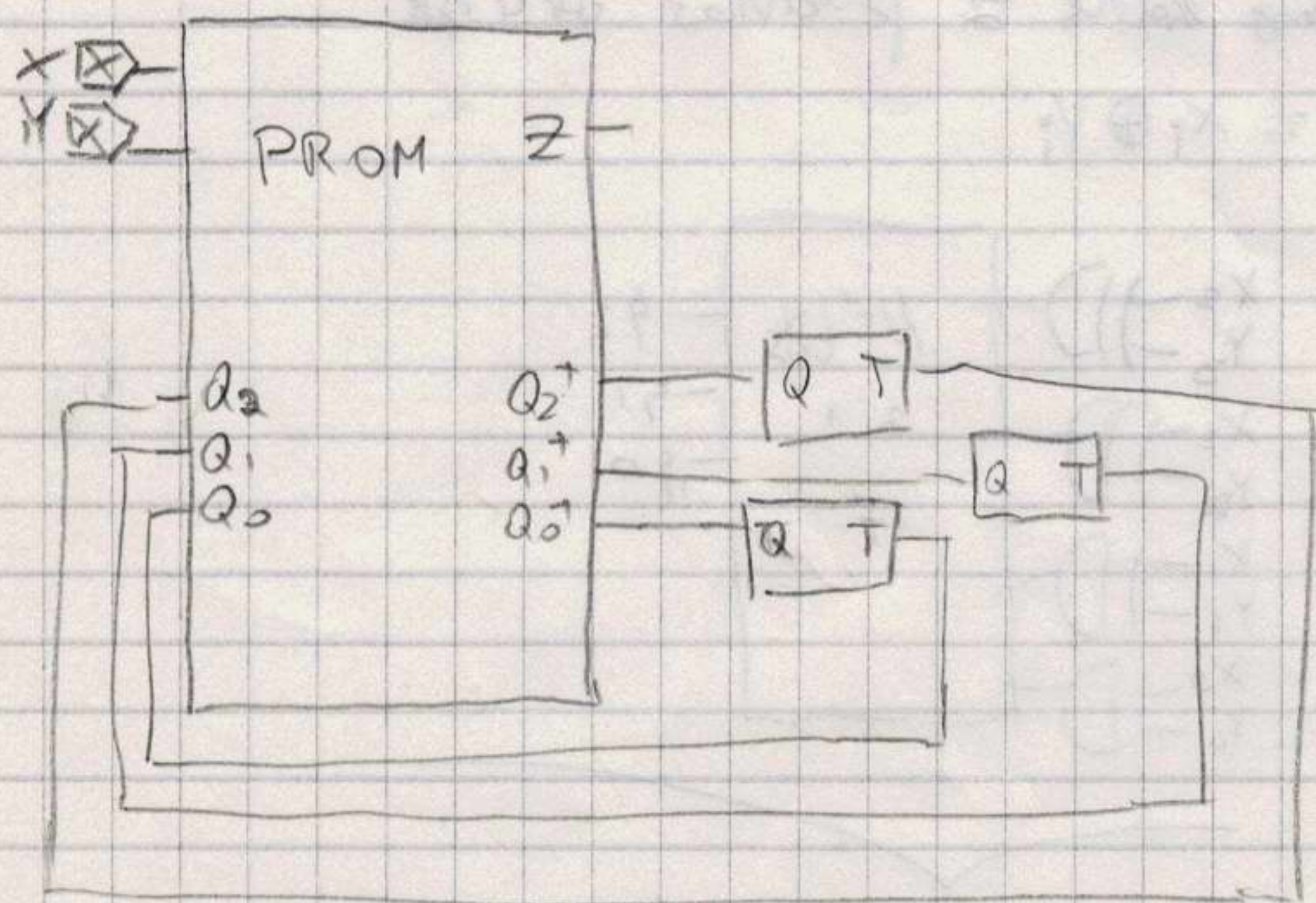
Notação Moore: Saídas associadas as entradas



Caixa estado: 
Caixa decisão: 

XY	$Q_2 Q_1 Q_0$	$Q_2^+ Q_1^+ Q_0^+$	Z
00	000	000	0
00	001	001	0
00	010	010	0
00	011	011	0
00	100	100	0
00	101	101	0
00	110	110	0
00	111	111	0
01	000	001	0
01	001	010	0
01	010	011	0
01	011	100	0
01	100	101	0
01	101	110	0
01	110	111	0
01	111	000	0
10	000	000	0
10	001	001	0
10	010	010	0
10	011	011	0
10	100	100	0
10	101	101	0
10	110	110	0
10	111	111	0
11	000	001	0
11	001	010	0
11	010	011	0
11	011	100	0
11	100	101	0
11	101	110	0
11	110	111	0
11	111	000	0

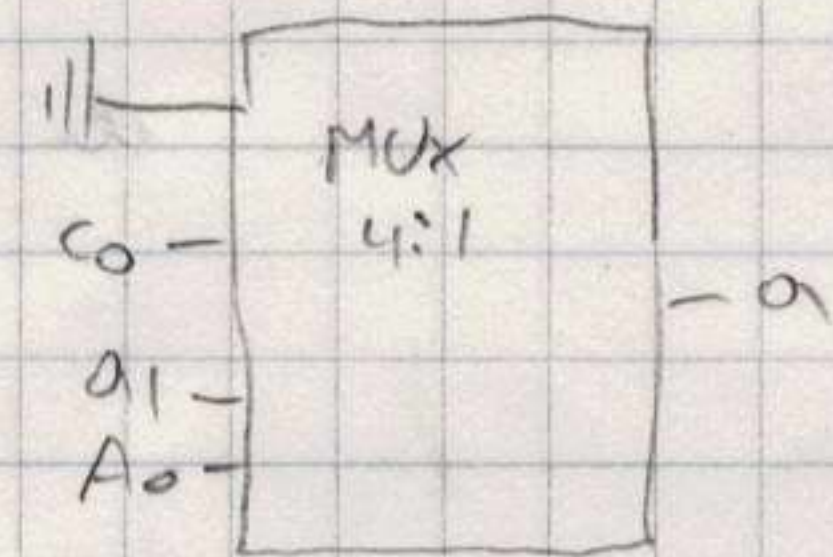
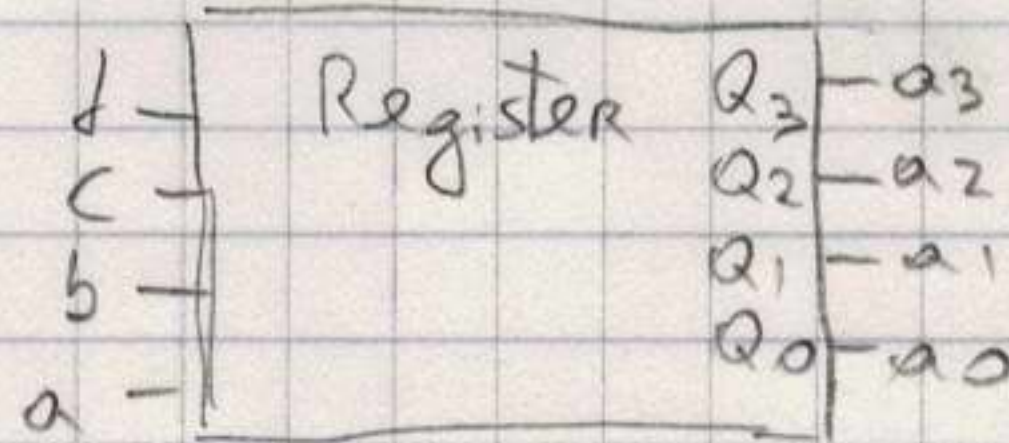
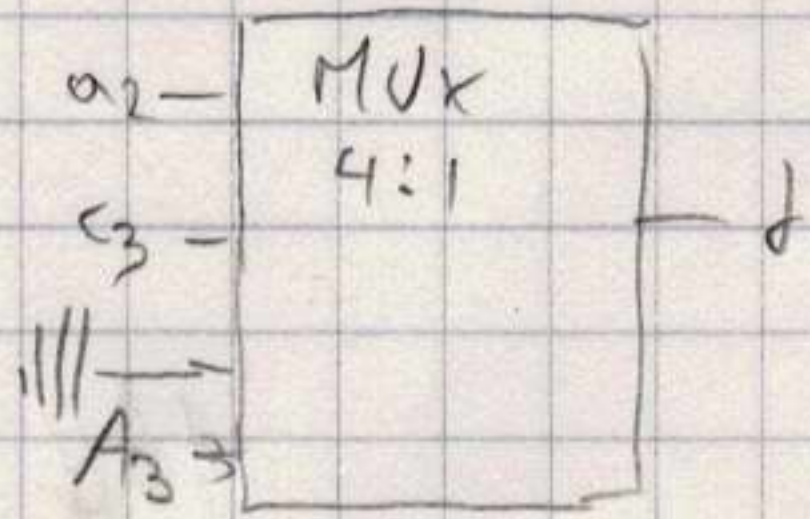
e) Implementação do SSS com base em FF-T e PROM



2

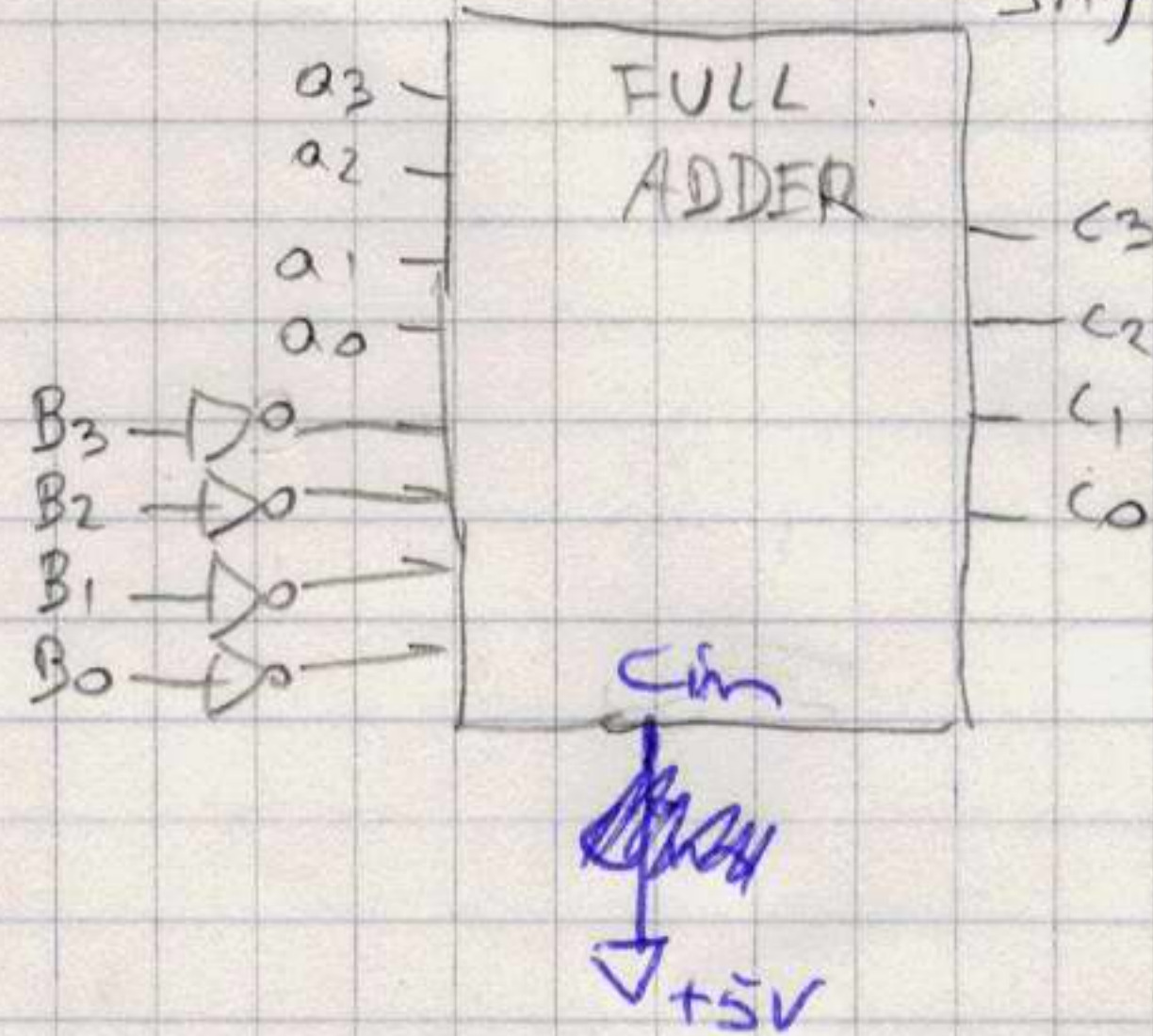
Algoritmo:

- a)
- Carregar $Q := A$
 - Dividir $Q/2$
 - Subtrair $Q - B \equiv Q + (-B)$
- complemento de 2 é obtido fazendo o complemento bit a bit de B e somando 1 (pode-se usar para isso o carry in do somador curto-circuitado a +5V)



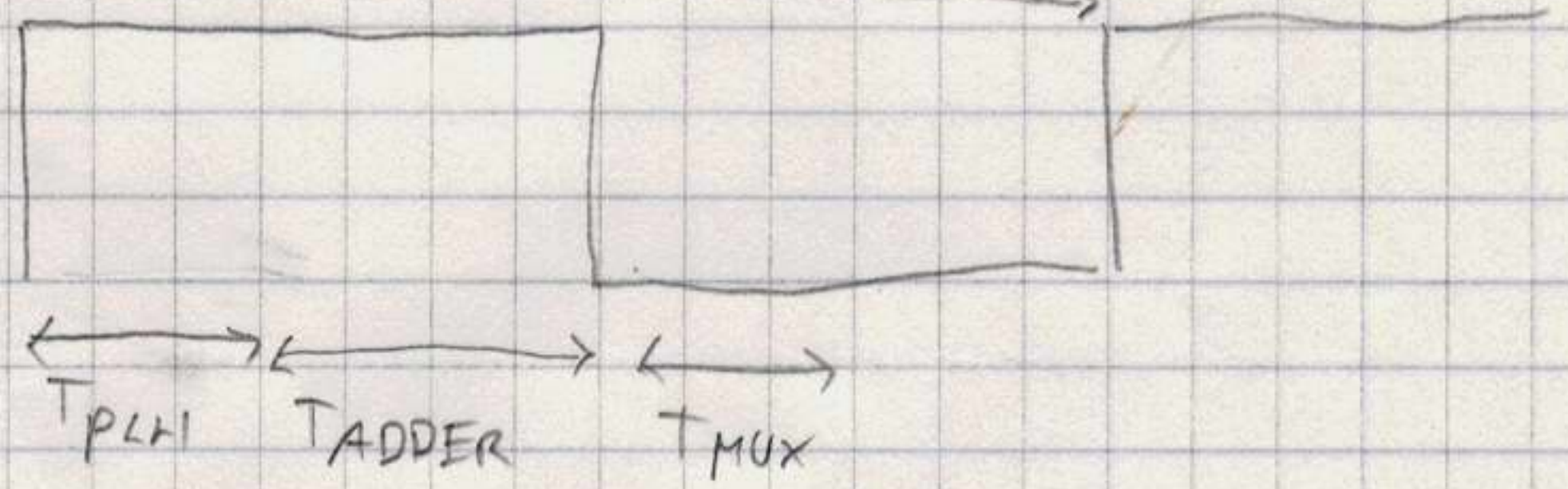
Para dividir por 2
shift right 2
 $\Rightarrow (a_3 a_2 a_1 a_0) \gg 2$
 $0 a_3 a_2 a_1$

Para multiplicar por 2
shift left 2
 $(a_3 a_2 a_1 a_0) \ll 2$
 $a_2 a_1 a_0 \phi$



b) considerando o período do sinal clk T_{clk}

Por cada período do relógio temos um atraso de propagação na saída do Register, como se considera sempre o pior caso e $T_{PLH} > T_{PHL}$ usamos T_{PLH} , em seguida as saídas do Register são as entradas dos MUX e do ADDER (como $T_{ADDER} > T_{MUX}$, considera-se apenas o atraso T_{ADDER}).



As saídas do ADDER vão ser entradas dos MUX, temos então um atraso T_{MUX} (pois os MUX estão em paralelo) e finalmente as saídas dos MUX são entradas do register, sendo necessário garantir um tempo igual ou maior do que T_{setup} para a configuração dos FF e inicialização de novo período do relógio.