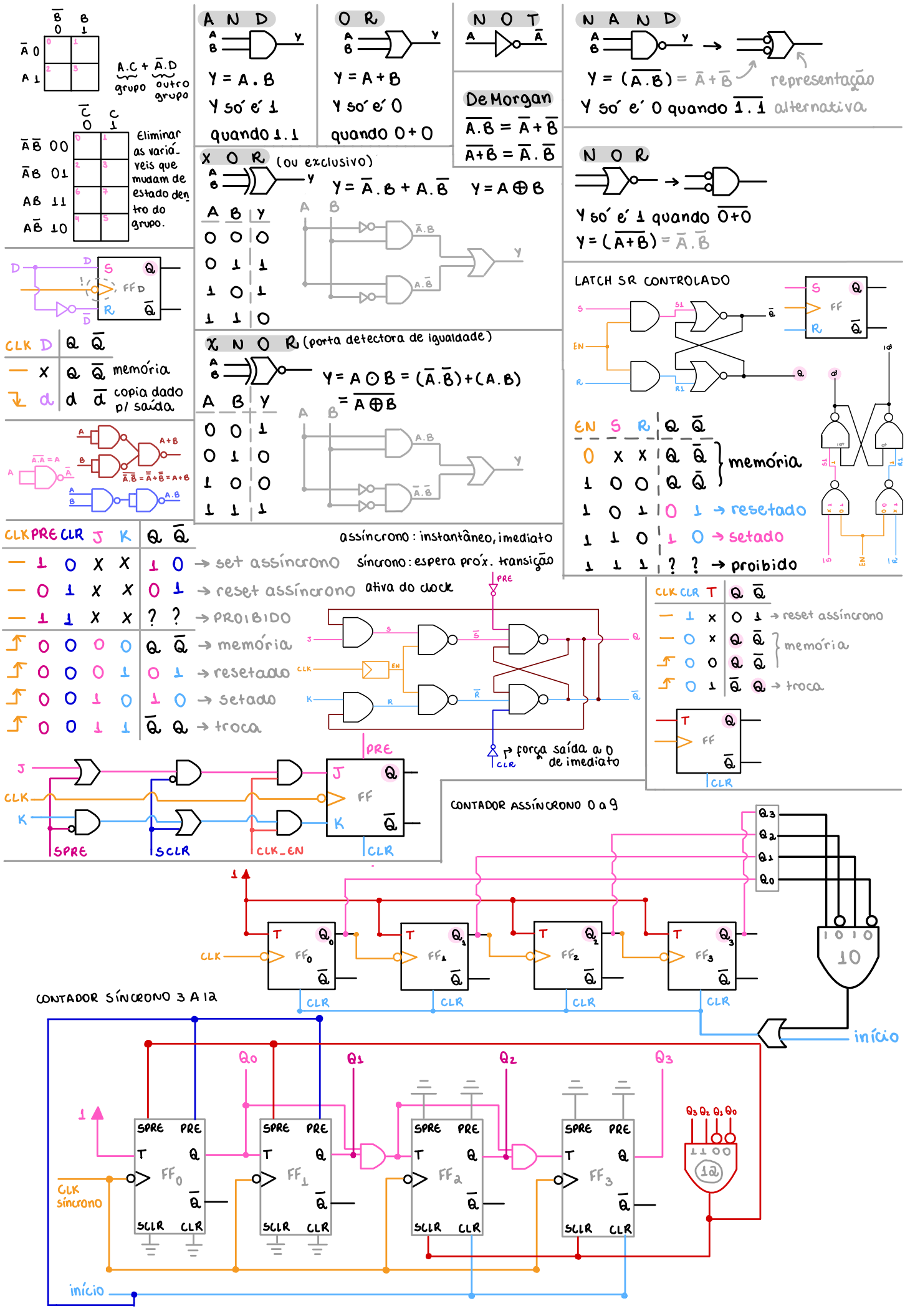
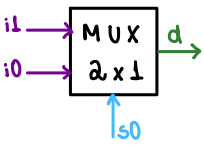
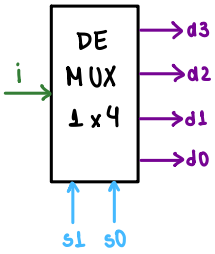

AND
Y = A . B
Y só é 1 quando 1.1
OR
Y = A + B
Y só é 0 quando 0 + 0
NOT
De Morgan
A.B = Ā + B̄
A+B = Ā . B̄
NAND
Y = (A.B) = Ā + B̄
representação alternativa
Y só é 0 quando 1.1
NOR
Y só é 1 quando 0+0
Y = (A+B) = Ā . B̄
XOR (ou exclusivo)
Y = Ā.B + A.B̄
Y = A ⊕ B
XNOR (porta detectora de igualdade)
Y = A ⊙ B = (Ā.B̄)+(A.B)
= Ā ⊕ B̄
Eliminar as variáveis que mudam de estado dentro do grupo.
A.C + Ā.D
grupo
outro grupo
FF D
CLK D | Q Q̄
— X | Q Q̄ memória
↓ d | d d̄ copia dado p/ saída
LATCH SR CONTROLADO
EN S R | Q Q̄
0 X X | Q Q̄ } memória
1 0 0 | Q Q̄
1 0 1 | 0 1 → resetado
1 1 0 | 1 0 → setado
1 1 1 | ? ? → proibido
CLK PRE CLR J K | Q Q̄
— 1 0 X X | 1 0 → set assíncrono
— 0 1 X X | 0 1 → reset assíncrono
— 1 1 X X | ? ? → PROIBIDO
↑ 0 0 0 0 | Q Q̄ → memória
↑ 0 0 0 1 | 0 1 → resetado
↑ 0 0 1 0 | 1 0 → setado
↑ 0 0 1 1 | Q̄ Q → troca
assíncrono: instantâneo, imediato
síncrono: espera próx. transição ativa do clock
força saída a 0 de imediato
CLK CLR T | Q Q̄
— 1 X | 0 1 → reset assíncrono
— 0 X | Q Q̄ } memória
↑ 0 0 | Q Q̄
↑ 0 1 | Q̄ Q → troca
CONTADOR ASSÍNCRONO 0 a 9
CONTADOR SÍNCRONO 3 A 12
início
CLK síncrono

MUX → N entradas de dados e 1 saída

DEMUX → 1 entrada de dados e N saídas

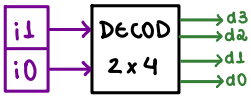


Nos seletores colocamos em binário a posição da entrada desejada.
Ex.: s1 = 1 e s2 = 0 seleciona a entrada i2 para passar.

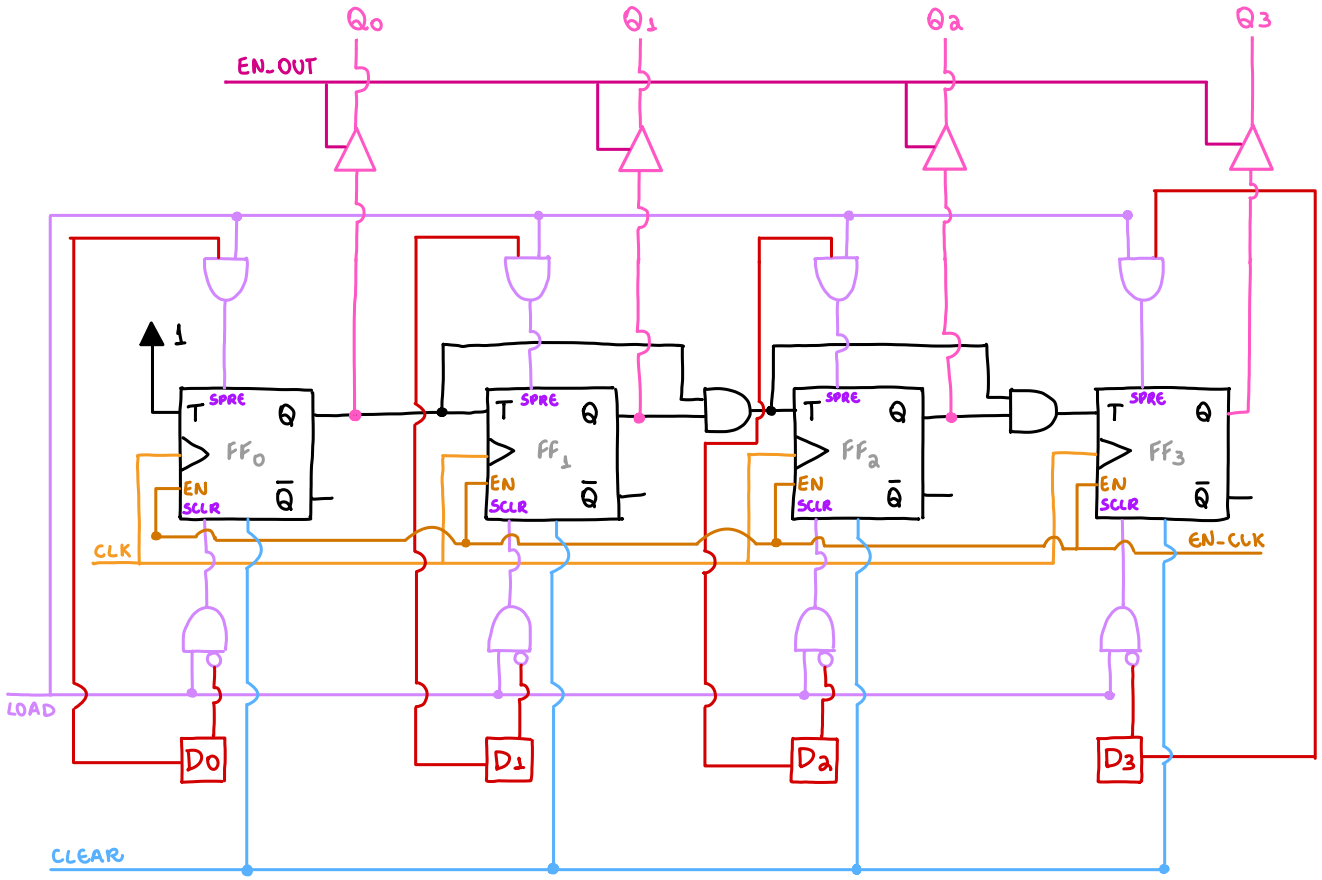


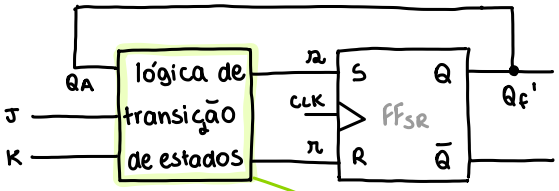
Passa a entrada para a saída determinada pelos seletores. As outras saídas permanecem em 0.

DECODIFICADOR



Descobrimos qual foi a entrada a partir da saída ativada.





simboliza o que queremos que ocorra quando o clock for dado no estado atual, qual o próximo estado.

	Q_A	J	K	$\overline{s}$	$\overline{r}$	Q_f'	
0	0	0	0	0	~~0~~	0	→ memória
1	0	0	1	0	~~1~~	0	→ reset
2	0	1	0	1	0	1	→ set
3	0	1	1	1	0	1	→ troca
4	1	0	0	~~0~~	0	1	→ memória
5	1	0	1	0	1	0	→ reset
6	1	1	0	~~1~~	0	1	→ set
7	1	1	1	0	1	0	→ troca

X = don't care

Quando $Q_A = 1$, daí na mesma fazer memória → ou setar, o importante mesmo é $\overline{r} = 0$.

M.K. para $\overline{s}$:

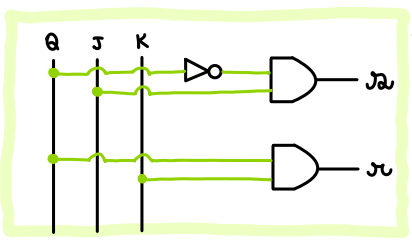
Q_A \ JK	$\overline{K}\overline{J}$	$\overline{K}J$	KJ	$K\overline{J}$
$\overline{Q}_A$	0	0	X	X
Q_A	1	1	0	0

$\overline{s} = \overline{Q}_A \cdot J$

M.K. para $\overline{r}$:

Q_A \ JK	$\overline{K}\overline{J}$	$\overline{K}J$	KJ	$K\overline{J}$
$\overline{Q}_A$	X	X	0	0
Q_A	0	0	1	1

$\overline{r} = Q_A \cdot K$



$\overline{s}$ e $\overline{r}$ são as saídas da nossa lógica de transição de estados.

Queremos que o uso proibido do SR (as duas entradas ativas) se torne TROCA.

- Quando temos esse estado atual, o que queremos que saia no próx. estado? (futuro)
- O que preciso ter em $\overline{s}$ e em $\overline{r}$ para que Q_f seja o que eu quero?