

# Prova P4 EAI-21-2020 -COMP

Nome: RODRIGO ALVES DE ALMEIDA Duração: 3:00h

Dia: 31/07/20 Horário - Início 00:45 Término 3:40

**1Q (2.0) :** Seja uma macro-célula de um dispositivo programável composta de um FF JK e lógica adicional (ver Figura 1). Através das entradas M e N esta macro-célula é reconfigurável por programação. Projete esta macro-célula com o menor número de portas NAND e Inversoras, onde ela realiza as seguintes operações: se  $M=N=0$  a macro-célula é um simples FF JK; se  $M=N=1$  a macro-célula é um FF JK invertido, isto é a saída da macro-célula é invertida; se  $M=0$  e  $N=1$ , a macro-célula é um simples FF D; se  $M=1$  e  $N=0$ , a macro-célula é um simples FF T.

Obs: as equações características dos FFs são:  $Q_{N+1} = JQ_N' + K'Q_N$ ,  $Q_{N+1} = D$  e  $Q_{N+1} = T \oplus Q_N$ .

Tabela de Reconfiguração

M	N	W	Z	Q
0	0	J	K	Q
0	1	D	—	Q
1	1	J	K	$\overline{Q}$
1	0	T	—	Q

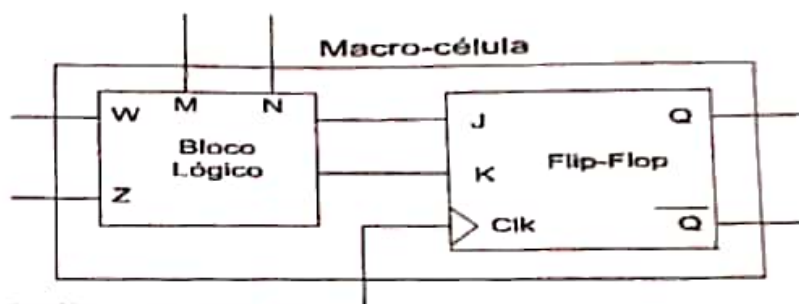


Figura 1. Macro-célula de um dispositivo programável.

•  $MN=00$

$J=W$        $K=Z$

•  $MN=11$

$$Q_{N+1} = \overline{W} \overline{Q_N} + \overline{Z} Q_N = (\overline{W} + Q_N)(\overline{Z} + \overline{Q_N}) = \overline{W} \overline{Q_N} + \overline{Z} Q_N + \overline{W} \overline{Z}$$

$J=\overline{W}$        $K=\overline{Z}$

•  $MN=01$

$$Q_{N+1} = W = W \cdot \overline{Q_N} + W \cdot Q_N$$

↳ faz papel de D

$J=W$        $K=\overline{W}$

•  $MN=10$

$$Q_{N+1} = W \oplus Q_N = W \cdot \overline{Q_N} + \overline{W} \cdot Q_N$$

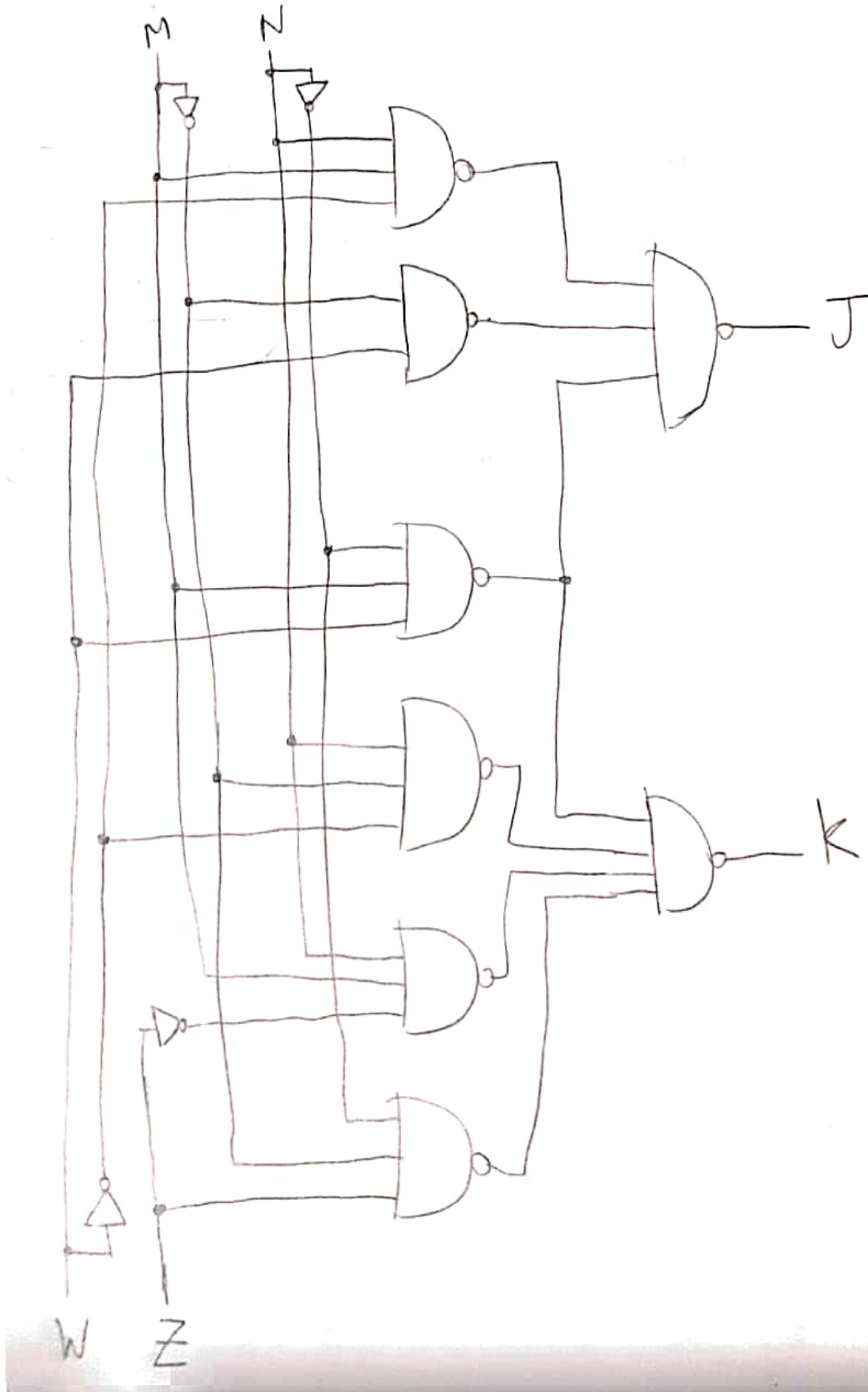
$J=W$        $K=W$

$$J = \overline{M} \overline{N} W + \overline{M} N W + M \overline{N} W + M N W =$$

$$= \overline{M} W + M N \overline{W} + M N W = \overline{M} W \cdot \overline{M N \overline{W}} \cdot \overline{M N W}$$

$$K = \overline{M} \overline{N} Z + M N \overline{Z} + \overline{M} N \overline{W} + M \overline{N} W =$$

$$= \overline{M} \overline{N} Z \cdot \overline{M N \overline{Z}} \cdot \overline{M N \overline{W}} \cdot \overline{M \overline{N} W}$$



**2Q: (2.5)** Usando funções MSI (menos contador e registrador), FFs de sua preferência e portas projetar um **contador síncrono** crescente programável de 4 bits, que realiza a tabela de operações descrita na figura 10. A variável síncrona *inicio*, inicializa a mudança de operação (*inicio*=1 inicializa a operação).

Clk	S	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>
↑	0	módulo variável com início zero			
↑	1	módulo 8 com início variável			

Figura 10 – Tabela de operações.

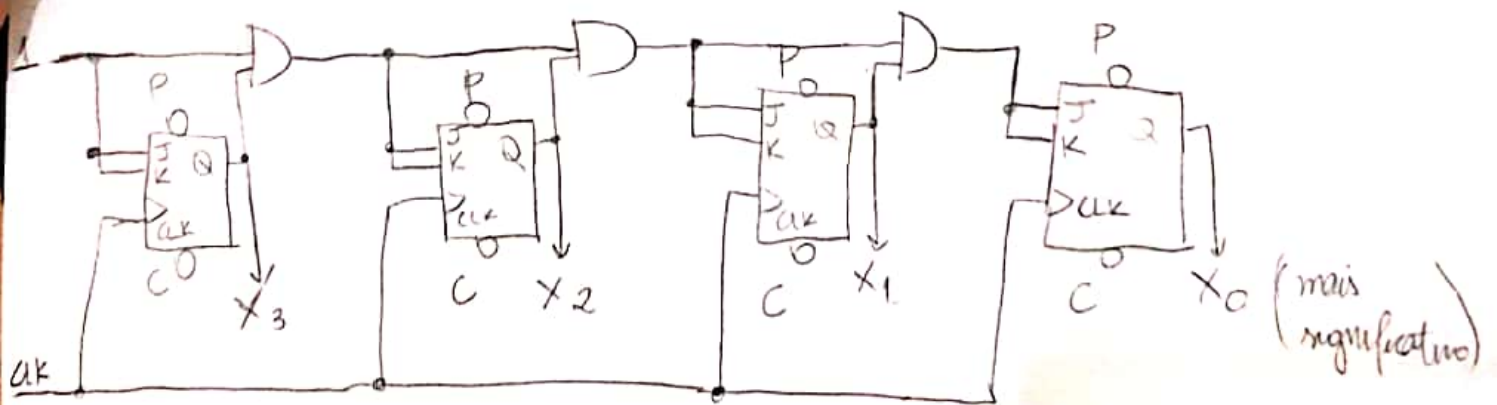
**Obs:** Ilustrando o módulo 8 com início variável: Por exemplo, com o início=10, então temos:  $10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 0 \rightarrow 1 \rightarrow 10 \dots$

Ilustrando o módulo variável com início zero: Por exemplo, com módulo=7, então temos:  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 0 \dots\dots$

$S=1$  módulo 8 com uma variável

- a soma do valor de início com 3 ignorando o cast indica o valor em que a contagem deve ser resetada

Contador interno 4 bits:



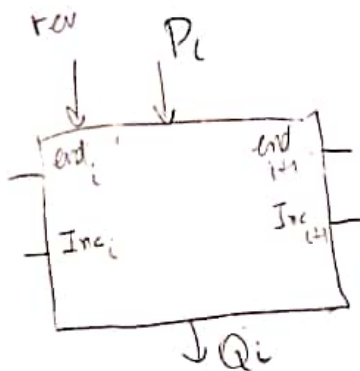
$S=0 \rightarrow$  usar comparador e, se for igual a módulo variável, acionar CLEPR de todos FF's

$S = 1 \rightarrow$  Comparar octogon máxima e se for igual, usar CLEAR/PRESET





**3Q (3.0):** Usando a técnica de redes iterativa sequencial, flip-flops T e portas, sintetize uma célula básica de um contador síncrono binário reversível de N bits, com início variável. Pede-se célula básica otimizada e o esquema células do contador de N bits



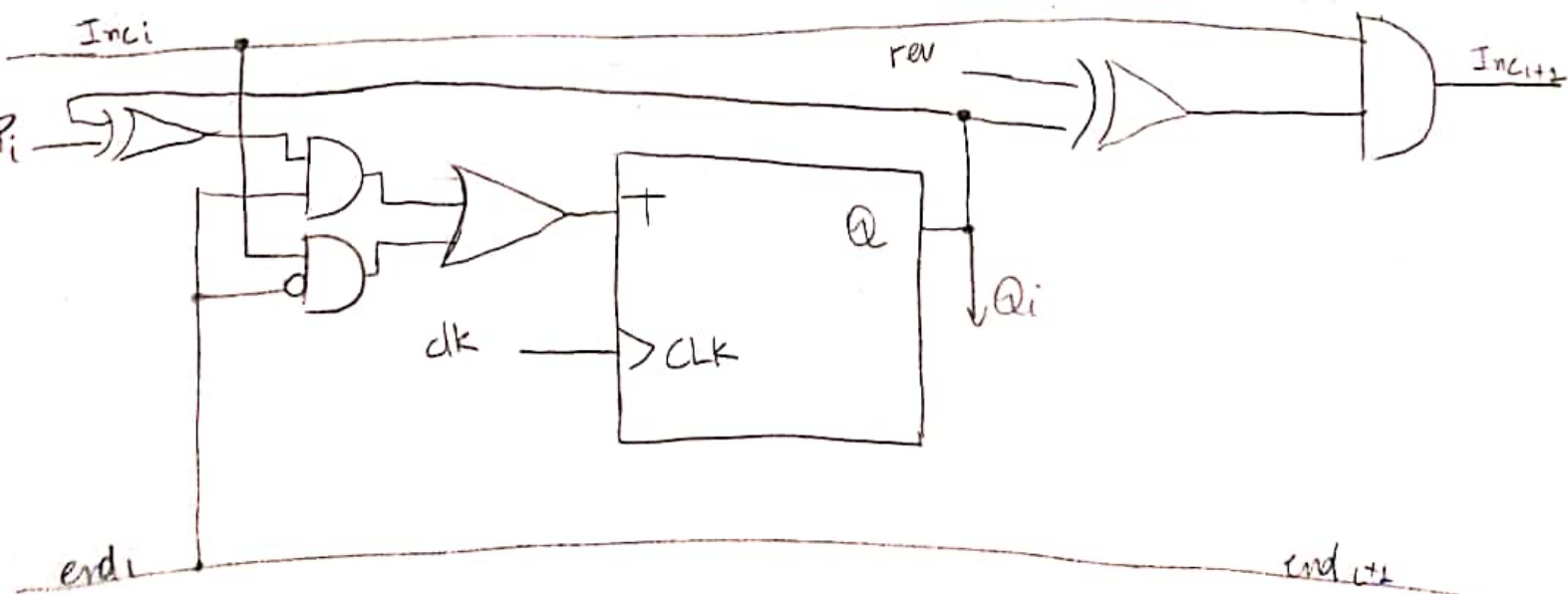
end → indica fim da contagem  
 inc → indica se célula deve incrementar no clock  
 Pi → bit de início automático

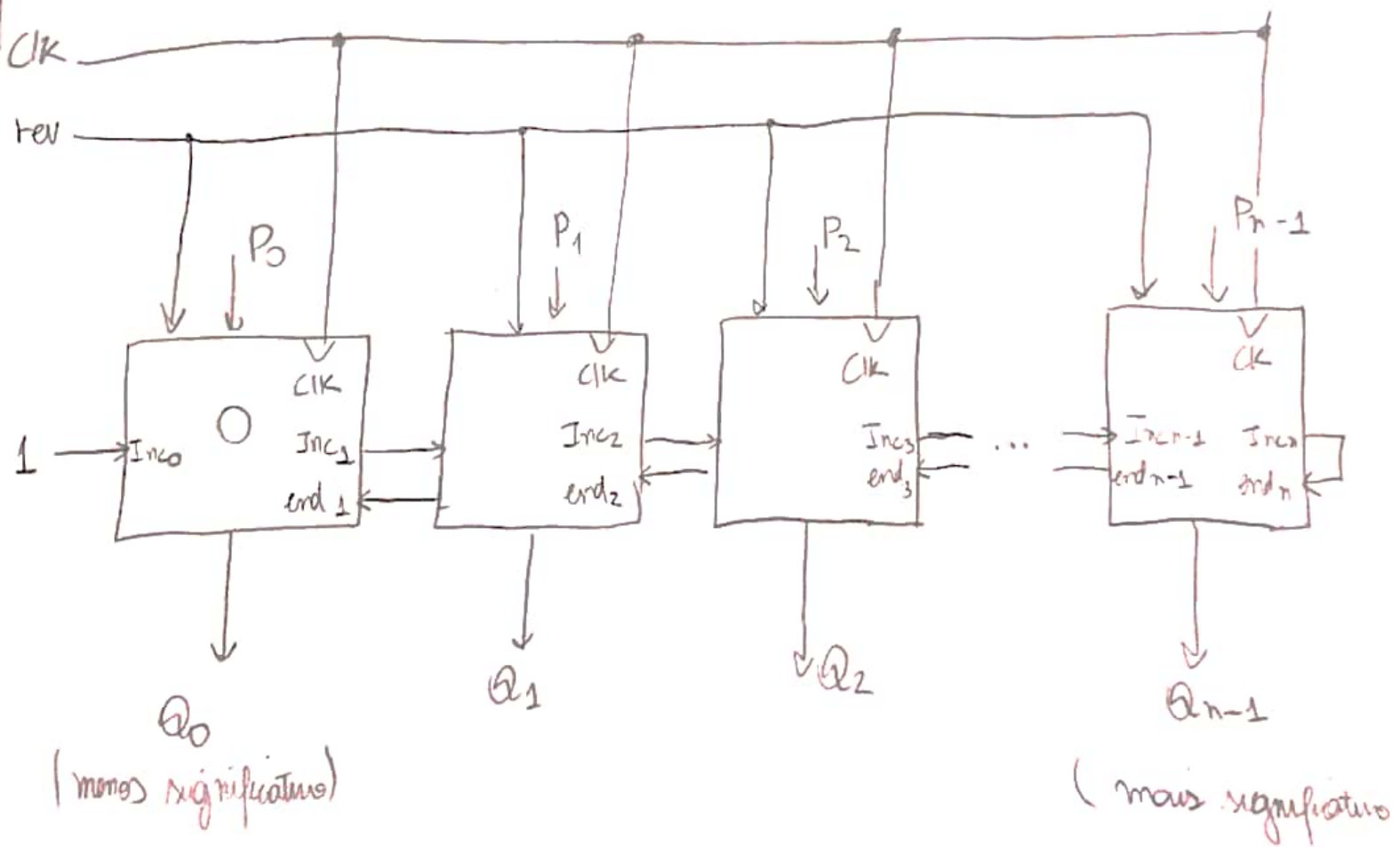
$$Inc_{i+1} = \overline{rev} \cdot Inc_i \cdot Q_i + rev \cdot Inc_i \cdot \overline{Q_i} = Inc_i (Q_i \oplus rev)$$

$$Inc_0 = 1$$

end\_i = Inc\_n no contagem chega ao fim no "incremento" do mais significativo  
 quando end\_i = 1  $T = P_i \oplus Q_i$

$$T = \overline{end_i} \cdot Inc_i + end_i (P_i \oplus Q_i)$$





4Q (2.5) Usando funções MSI, FF's e portas, projete uma **fechadura digital** ( ver figura 1 – esquema) que se a sequência 50, 100 e 200 for feita à tranca se abre (saída  $Z_3=1$ ) caso contrário à tranca não se abre (saída  $Z_3=0$ ). Duas saídas  $Z_1$  e  $Z_2$  mostram se a sequência está correta. Quando  $Z_1=1$  e  $Z_2=0$  a **sequência está errada**; quando  $Z_1=0$  e  $Z_2=1$  a **sequência está correta**; quando  $Z_1=Z_2=0$  **inicio da sequência** deve ser introduzida; quando  $Z_1=Z_2=1$  é **fim da sequência** e está correta. A entrada X de 8 bits é usada para gerar a sequência (entrada BCD). A variável **inicio**=0, a fechadura está trancada; para **inicio**=1, começa a leitura do código (X). A variável **inicio** é síncrona.

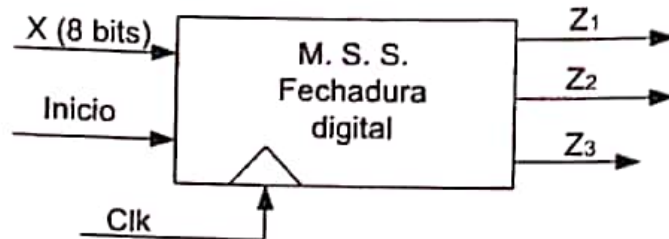
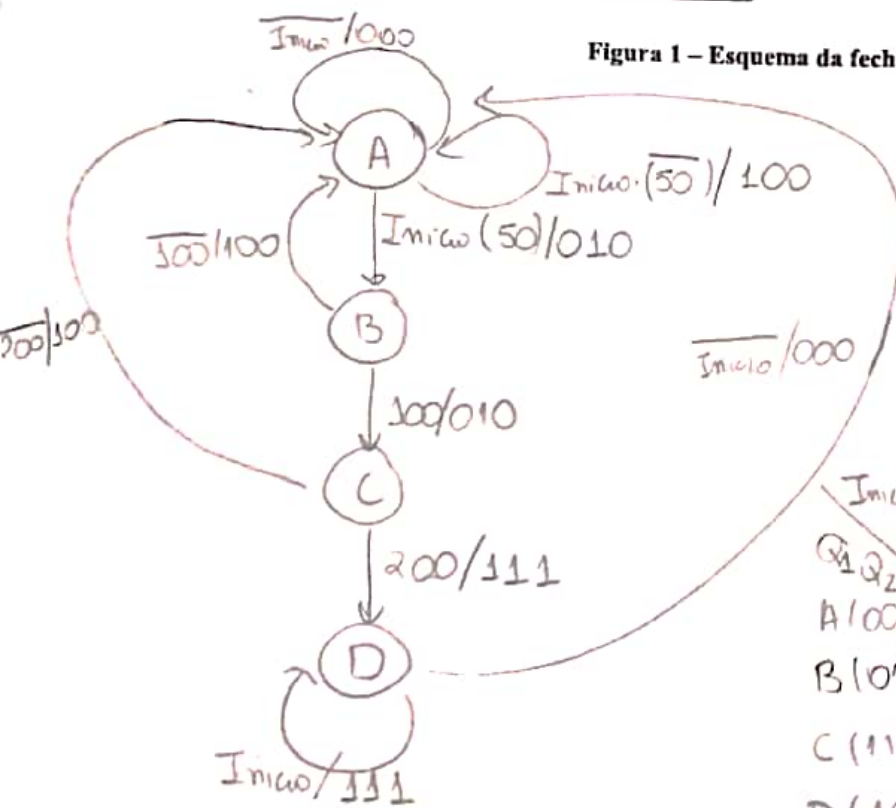


Figura 1 – Esquema da fechadura digital.



$Q_1 Q_2$	Comp	00	01	11	10
A (00)		00/000	00/000	01/010	00/100
B (01)		00/100	11/010	11/010	00/100
C (11)		00/100	10/111	10/111	00/100
D (10)		00/000	00/000	10/111	10/111

$$Q_1' = \text{Comp} Q_2 + Q_1 \overline{Q_2} \text{ Inicio}$$

$$Q_2' = \text{Inicio} \text{Comp} \overline{Q_1} + \text{Comp} \overline{Q_1} Q_2 = \text{Comp} \overline{Q_1} (\text{Inicio} + Q_2)$$

$$Z_1 = \text{Inicio} \text{Comp} + Q_1 Q_2 + \text{Comp} Q_2 + \text{Inicio} Q_1$$

$$Z_2 = \text{Inicio} \text{Comp} + \text{Comp} Q_2 + Q_1 \overline{Q_2} \text{ Inicio}$$

$$Z_3 = Z_1 Z_2$$

