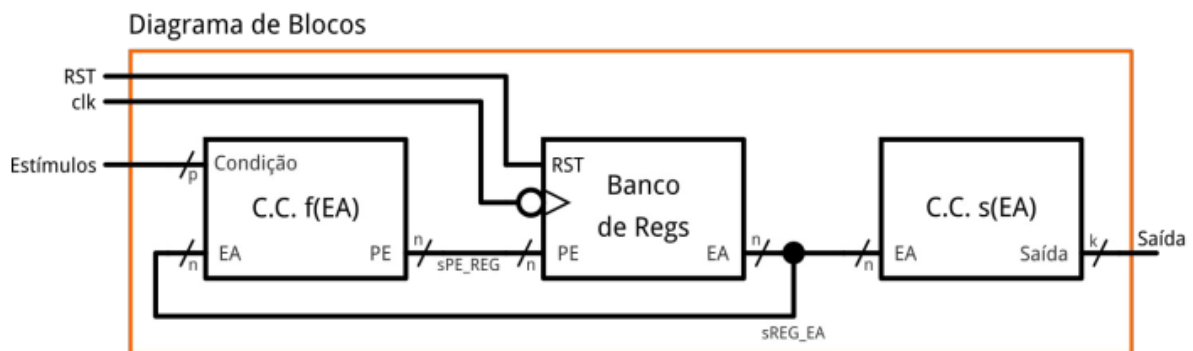


Máquina de Estado

Componentes da Máquina de Estados:

É um autômato finito com vários estados que mudam de acordo com o estado atual e o estímulo recebido e também tem saída(s) de acordo com o estado atual.

Usa dois circuitos combinacionais e um banco de registradores, os C.Cs são o de função de próximo estado, que recebe o estado atual, os estímulos e tem como saída o próximo estado, e o de saída, que a partir do estado atual oferece as saídas do sistema.



Entradas e saídas:

Entradas:

- Moeda de 50 centavos (m50);
- Moeda de 25 centavos (m25).

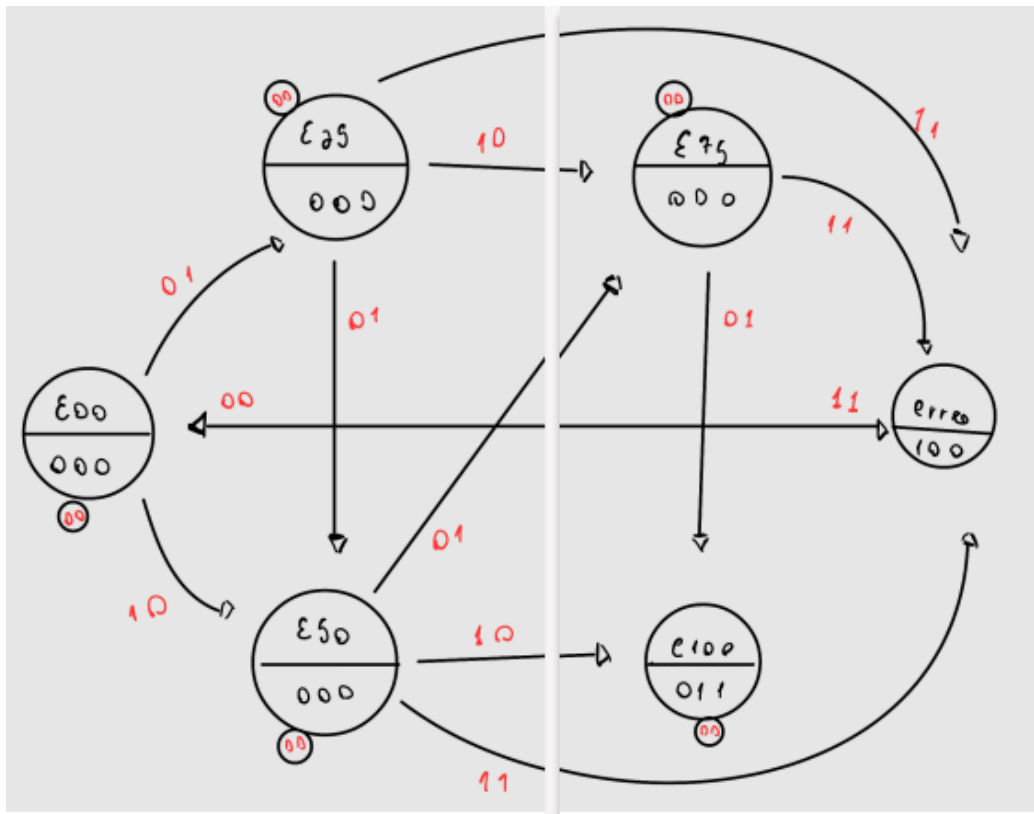
Saídas:

- Dispenser de chocolates (disp);
- Colocar as moedas no cofre (cof);
- Devolver as moedas (dev);
- Bloquear o recebimento de moedas (m_rec).

Estados:

- E₀₀: 000;
- E₅₀: 001;
- E₂₅: 010;
- E₇₅: 011;
- E₁₀₀: 100;
- E_{erro}: 111;
- E_{NP}: XXX(101 e 110).

Diagrama de Estados:



Função de Próximo Estado:

As entradas são os 3 bits do estado atual e os 2 estímulos, as saídas são o próximo estado, como representado na tabela:

EA	E(2)	E(1)	E(0)	M50	M25	PE	E(2)'	E(1)'	E(0)'
E00	0	0	0	0	0	E00	0	0	0
				0	1	E25	0	1	0
				1	0	E50	0	0	1
				1	1	Erro	1	1	1
E50	0	0	1	0	0	E50	0	0	1
				0	1	E75	0	1	1
				1	0	E100	1	0	0
				1	1	Erro	1	1	1
E25	0	1	0	0	0	E25	0	1	0
				0	1	E50	0	0	1
				1	0	E75	0	1	1
				1	1	Erro	1	1	1
E75	0	1	1	0	0	E75	0	1	1
				0	1	E100	1	0	0
				1	0	Erro	1	1	1
				1	1	Erro	1	1	1
E100	1	0	0	0	0	E100	1	0	0
				0	1	Erro	1	1	1
				1	0	Erro	1	1	1
				1	1	Erro	1	1	1

ENP	1	0	1	0 0 1 1	0 1 0 1	X	X	X	X
ENP	1	1	0	0 0 1 1	0 1 0 1	X	X	X	X
Eerro	1	1	1	0 0 1 1	0 1 0 1	E00 E25 E50 Eerro	0 0 0 1	0 1 0 1	0 0 1 1

Q3

e2

$e_2' = (e_2, e_0, m_{50}) + (m_{25}, m_{50}) + (e_2, e_1)$

e0

e1

$e_1' = (e_1, m_{25}) + (m_{25}, m_{50}) + (e_0, e_1, m_{25}) + (e_2, e_0, m_{50}) + (e_2, e_0, m_{25})$

e3

e2

$e_0' = (m_{25}, m_{50}) + (e_0, m_{50}) + (e_1, e_0, m_{25}) + (e_1, e_0, m_{50}) + (e_2, e_1, m_{25}) + (e_1, e_0, m_{50}) + (e_2, e_0, m_{50})$

Que, depois de simplificar com DVKs, fica assim:

```

pe(2) <= (m50 and m25) or (not ea(2) and ea(0) and m50) or (not ea(2) and ea(1) and ea(0) and m25);

pe(1) <= (not ea(1) and m25) or (m50 and m25) or (not ea(2) and ea(1) and not m25) or (ea(2) and m25);

pe(0) <= (not ea(0) and m50) or (ea(1) and m50) or (ea(0) and not ea(1) and m25) or (not ea(0) and |
| ea(1) and m25) or (ea(0) and ea(2) and not m25 and not m50);

```

Função de saída:

As entradas são os 3 bits do estado atual, que vem dos registradores.

EA	E(2)	E(1)	E(0)	Disp	Cof	Dev	M_rec
E00	0	0	0	0	0	0	1
E50	0	0	1	0	0	0	1
E25	0	1	0	0	0	0	1
E75	0	1	1	0	0	0	1
E100	1	0	0	1	1	0	0
Enp	1	0	1	X	X	X	X

E _{np}	1	1	0	X	X	X	X
E _{erro}	1	1	1	0	0	1	0

DVKs:

Dispenser	~E1	~E1	E1	E1
~E2	0	0	0	0
E2	1	X	0	X
	~E0	E0	E0	~E0

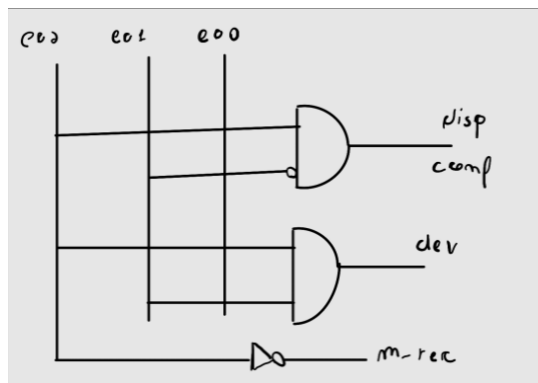
Cofre	~E1	~E1	E1	E1
~E2	0	0	0	0
E2	1	X	0	X
	~E0	E0	E0	~E0

Devolver	~E1	~E1	E1	E1
~E2	0	0	0	0
E2	0	X	1	X
	~E0	E0	E0	~E0

M _{rec}	~E1	~E1	E1	E1
~E2	1	1	1	1
E2	0	X	0	X
	~E0	E0	E0	~E0

Que, ao simplificar, fica assim:

```
disp <= ea(2) and not ea(1);
cof <= ea(2) and not ea(1);
dev <= ea(1) and ea(2);
m_rec <= not ea(2);
```



Banco de registradores:

Basicamente, são 3 registradores, um para cada bit de entrada, ou seja, 3 FlipFlops do tipo D, que guardam os bits do estado atual, recebidos da C.C. fPE. As saídas vão para o fPE e o fS, implementado assim:

```
r0 : ffd port map(din(0), clk, '1', cl, dout(0));
r1 : ffd port map(din(1), clk, '1', cl, dout(1));
r2 : ffd port map(din(2), clk, '1', cl, dout(2));
```

Implementação

A máquina de estados foi implementada conforme segue o print:

```
42 -- Sinais internos
43 signal spe_reg, sreg_ea : std_logic_vector(2 downto 0);
44
45 begin
46
47 -- f(ea)
48 u_fpe : fpe port map(sreg_ea, m25, m50, spe_reg);
49
50 -- registradores
51 u_reg : bancoReg3b port map(spe_reg, rst, clk, sreg_ea);
52
53 -- s(ea)
54 u_fs : fs port map(spe_reg, disp, cof, dev, m_rec);
55
```

Simulação:

O primeiro teste foi recebendo duas moedas de 25 e uma de 50, transicionando entre os estados E00, E25, E75 e E100, e ativando as saídas de cofre, dispenser e desativando o m_rec.

O segundo foi recebendo duas moedas de 50, indo de E00->E50->E100 e ativando as mesmas saídas.

O terceiro teste foi recebendo duas moedas ao mesmo tempo, $m25 \leq '1'$ e $m50 \leq '1'$, simulando um erro, e ativando a devolução de moedas:

