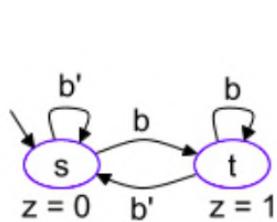


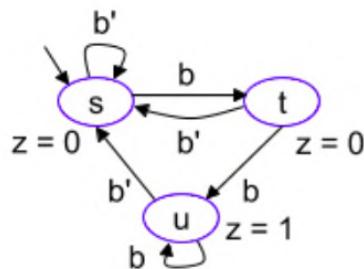
[1 ponto] 1. Relacione as FSMs abaixo com suas descrições

Inputs: b  
Outputs: z



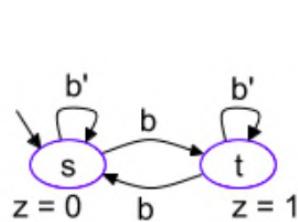
(a)

Inputs: b  
Outputs: z



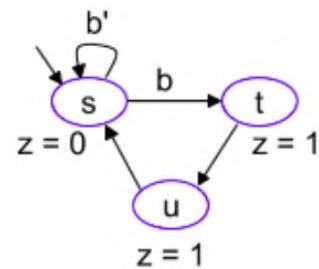
(b)

Inputs: b  
Outputs: z



(c)

Inputs: b  
Outputs: z



(d)

- (**d**) Quando b é 1, z é 1 por dois ciclos de clock.  
(**a**) Quando b é 1, z é 1. Quando b é 0, z é 0.  
(**b**) Quando b for 1 por pelo menos dois ciclos de clock z se torna 1 e permanece em 1 até que b = 0.  
(**C**) Quando b é 1, z alterna entre 0 e 1. Quando b é 0 z mantém seu valor atual.

[1 ponto] 2. Indique os valores de (a), (b), (c), (d) e (e) no diagrama de tempo abaixo.

Inputs: j      Outputs: y, z

- a) s
- b) 0
- c) s
- d) t
- e) 0
- f) u

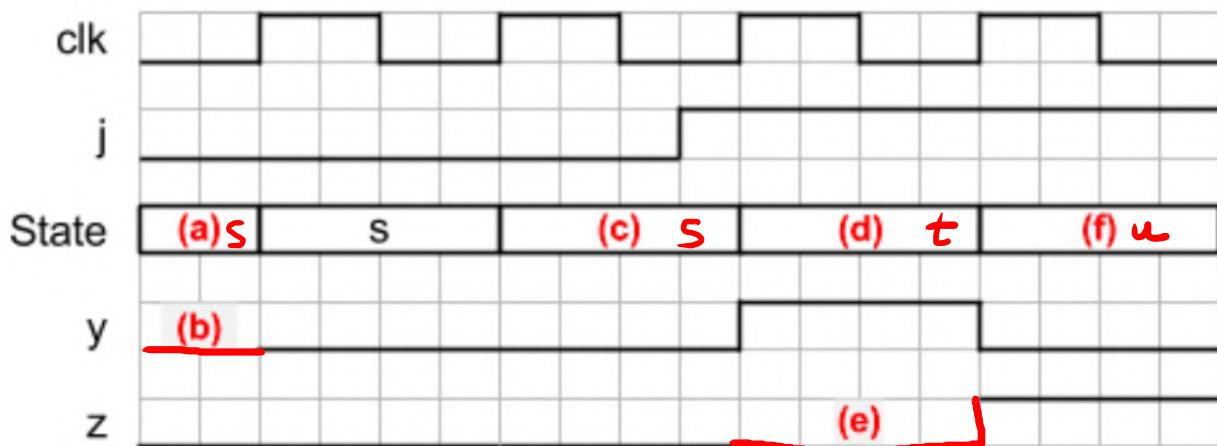
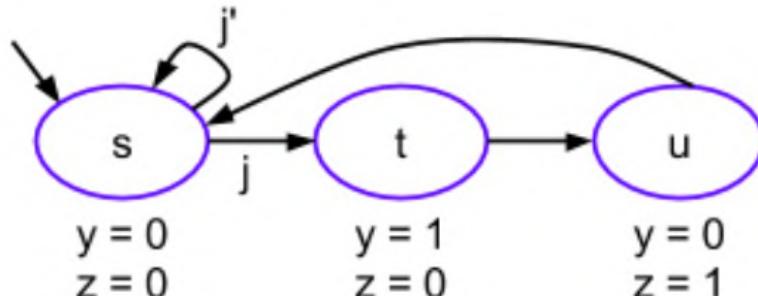
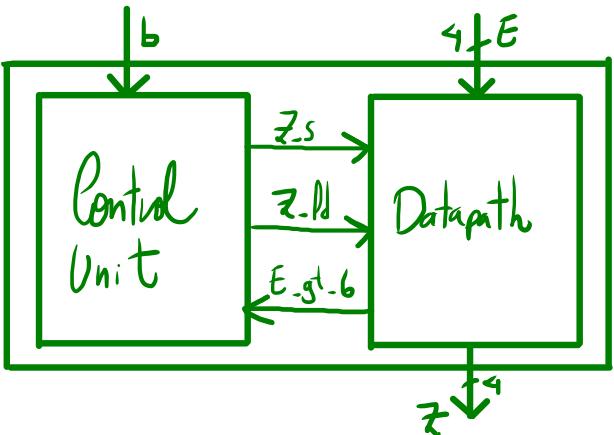
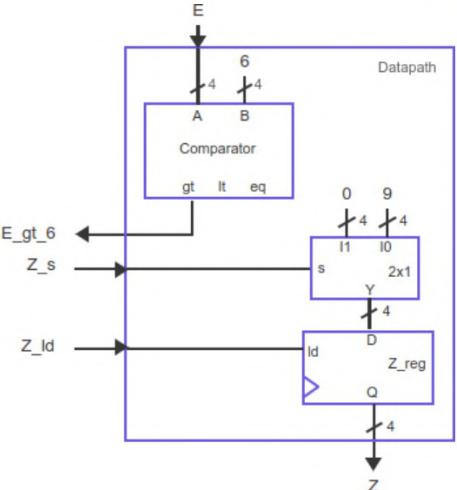
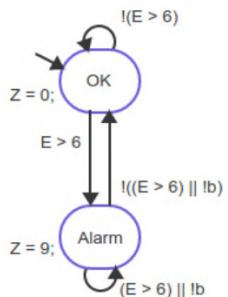


Figura 1

[1 ponto] 3. Desenhe o bloco de controle e a tabela de estados correspondentes ao datapath e à FSM abaixo.

Inputs: b, E(4)  
Outputs: Z(4)



\* FSM de Moore:  
 $Z_{-s} = y$ ,  $Z_{-ld} = 1$ .

Estado Atual	Entradas		Estado Próx.
	b	E_gt_6	
OK	X	0	OK
Alarm	X	1	Alarm

Estado Atual	Entradas		Estado Próx.
	b	E_gt_6	
Alarm	0	0	Alarm
OK	1	1	OK

# \* Codificação dos Estados:

Estado Atual	Entradas		Estado Próx.
	b	E-gt-6	
OK	X X X X	0 1 0 1	OK Alarm OK Alarm
Alarm	0 0 1 1	0 1 0 1	0 1 0 1



Estado Atual	Entradas		Estado Próx.
	b	E-gt-6	
OK	X X X X	0 1 0 1	0 1 0 1
Alarm	0 0 1 1	0 1 0 1	0 1 0 1

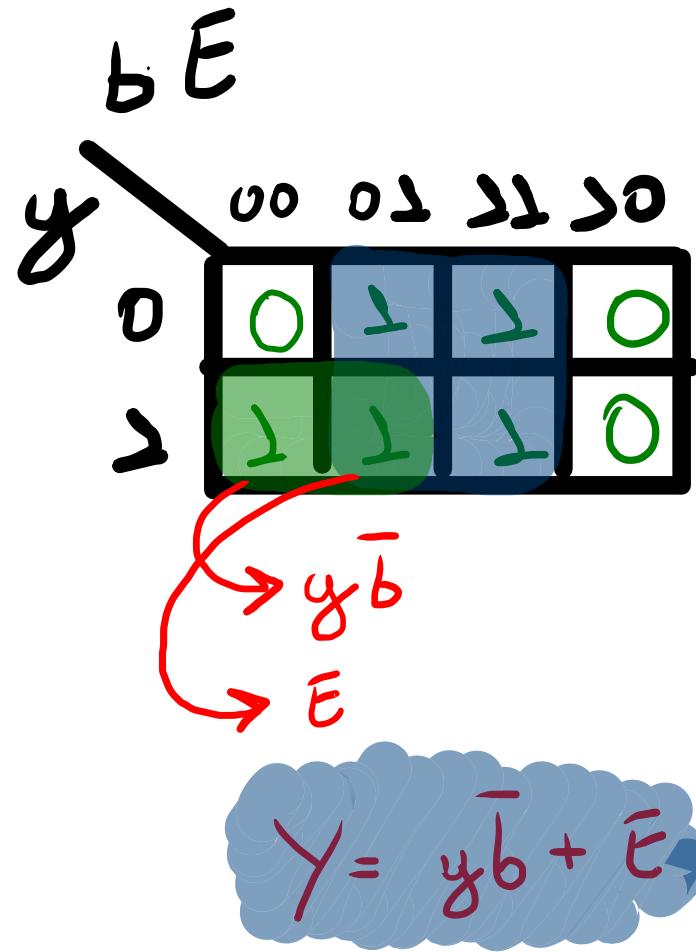
Estado Atual	Entradas		Estado Próx.
	b	E-gt-6	
OK	0 0 1 1	0 1 0 1	OK OK Alarm Alarm
Alarm	0 0 1 1	0 1 0 1	OK OK Alarm Alarm

Estado Atual	Entradas		Estado Próx.
	b	E-gt-6	
OK	0 0 1 1	0 1 0 1	OK OK Alarm Alarm
Alarm	0 0 1 1	0 1 0 1	OK OK Alarm Alarm

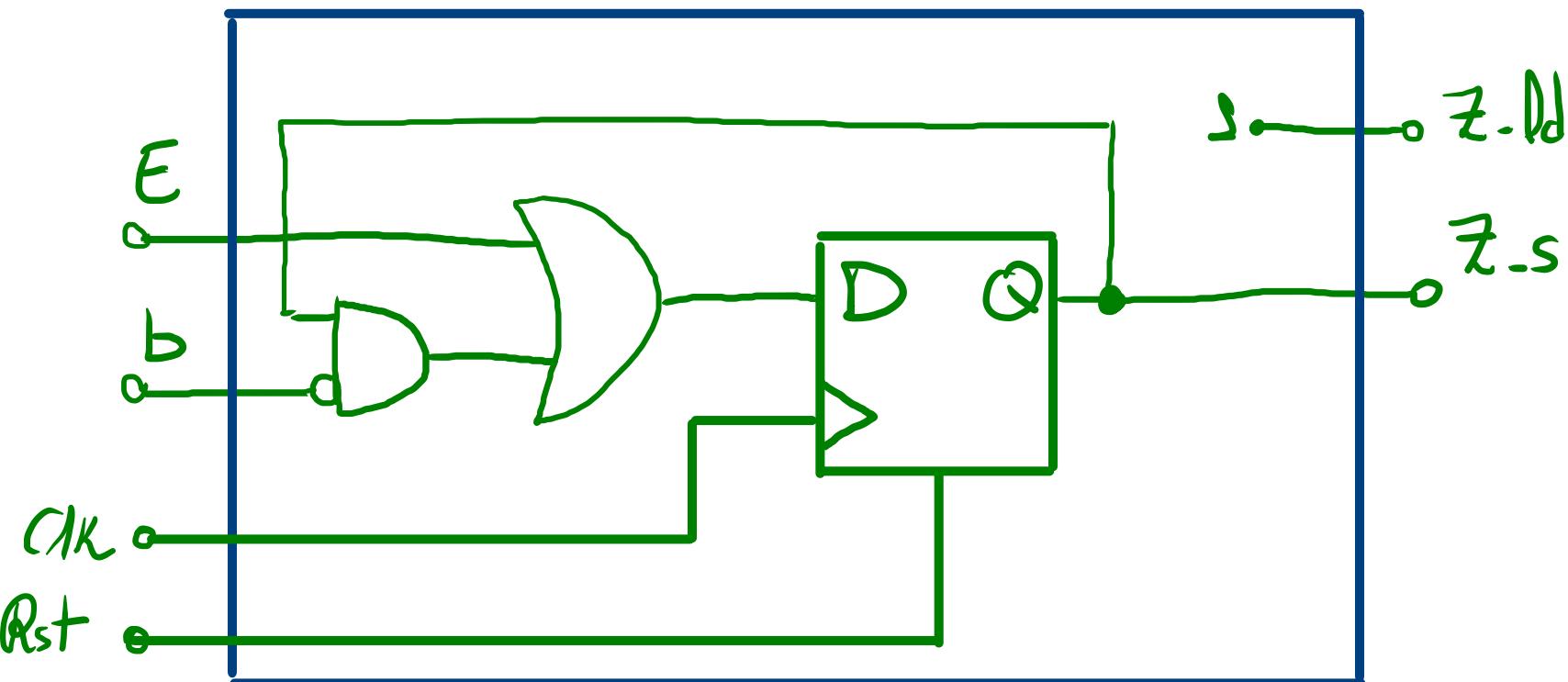
# \* Expressão do Próximo Estado

Estado Atual	Entradas		Estado Próx.
	b	E- $g_{t-6}$	
0	x x x x	0 1 0 1	0 1 0 1

Estado Atual	Entradas		Estado Próx.
	b	E- $g_{t-6}$	
1	0 0 1 1	0 1 0 1	1 1 0 1



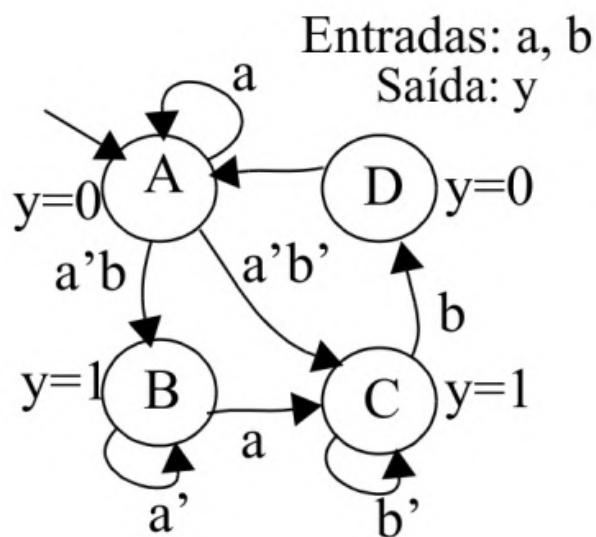
# \* Unidade de Controle:



$$Y = \bar{y}b + E$$

$$Z_S = Y \quad Z_{Dd} = 1$$

[1 ponto] 4. Usando o processo para projetar um bloco de controle, converte o FSM em um bloco de controle, implementando o controlador usando um registro de estado e portas lógicas.



Estado Atual	Próximo Estado				Saída
	$a = 0$	$a = 0$	$a = 1$	$a = 1$	
$b = 0$	$b = 1$	$b = 0$	$b = 1$	y	
A	C	B	A	A	0
B	B	B	C	C	1
C	C	D	C	D	1
D	A	A	A	A	0

# \* Codificação dos Estados

Estado Atual	Próximo Estado				Saída
	a = 0	a = 0	a = 1	a = 1	
	b = 0	b = 1	b = 0	b = 1	
A	C	B	A	A	0
B	B	B	C	C	1
C	C	D	C	D	1
D	A	A	A	A	0

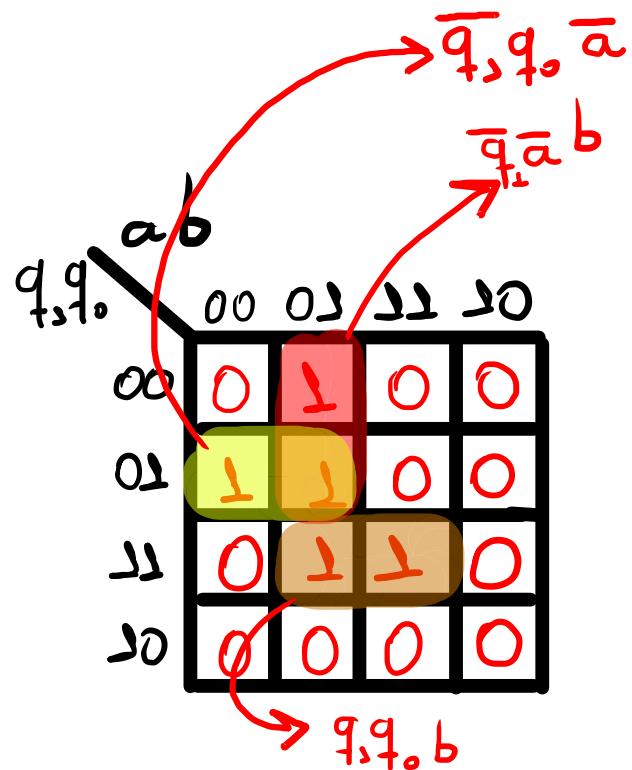
→

Estado Atual	Próximo Estado				Saída
	a = 0	a = 0	a = 1	a = 1	
	b = 0	b = 1	b = 0	b = 1	
00	10	01	00	00	0
01	01	01	10	10	1
10	10	11	10	11	1
11	00	00	00	00	0

$$\hookrightarrow y = q_0 \oplus q_1$$

# Estado $Q_0$

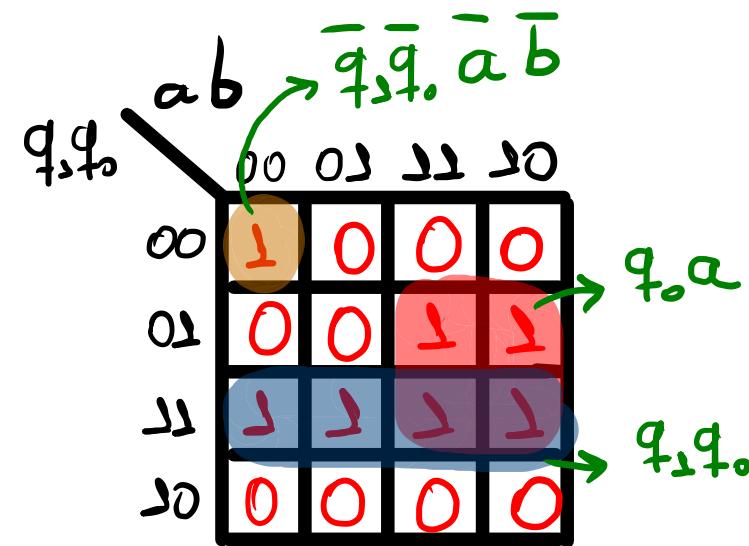
Estado Atual	Próximo Estado				Saída
	$a = 0$	$a = 0$	$a = 1$	$a = 1$	
$b = 0$	$b = 1$	$b = 0$	$b = 1$	$b = 0$	$y$
00	20	01	00	00	0
01	02	01	20	20	1
20	20	21	20	21	1
21	00	00	00	00	0



$$Q_0 = q_t q_0 b + \bar{q}_t q_0 \bar{a} + \bar{q}_t \bar{a} b$$

# Estado $Q_2$

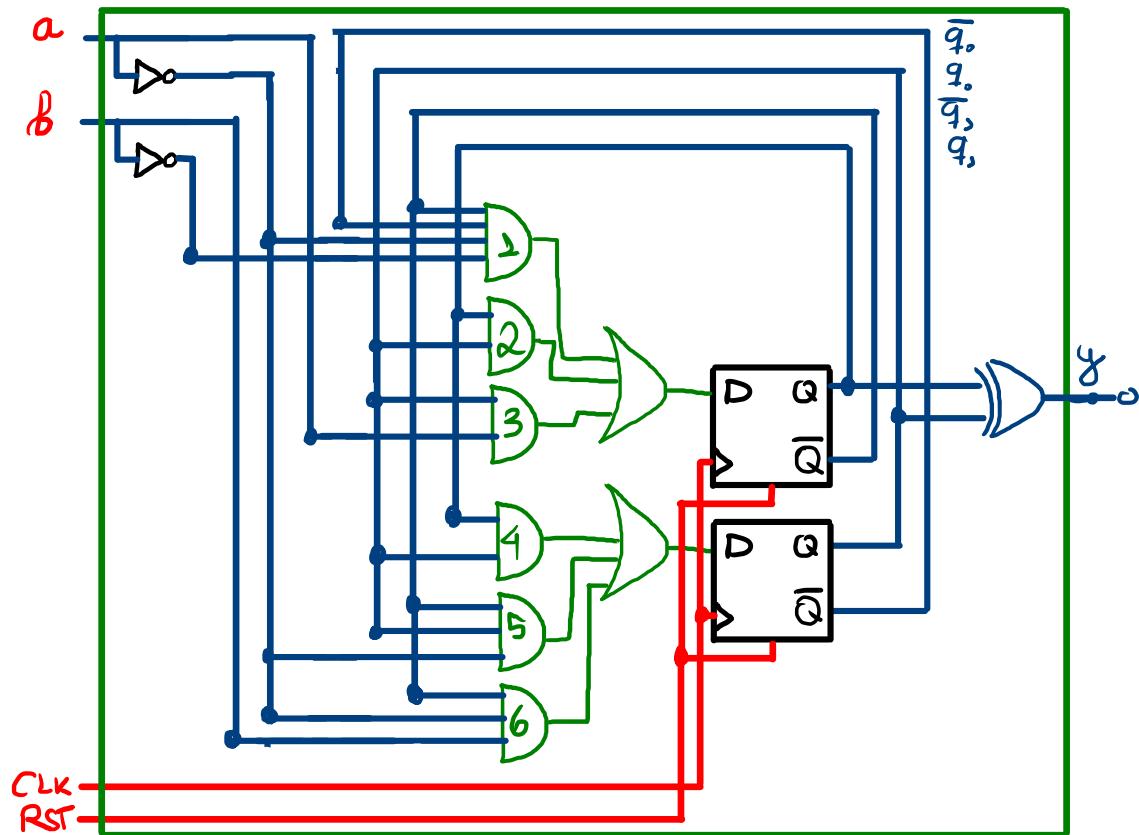
Estado Atual	Próximo Estado				Saída
	$a = 0$	$a = 0$	$a = 1$	$a = 1$	
	$b = 0$	$b = 1$	$b = 0$	$b = 1$	
00	20	01	00	00	0
01	01	01	20	20	1
10	20	11	20	11	1
11	00	00	00	00	0



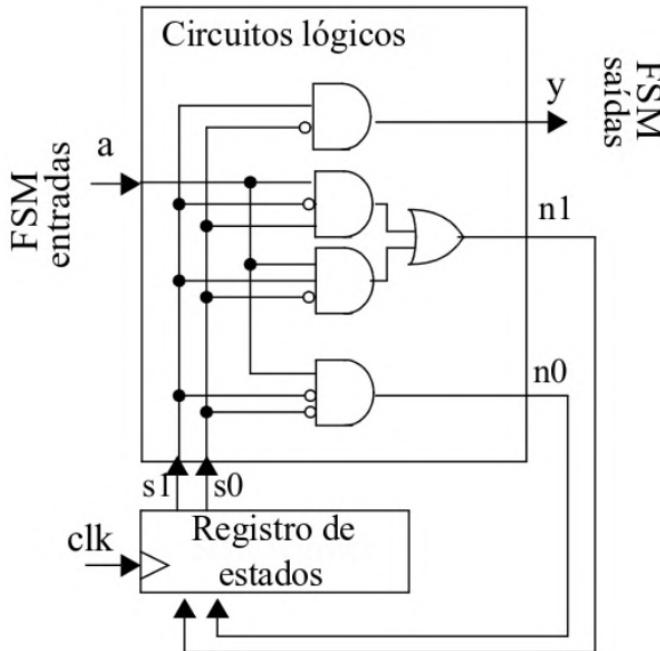
$$Q_2 = \bar{q}_2 \bar{q}_0 \bar{a} \bar{b} + q_0 a + q_1 q_0$$

## \* Bloco de Controle:

$$\left\{ \begin{array}{l} Q_0 = \underbrace{q_1 q_0}_1 b + \underbrace{\bar{q}_1 q_0}_5 + \underbrace{\bar{q}_1 \bar{a}}_6 \\ Q_1 = \underbrace{\bar{q}_1 \bar{q}_0 \bar{a} b}_2 + \underbrace{q_1 q_0}_3 + \underbrace{q_0 a}_4 \\ y = q_0 \oplus q_1 \end{array} \right.$$



[1 ponto] 5. Faça a engenharia reversa do comportamento do circuito sequencial.



$y = s_1 \cdot \bar{s}_0$  Saída

$n_1 = a \cdot \bar{s}_1 \cdot s_0 + a \cdot s_1 \cdot \bar{s}_0$

$n_0 = a \cdot \bar{s}_1 \cdot \bar{s}_0$

Próx. Estado

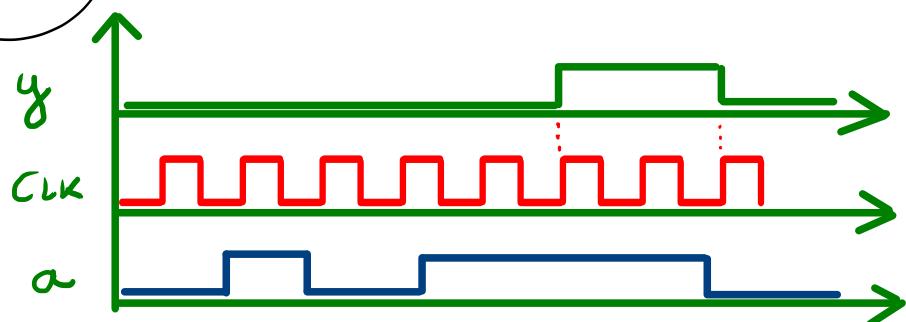
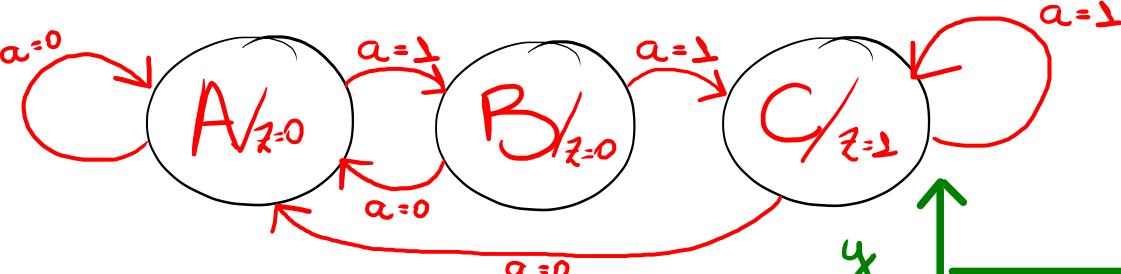
Estado Atual	Próximo Estado		Saída y
	$a=0$	$a=1$	
00	00	01	0
01	00	10	0
10	00	10	1
11	00	00	0

Estado Atual	Próximo Estado	Saida
	$a=0$	$a=1$
00	00	01
01	00	10
10	00	10
11	00	00



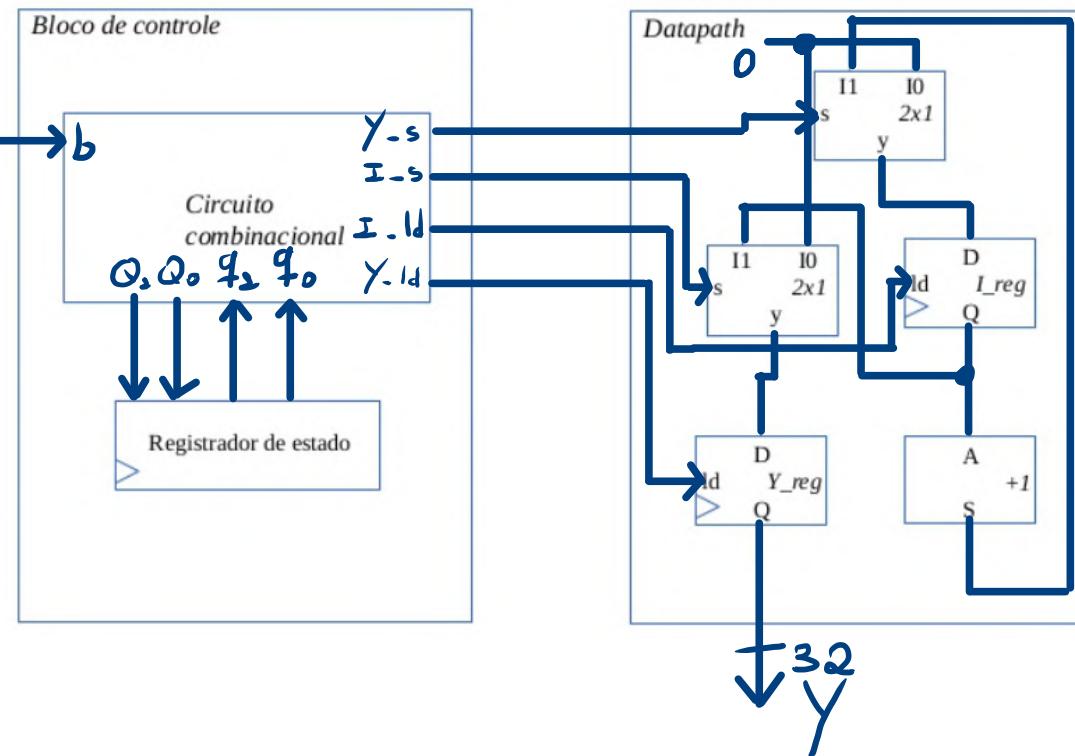
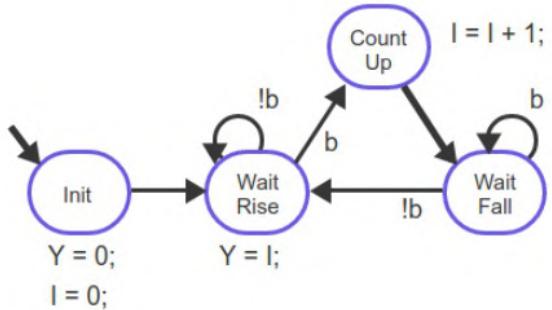
Estado Atual	Próximo Estado	Saida
	$a=0$	$a=1$
A	A	B
B	A	C
C	A	C

“Detecta quando a entrada permanece em alto por 2 ciclos de clock ou mais.”



[2 pontos] 6. Desenhe o circuito completo (bloco de controle e datapath) para realizar a computação referente à FSM mostrada abaixo.

Inputs: b  
Outputs: Y(32)  
Variables: I(32)



# \*Tabela de Transições

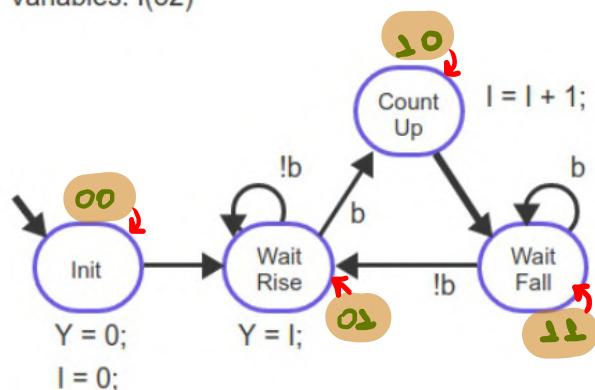
Control

- $Y_{-s}$  •  $Y_{-ld}$
- $I_{-s}$  •  $I_{-ld}$

Inputs: b

Outputs: Y(32)

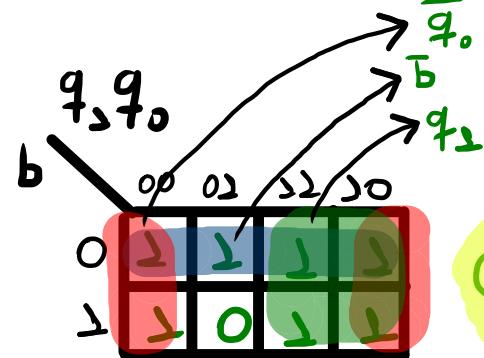
Variables: I(32)



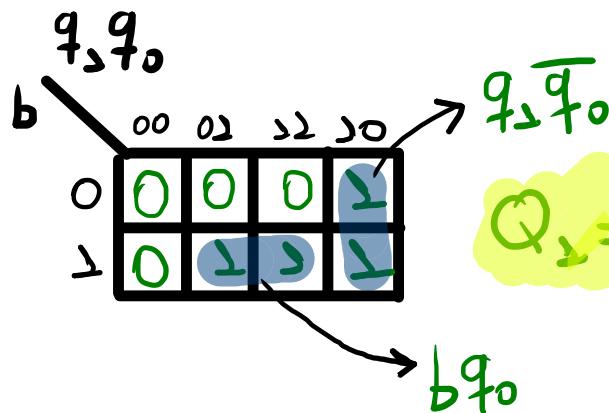
Estado Atual	Próximo Estado		Saídas			
	$b=0$	$b=1$	$Y_{-s}$	$I_{-s}$	$Y_{-ld}$	$I_{-ld}$
00	01	01	0	0	1	1
01	01	10	1	X	1	0
10	11	11	X	1	0	1
11	01	11	X	X	0	0

# Lógica de Próximo Estado:

Estado Atual	Próximo Estado	
	$b=0$	$b=1$
00	01	01
01	01	10
10	11	11
11	02	11



$$Q_0 = \bar{q}_0 + \bar{b} + q_1$$



$$Q_1 = q_1 \bar{q}_0 + b q_0$$

# \*Lógica de Saída:

Estado		Saídas			
$q_1$	$q_0$	$Y_S$	$I_S$	$Y_{Pd}$	$I_{Pd}$
0	0	0	0	1	1
0	1	1	X	1	0
1	0	X	1	0	1
1	1	X	X	0	0

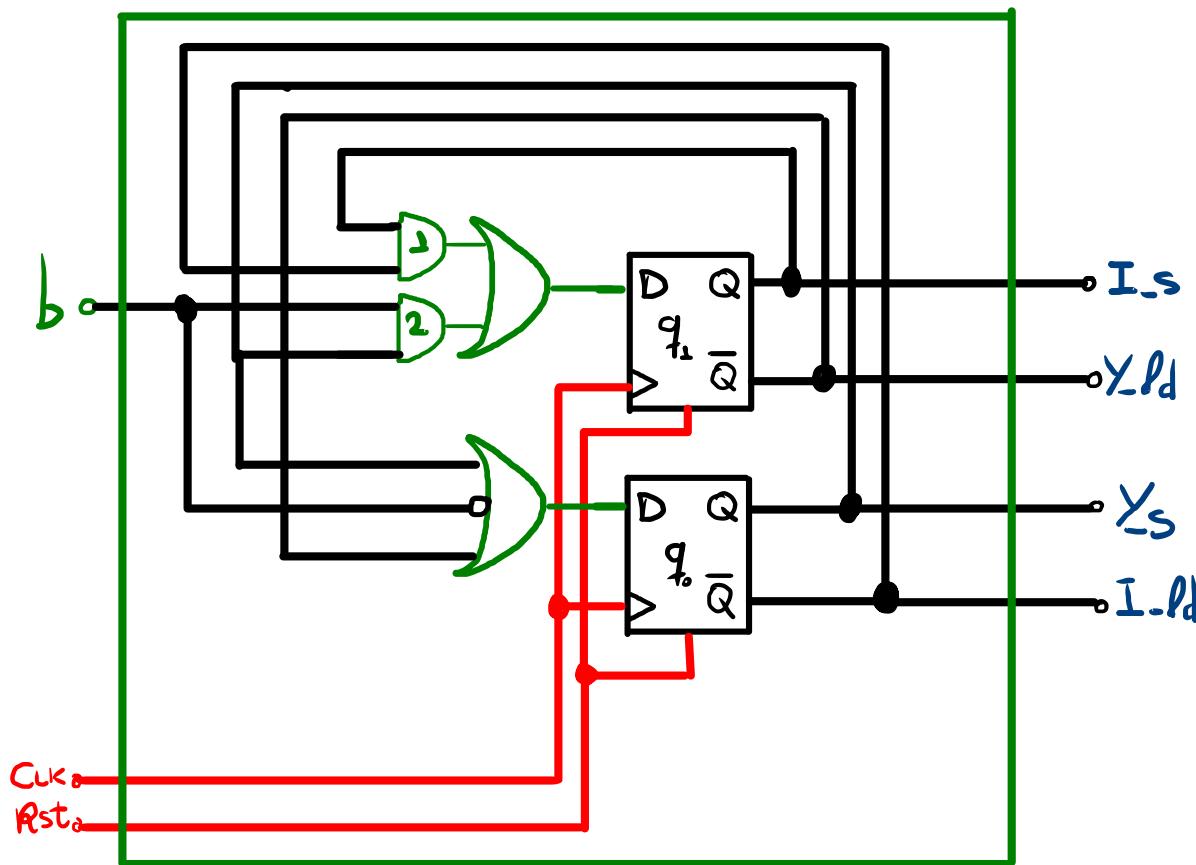
- $Y_S = q_0$
- $I_S = \bar{q}_1$
- $Y_{Pd} = \bar{q}_1$
- $I_{Pd} = \bar{q}_0$

$$\cancel{q_1 + q_0} \quad \cancel{q_1 q_0} \quad \downarrow \quad \bar{q}_1 \bar{q}_0 + q_1 \bar{q}_0 \Rightarrow \bar{q}_0 (\bar{q}_1 + q_1) \Rightarrow \bar{q}_0$$

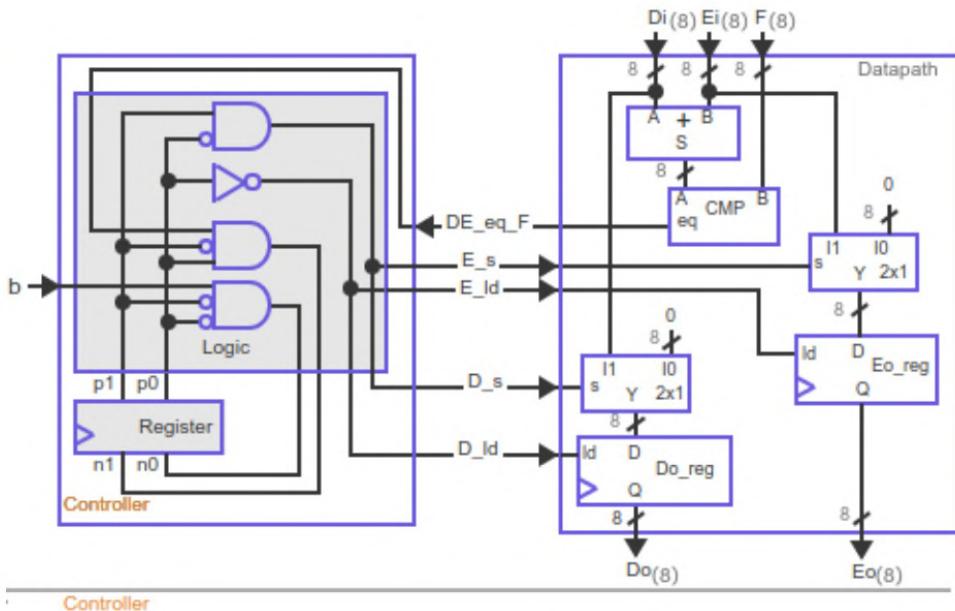
$$\bar{q}_1 \bar{q}_0 + \bar{q}_1 q_0 \Rightarrow \bar{q}_1 (\bar{q}_0 + q_0) \Rightarrow \bar{q}_1$$

# \* Unidade de Controle

- $Y_S = q_0$
- $I_{-S} = q_1$
- $Y_{Pd} = \bar{q}_1$
- $I_{-Pd} = \bar{q}_0$
- $Q_2 = \frac{\bar{q}_1 \bar{q}_0}{2} + \frac{b q_0}{2}$
- $Q_0 = \bar{q}_0 + b + q_1$



[2 pontos] 7. Mostre a FSM e a tabela de estados do bloco de controle para a arquitetura abaixo.



\* Saldas

- $E_S = P_1 \cdot \bar{P}_0$
- $D_{-S} = P_1 \cdot \bar{P}_0$
- $E_{-D_I} = \bar{P}_0$
- $D_{-D_I} = \bar{P}_0$

\* Próximo Estado:

- $n_1 = DE_{eq} \cdot F \cdot \bar{P}_1 \cdot P_0$
- $n_0 = b \cdot \bar{P}_1 \cdot \bar{P}_0$

# \*Tabela de Excitações:

Estado Atual	Próximo Estado				Saídas			
	$a=0$	$a=1$	$a=0$	$a=1$	$E_s$	$D_s$	$E_{Id}$	$D_{Id}$
	$b=0$	$b=0$	$b=1$	$b=1$				
00	00	00	01	01	0	0	1	1
01	00	10	00	10	0	0	0	0
10	00	00	00	00	1	1	1	1
11	00	00	00	00	0	0	0	0

↳ Obs:  $a = D_{Eq} - F$

## \* Saídas

$$E_s = P_1 \cdot \bar{P}_0 \quad \circ \quad D_s = P_1 \cdot \bar{P}_0$$

$$\circ \quad E_{Id} = \bar{P}_0 \quad \circ \quad D_{Id} = \bar{P}_0$$

## \* Próximo Estado:

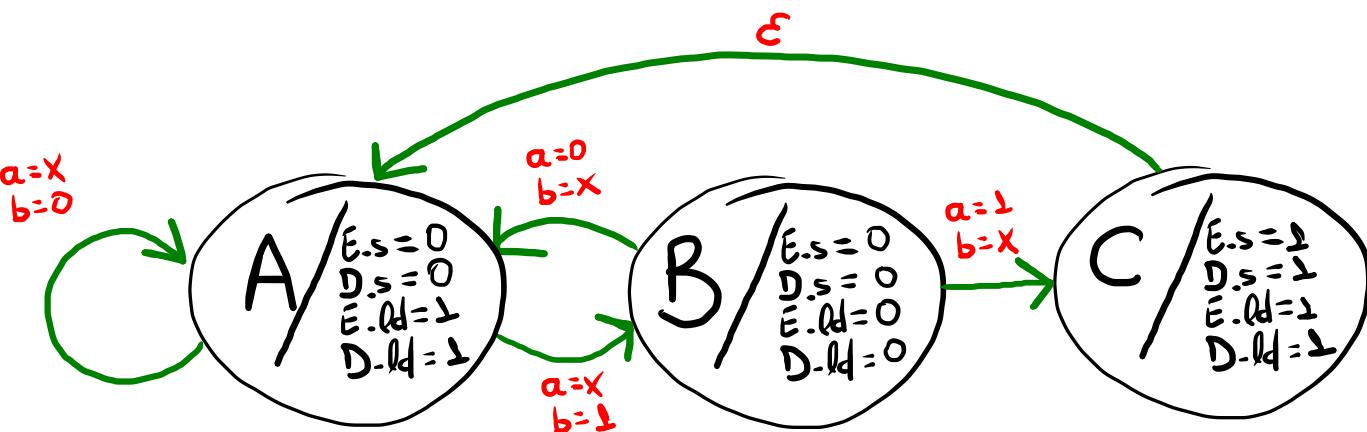
$$n1 = D_{Eq} - F \cdot \bar{P}_1 \cdot P_0$$

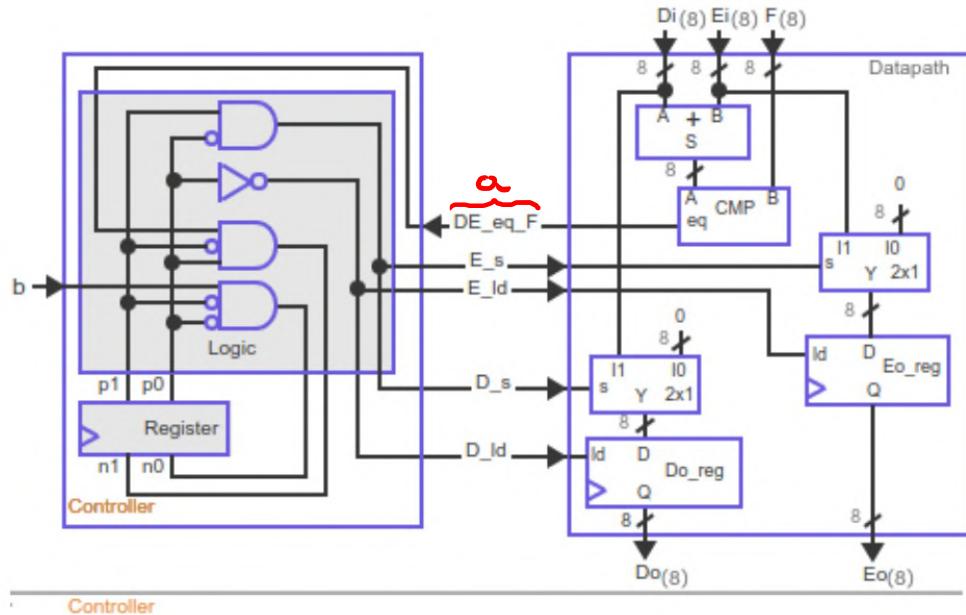
$$n0 = b \cdot \bar{P}_1 \cdot \bar{P}_0$$

# Transições de Estados

Estado Atual	Próximo Estado				Sádias			
	$a=0$	$a=1$	$a=0$	$a=1$	$E\_s$	$D\_s$	$E\_Rd$	$D\_Rd$
	$b=0$	$b=0$	$b=1$	$b=1$				
00	00	00	01	01	0	0	1	1
01	00	10	00	10	0	0	0	0
10	00	00	00	00	1	1	1	1
11	00	00	00	00	0	0	0	0

Estado Atual	Próximo Estado				Sádias			
	$a=0$	$a=1$	$a=0$	$a=1$	$E\_s$	$D\_s$	$E\_Rd$	$D\_Rd$
	$b=0$	$b=0$	$b=1$	$b=1$				
A	A	A	B	B	0	0	1	1
B	A	C	A	C	0	0	0	0
C	A	A	A	A	1	1	1	1





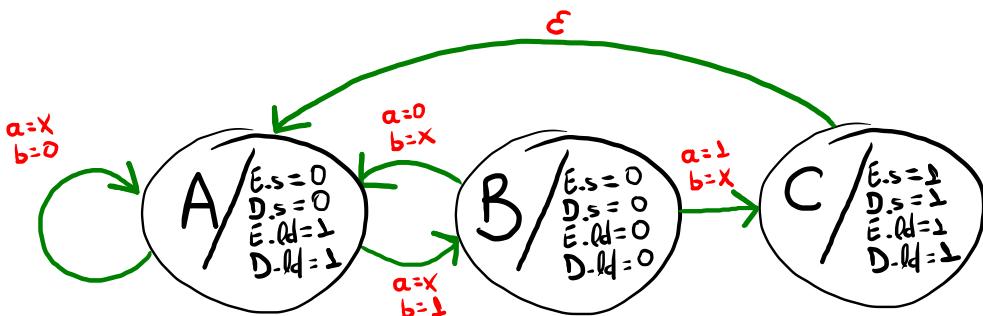
# \* Projeto RTh

Gestados:

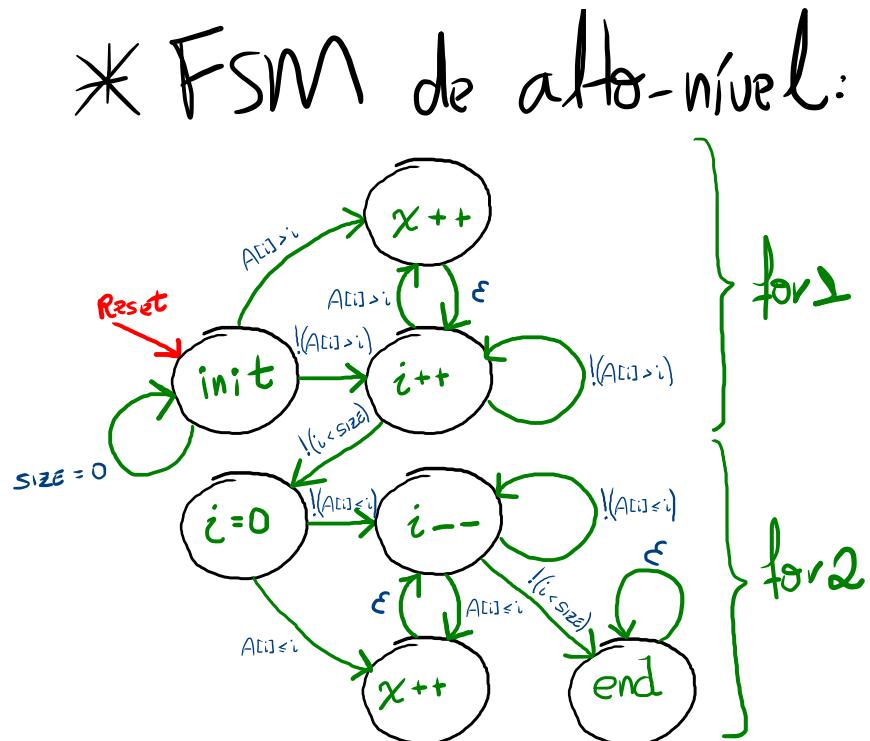
\*A: Carrega 0 em  $E_{o\_reg}$  e 0 em  $D_o\_reg$ .

\*B: Se  $D_i + E_i = F$ , vá para C, senão vá para A.

\*C: Carrega  $E_i$  em  $E_{o\_reg}$  e  $D_i$  em  $D_o\_reg$ .



[2 pontos] 8. Mostre como seria o circuito, incluindo os sinais de controle, para a implementação de cada um dos códigos abaixo.

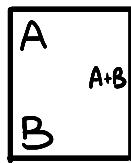
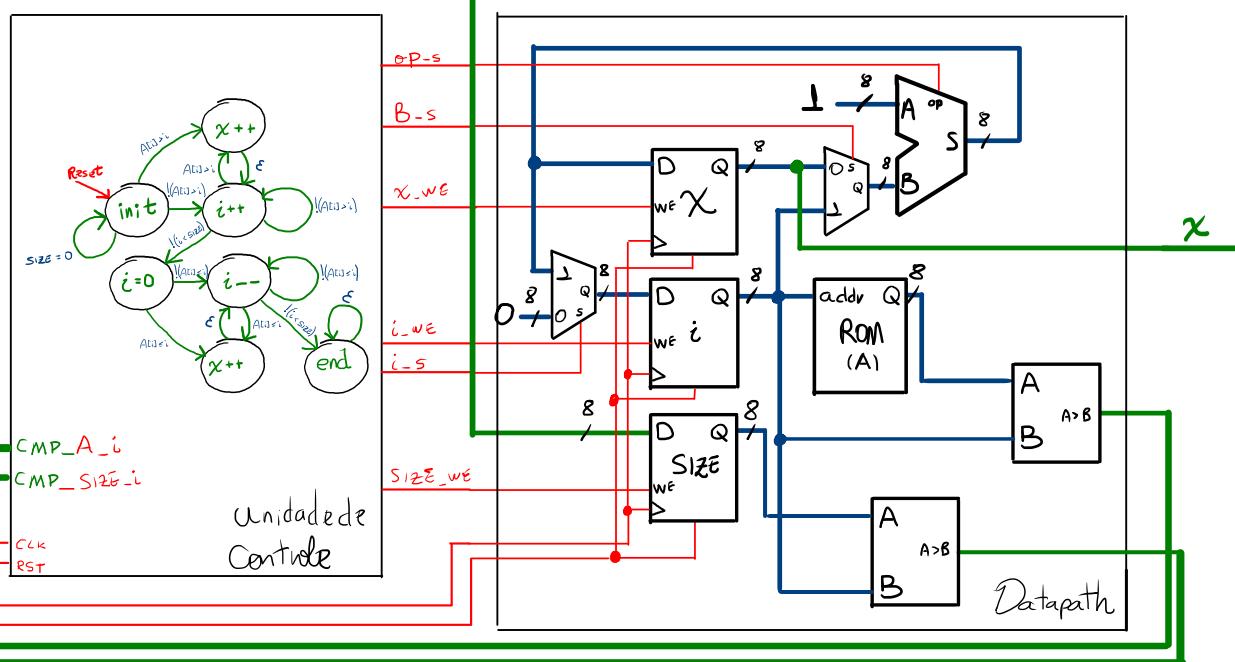


```
1 x = 0;
2 for (i=0; i<SIZE; i++){
3     if (A[i] > i){
4         x = x + 1;
5     }
6 }
7 for (i=0; i<SIZE; i++){
8     if (A[i] <= i){
9         x = x - 1;
10    }
11 }
```

→ Variáveis:  $i(8)$ ,  $A[256](8)$  → Entrada  $SIZE(8)$   
→ Saída  $x(8)$

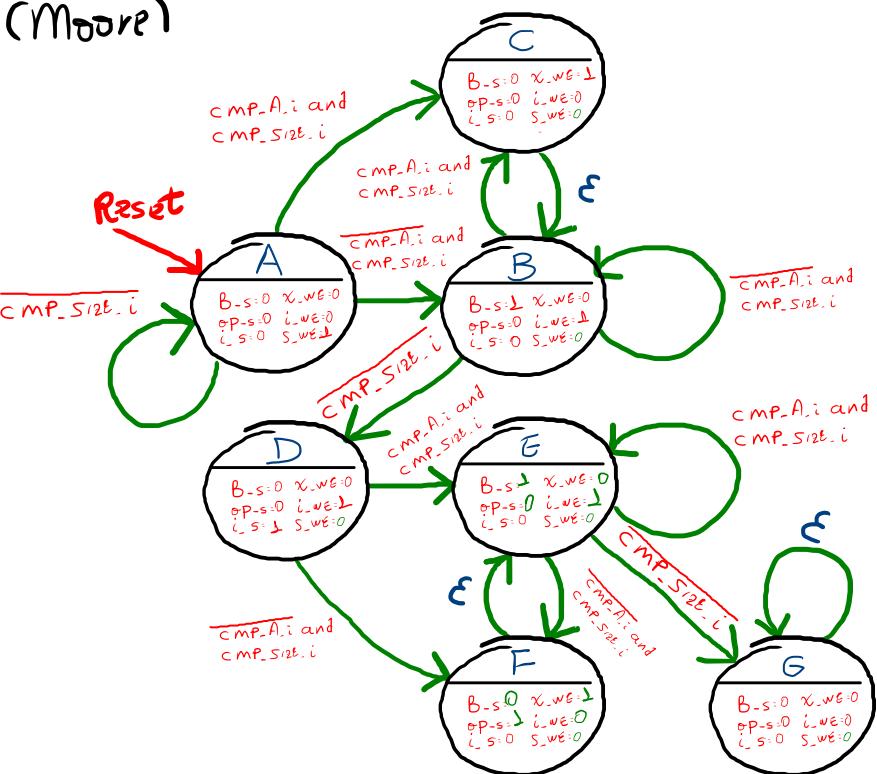
# \* Data path

## Sizes



# \* FSM da U.C.

(Moore)



States	Desc	Logic	cmp- $A_i$
A	init	$A[i] > i$ , $\neg(A[i] \leq i)$	1
B	$i++$	$A[i] \leq i$ , $\neg(A[i] > i)$	0
C	$x++$		
D	$i=0$		
E	$i++$		
F	$x--$		
G	end		

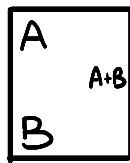
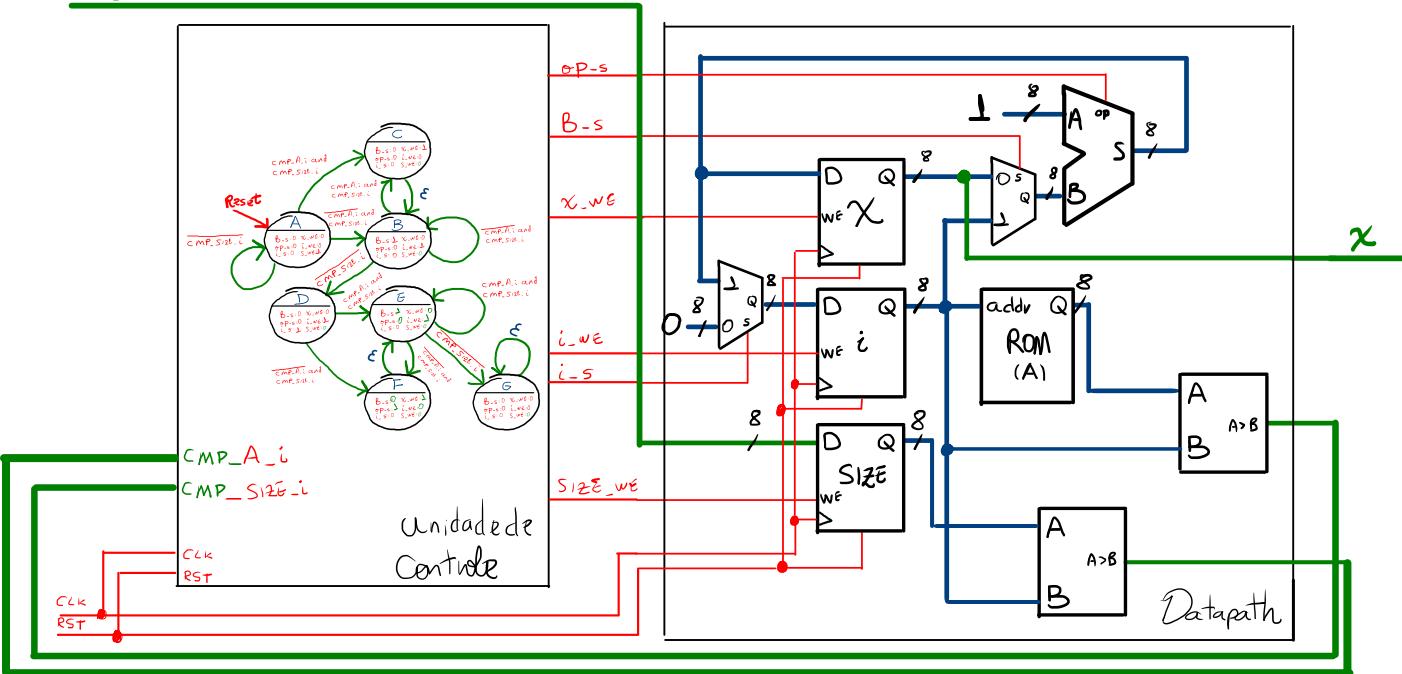
INPUT	OUTPUT
$cmp\_A_i$	$op\_s$
$cmp\_size\_i$	$B\_s$
	$x\_we$
	$i\_we$
	$i\_s$
	$s\_we$ ( $size\_we$ )

op-s	operation
0	$\oplus$
1	$\ominus$

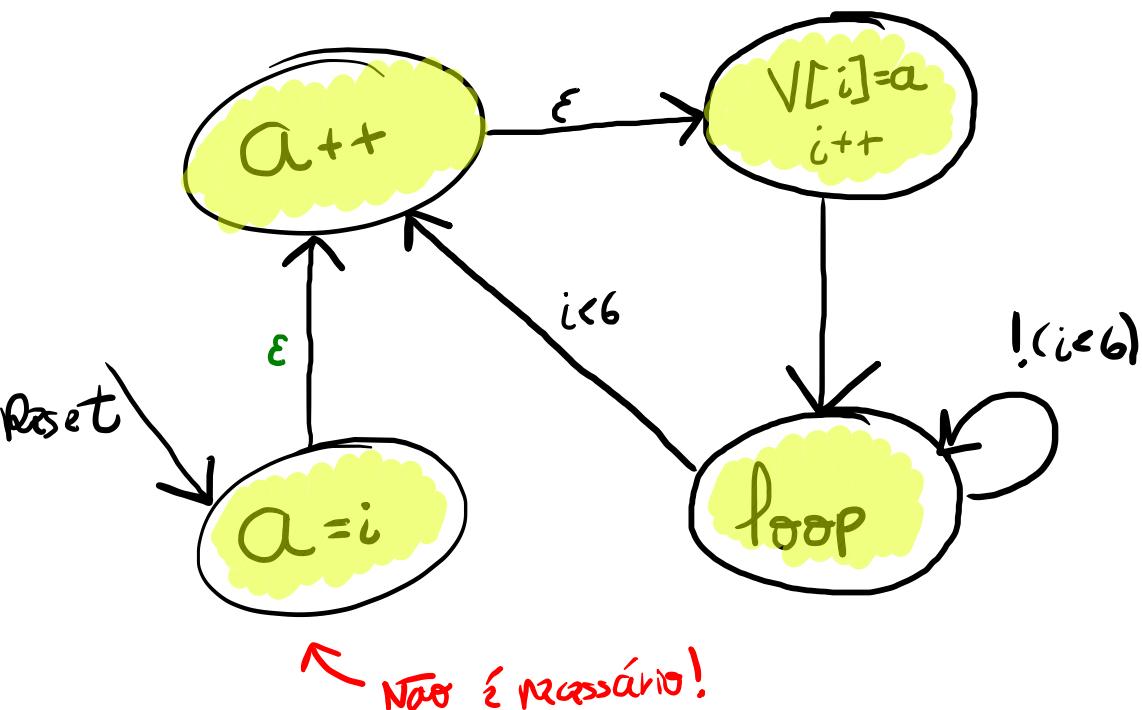
$B\_s$	$B$	$i\_s$	$i\_in$
0	$x$	0	0
1	$i$	1	$i+1$

# Projeto RTh

SIZE



# \* FSM em Alto Nível



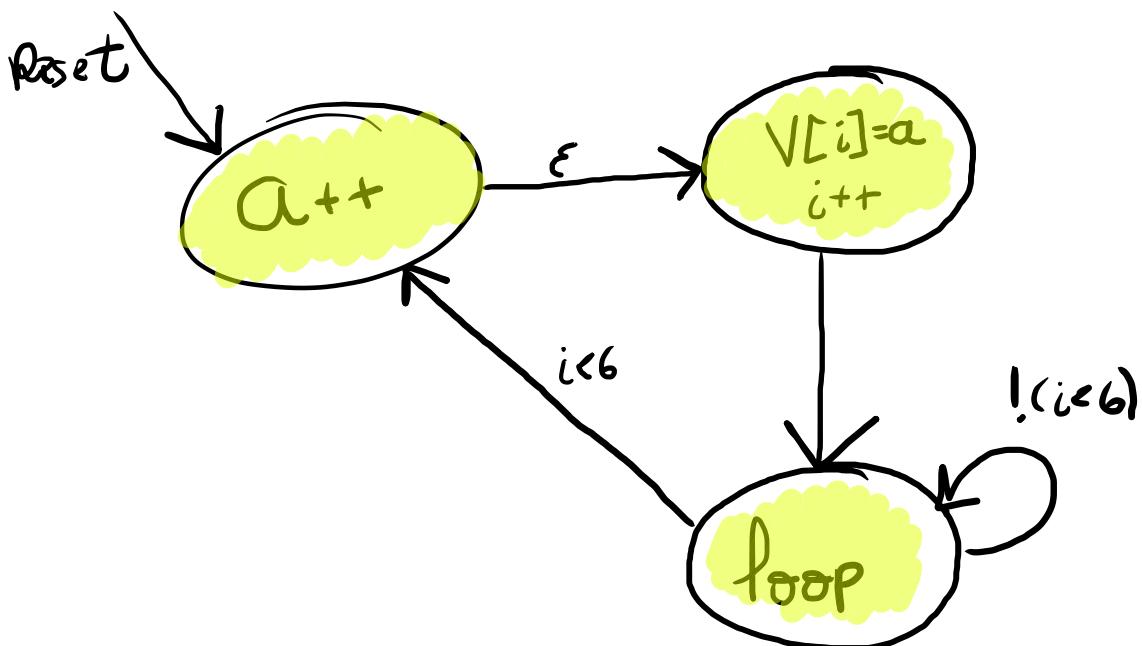
```

b) i = 0;
   a = i; ← a=0
3 while ( i < 6 ){
4     a += 1;
5     v[ i++ ] = a;
6 }
```

OBS: a poderia ser o mesmo que o .

→ Variáveis:  
 $i(8)$   
 $a(8)$   
 $V[256](8)$

# \* FSM em Alto Nível



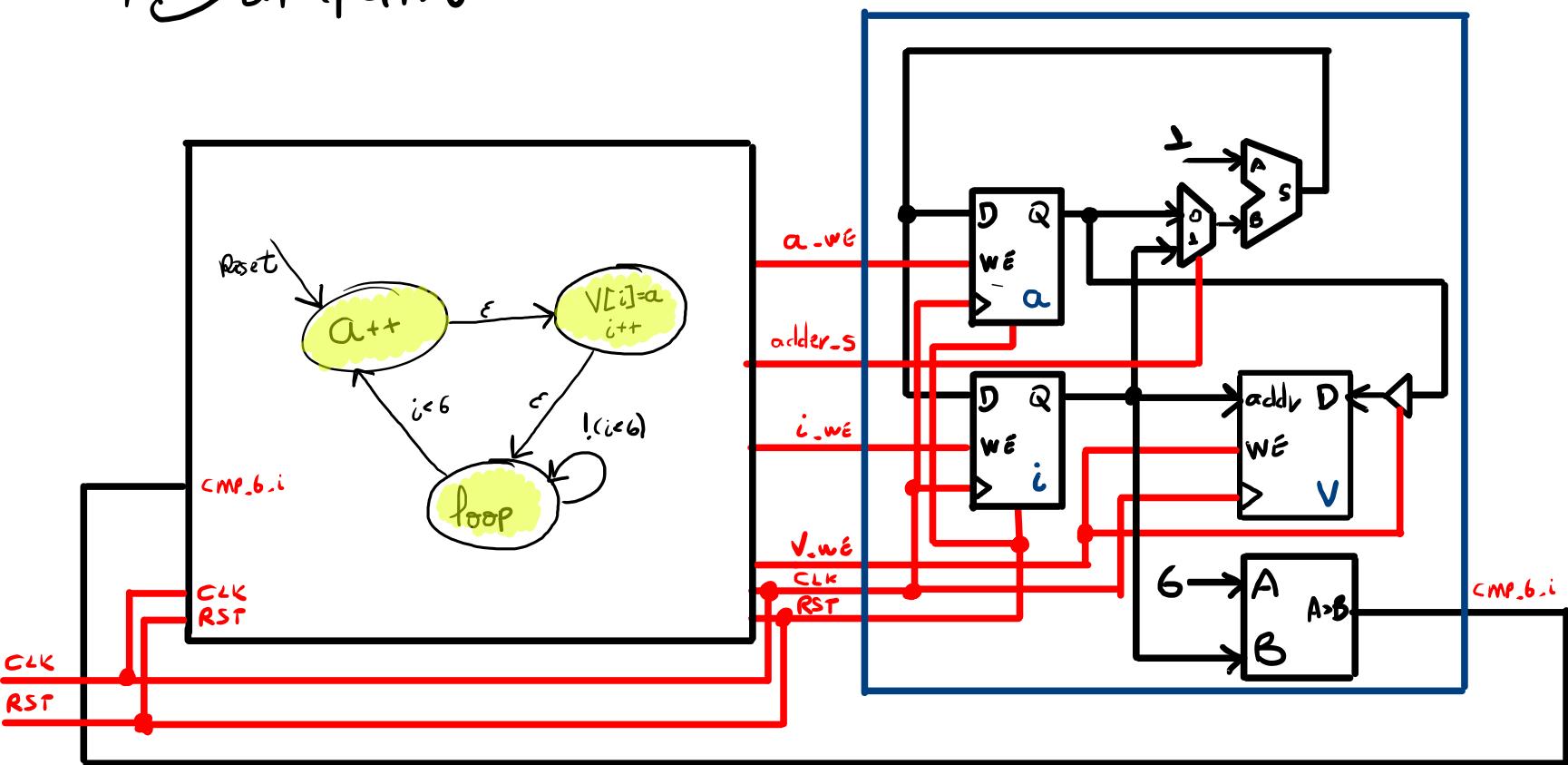
```

b) i = 0;
2 a = i; ← a=0
3 while ( i < 6 ){
4     a += 1;
5     v[ i++ ] = a;
6 }
  
```

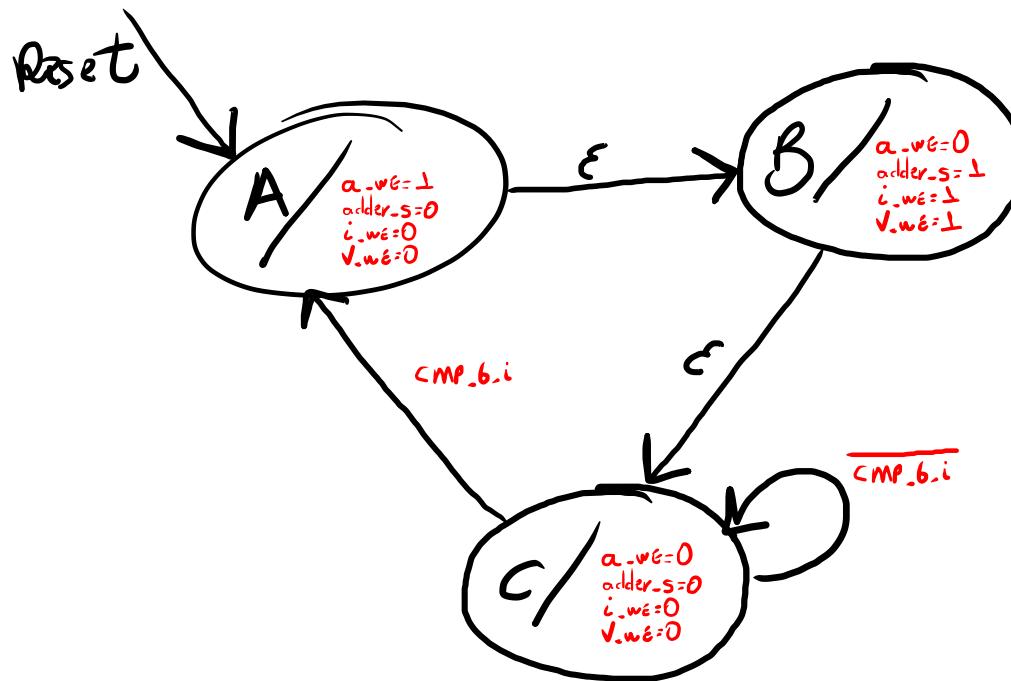
OBS: a poderia ser o mesmo que:

→ Variáveis:  
 $i(8)$   
 $a(8)$   
 $V[256](8)$

# \* Datapath



# \* FSM da U.C (Moore)



State	Desc.
A	$a++$
B	$V[i]=a$ $i++$
C	loop

# \* Projeto RTH

