



**Universidade Federal de Santa Catarina**  
**Centro Tecnológico**  
Departamento de Informática e Estatística  
**Ciências da Computação & Engenharia Eletrônica**



# **Sistemas Digitais**

**INE 5406**

## **Aula 3-T**

### **1. Componentes do Nível RT: Máquinas Sequenciais Síncronas. Processamento Multiciclo e o modelo BO/BC**

**Profs. José Luís Güntzel e Cristina Meinhardt**  
{j.guntzel, cristina.meinhardt}@ufsc.br

# Máquinas Sequenciais Síncronas

---

Suponha que necessita-se de um hardware dedicado para realizar o seguinte cálculo:

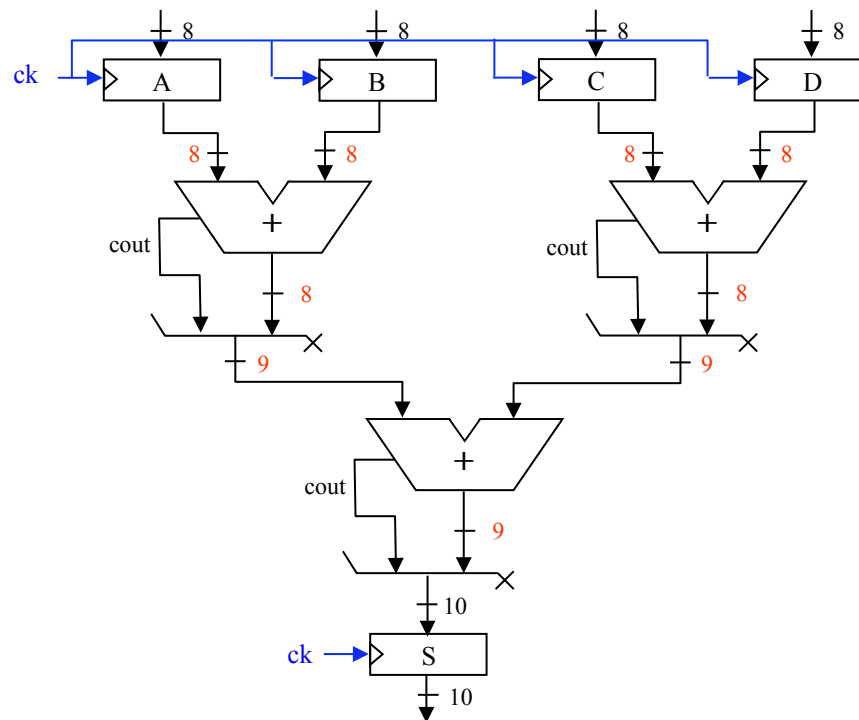
$$S \leftarrow A + B + C + D$$

Assumir que as variáveis A, B, C e D sejam números inteiros sem sinal, representados em binário com 8 bits, **estando disponíveis ao mesmo tempo.**

- Como as variáveis de entrada estão disponíveis ao mesmo tempo, pode ser vantajoso executar este cálculo em um ciclo de relógio.
- Para reduzir o período do relógio, é importante explorar o paralelismo que o cálculo pode proporcionar.

# Máquinas Sequenciais Síncronas

## Solução Monociclo (que evita ocorrência de *overflow* e tem custo mínimo)



### Pontos Positivos:

- Jamais ocorre overflow.
- Este esquemático deixa claro como é gerado o bit mais significativo da saída dos somadores.
- Somadores dimensionados de modo a economizar recursos (2 somadores p/ números com 8 bits e 1 somador para números de 9 bits) e minimizar o atraso crítico!

# Máquinas Sequenciais Síncronas

---

Suponha que necessita-se de um hardware dedicado para realizar o seguinte cálculo:

$$S \leftarrow A + B + C + D$$

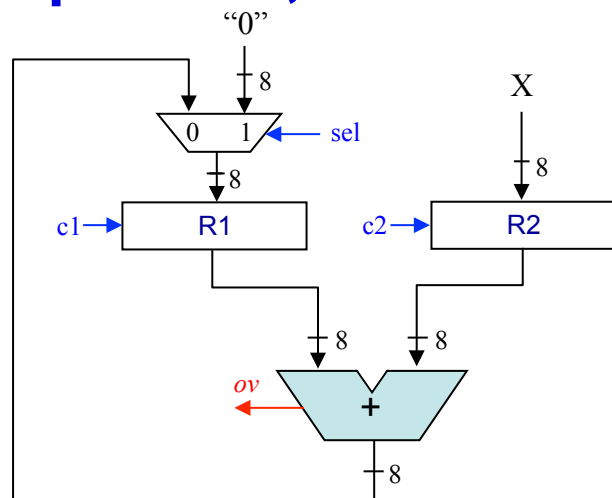
Assumir que as variáveis A, B, C e D sejam números inteiros sem sinal, representados em binário com 8 bits, **sendo fornecidas uma após a outra, em ciclos de relógio consecutivos,**

- Como as variáveis de entrada são fornecidas uma após a outra (i.e., de maneira sequencial), não há vantagem em realizar este cálculo em um ciclo de relógio, pois seria necessário esperar a chegada de todas as variáveis.
- Então, projeta-se um bloco operativo para execução sequencial, buscando minimizar o custo.

# Máquinas Sequenciais Síncronas

## Solução Multiciclo 1:

Processador dedicado que realiza o cálculo de maneira sequencial, e detecta *overflow*



$$S \leftarrow A + B + C + D$$

Execução sequencial

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Neste esquemático RT, o relógio está omitido...

- Cada linha é executada em um ciclo de relógio
- As operações em uma mesma linha são feitas em paralelo
- Um bloco de controle (omitido) gera os sinais “sel”, “c1” e “c2” para viabilizar a operação sequencial (multiciclo)
- Ao término da execução, o resultado estará em R1 (logo, R1 faz o papel de S)

# Máquinas Sequenciais Síncronas

## Solução Multiciclo 1:

### Análise do Comportamento por meio de exemplo numérico

```
1. R1 ← 0; R2 ← X; // A deve estar estável na entradaX
2. R1 ← R1 + R2; R2 ← X; // B deve estar estável na entradaX
3. R1 ← R1 + R2; R2 ← X; // C deve estar estável na entradaX
4. R1 ← R1 + R2; R2 ← X; // D deve estar estável na entradaX
5. R1 ← R1 + R2; // O resultado final S estará em R1
```

- A, B, C, D (e R1 e R2) possuem 8 bits
- O intervalo de representação de inteiros sem sinal com 8 bits é [ 0, 255 ]
- **Suponha que A=40, B=30, C=20, D=10.** Pergunta: ocorre *overflow* em alguma das 4 somas da execução sequencial?

# Máquinas Sequenciais Síncronas

## Solução Multiciclo 1:

### Análise do Comportamento por meio de exemplo numérico

```
1. R1 ← 0; R2 ← X; // A deve estar estável na entradaX
2. R1 ← R1 + R2; R2 ← X; // B deve estar estável na entradaX
3. R1 ← R1 + R2; R2 ← X; // C deve estar estável na entradaX
4. R1 ← R1 + R2; R2 ← X; // D deve estar estável na entradaX
5. R1 ← R1 + R2; // O resultado final S estará em R1
```

- A, B, C, D (e R1 e R2) possuem 8 bits
- O intervalo de representação de inteiros sem sinal com 8 bits é [ 0, 255 ]
- **Suponha que A=40, B=30, C=20, D=10.** Pergunta: ocorre *overflow* em alguma das 4 somas da execução sequencial?
- Simulação da execução:

passo	R1	R2
1.	0	40

# Máquinas Sequenciais Síncronas

## Solução Multiciclo 1:

### Análise do Comportamento por meio de exemplo numérico

```
1. R1 ← 0; R2 ← X; // A deve estar estável na entradaX
2. R1 ← R1 + R2; R2 ← X; // B deve estar estável na entradaX
3. R1 ← R1 + R2; R2 ← X; // C deve estar estável na entradaX
4. R1 ← R1 + R2; R2 ← X; // D deve estar estável na entradaX
5. R1 ← R1 + R2; // O resultado final S estará em R1
```

- A, B, C, D (e R1 e R2) possuem 8 bits
- O intervalo de representação de inteiros sem sinal com 8 bits é [ 0, 255 ]
- **Suponha que A=40, B=30, C=20, D=10.** Pergunta: ocorre *overflow* em alguma das 4 somas da execução sequencial?
- Simulação da execução:

passo	R1	R2
1.	0	40
2.	40	30



# Máquinas Sequenciais Síncronas

## Solução Multiciclo 1:

### Análise do Comportamento por meio de exemplo numérico

```
1. R1 ← 0; R2 ← X; // A deve estar estável na entradaX
2. R1 ← R1 + R2; R2 ← X; // B deve estar estável na entradaX
3. R1 ← R1 + R2; R2 ← X; // C deve estar estável na entradaX
4. R1 ← R1 + R2; R2 ← X; // D deve estar estável na entradaX
5. R1 ← R1 + R2; // O resultado final S estará em R1
```

- A, B, C, D (e R1 e R2) possuem 8 bits
- O intervalo de representação de inteiros sem sinal com 8 bits é [ 0, 255 ]
- **Suponha que A=40, B=30, C=20, D=10.** Pergunta: ocorre *overflow* em alguma das 4 somas da execução sequencial?
- Simulação da execução:

passo	R1	R2
1.	0	40
2.	40	30
3.	70	20

# Máquinas Sequenciais Síncronas

## Solução Multiciclo 1:

### Análise do Comportamento por meio de exemplo numérico

```
1. R1 ← 0; R2 ← X; // A deve estar estável na entradaX
2. R1 ← R1 + R2; R2 ← X; // B deve estar estável na entradaX
3. R1 ← R1 + R2; R2 ← X; // C deve estar estável na entradaX
4. R1 ← R1 + R2; R2 ← X; // D deve estar estável na entradaX
5. R1 ← R1 + R2; // O resultado final S estará em R1
```

- A, B, C, D (e R1 e R2) possuem 8 bits
- O intervalo de representação de inteiros sem sinal com 8 bits é [ 0, 255 ]
- **Suponha que A=40, B=30, C=20, D=10.** Pergunta: ocorre *overflow* em alguma das 4 somas da execução sequencial?
- Simulação da execução:

passo	R1	R2
1.	0	40
2.	40	30
3.	70	20
4.	90	10

# Máquinas Sequenciais Síncronas

## Solução Multiciclo 1:

### Análise do Comportamento por meio de exemplo numérico

```
1. R1 ← 0; R2 ← X; // A deve estar estável na entradaX
2. R1 ← R1 + R2; R2 ← X; // B deve estar estável na entradaX
3. R1 ← R1 + R2; R2 ← X; // C deve estar estável na entradaX
4. R1 ← R1 + R2; R2 ← X; // D deve estar estável na entradaX
5. R1 ← R1 + R2; // O resultado final S estará em R1
```

- A, B, C, D (e R1 e R2) possuem 8 bits
- O intervalo de representação de inteiros sem sinal com 8 bits é [ 0, 255 ]
- **Suponha que A=40, B=30, C=20, D=10.** Pergunta: ocorre *overflow* em alguma das 4 somas da execução sequencial?
- Simulação da execução:

passo	R1	R2
1.	0	40
2.	40	30
3.	70	20
4.	90	10
5.	<b>100</b>	?

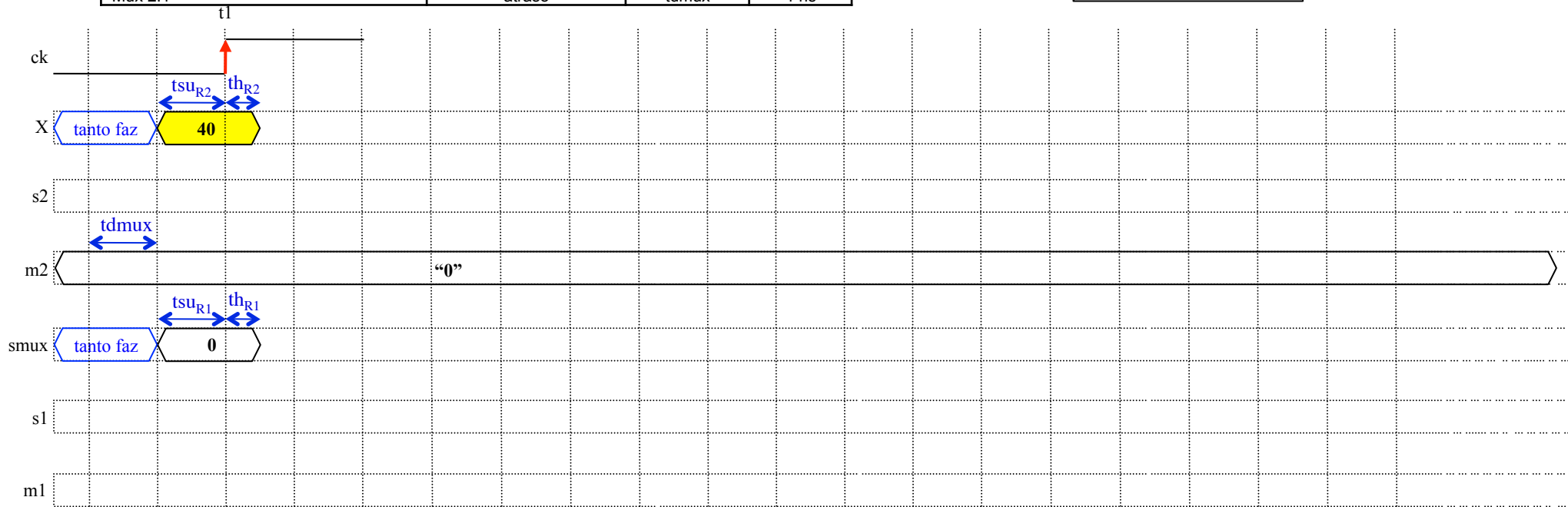
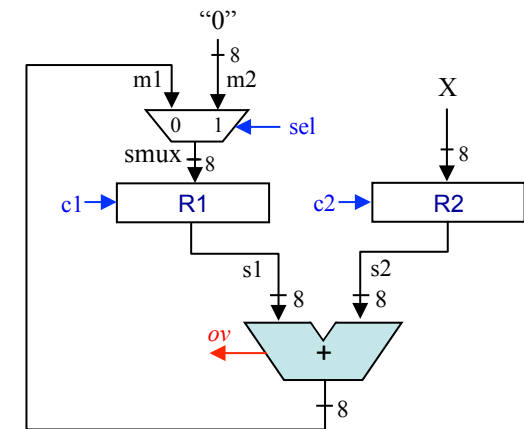
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0



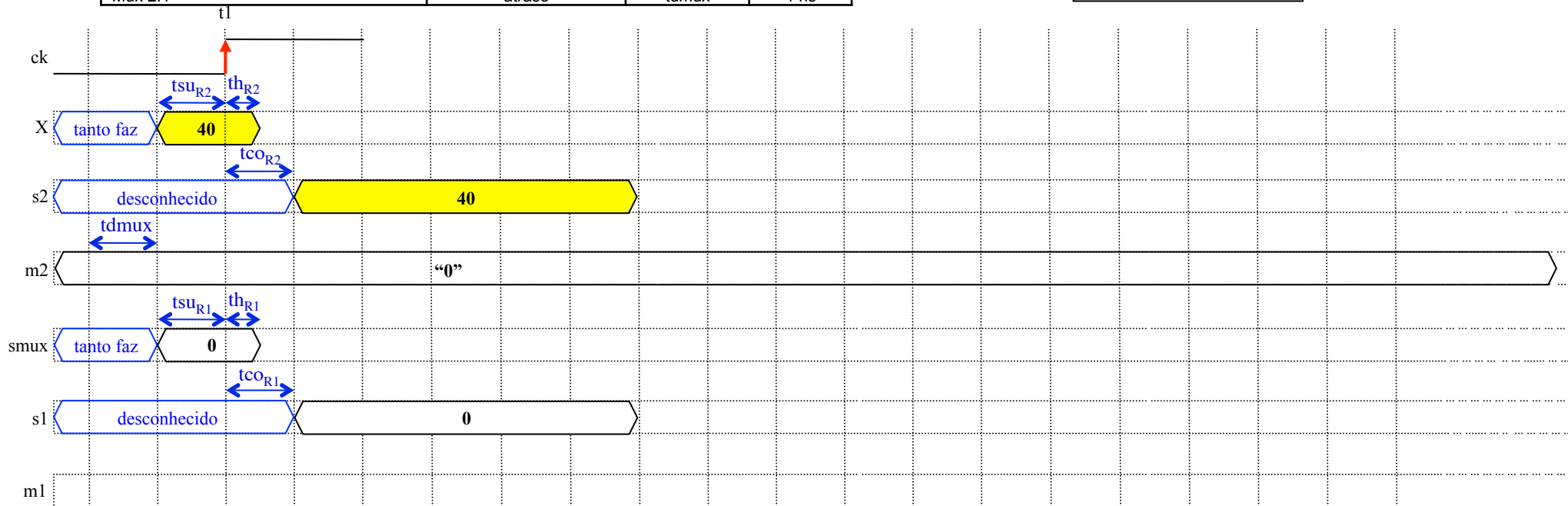
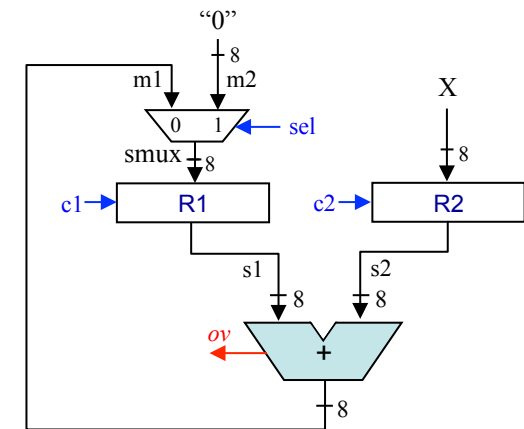
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0



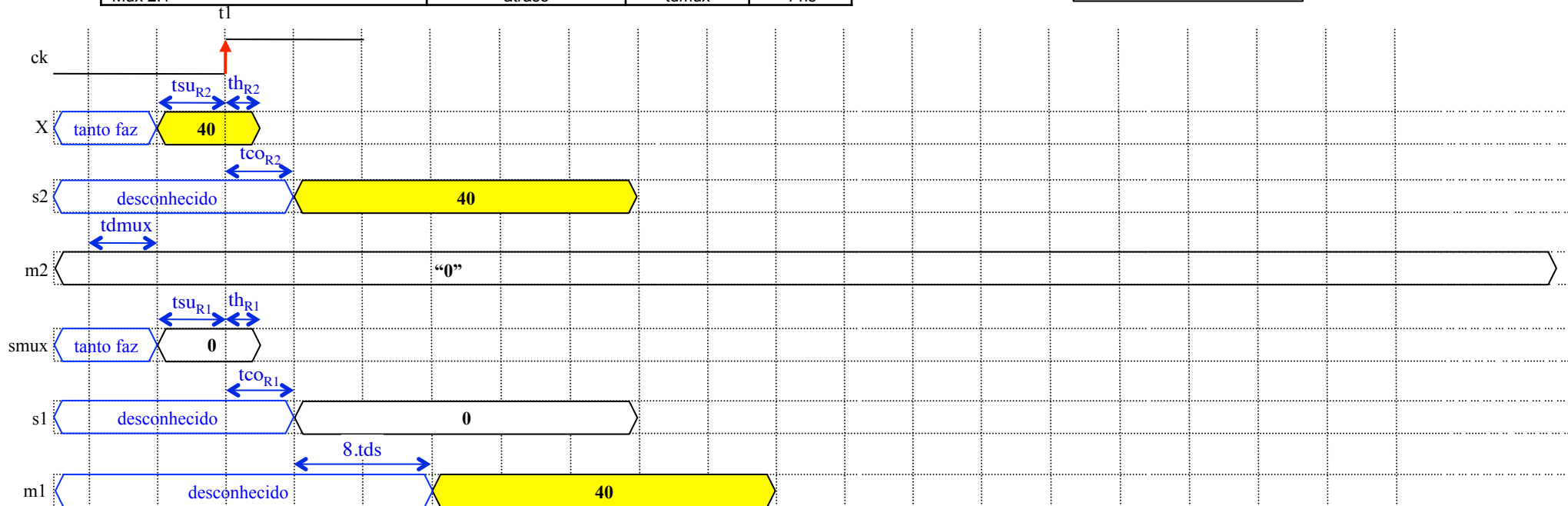
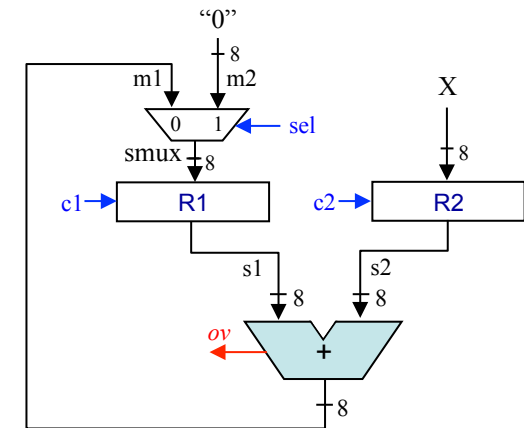
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0



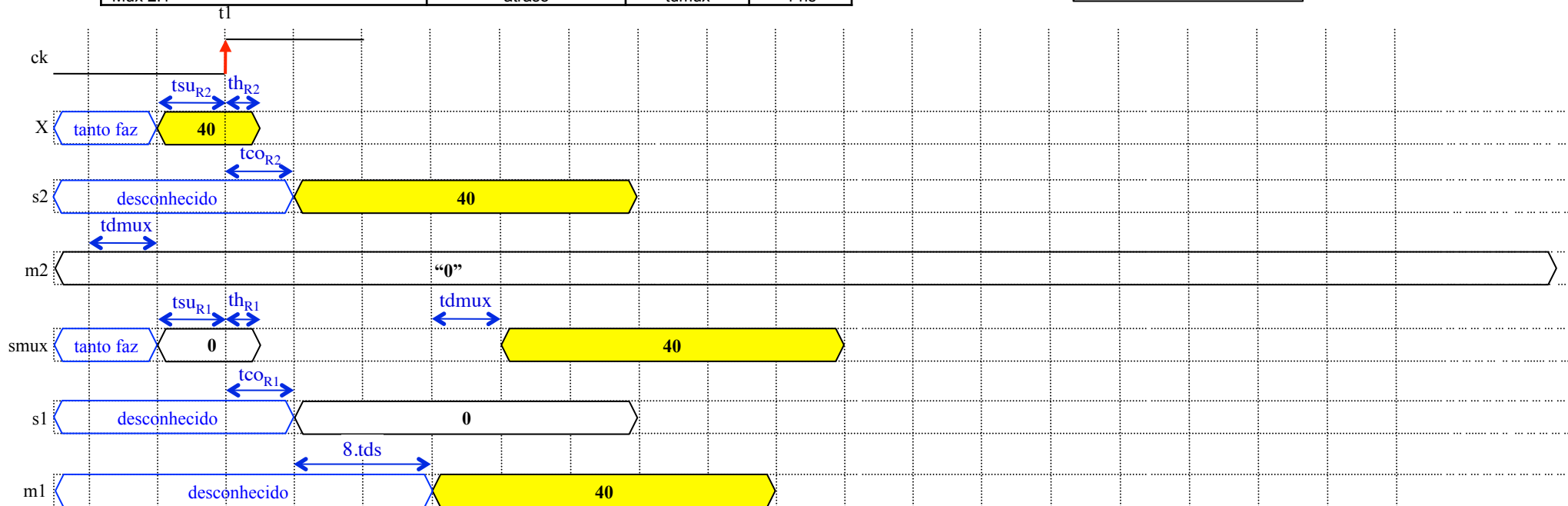
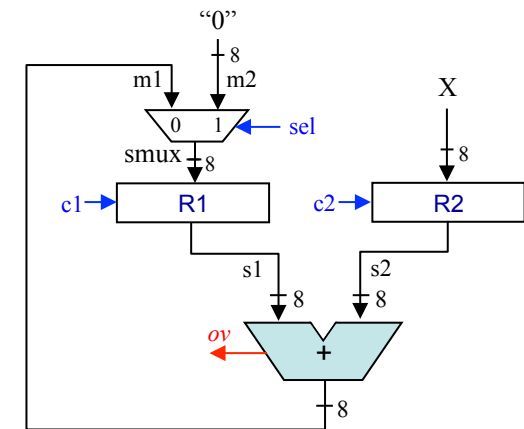
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0



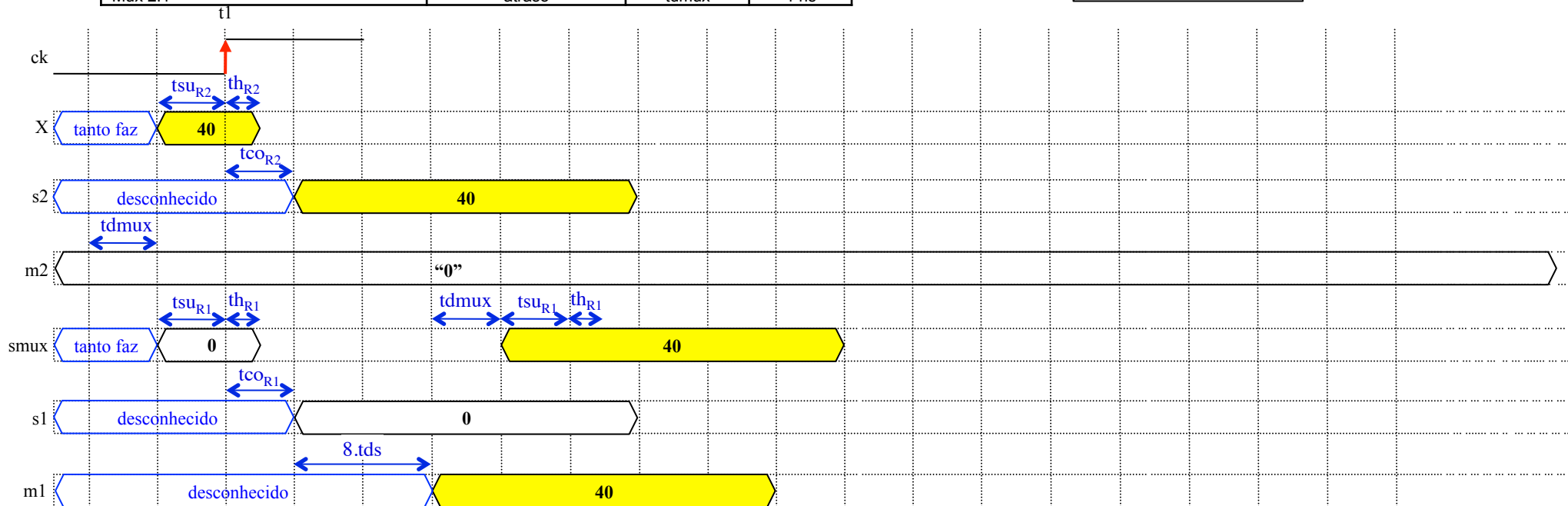
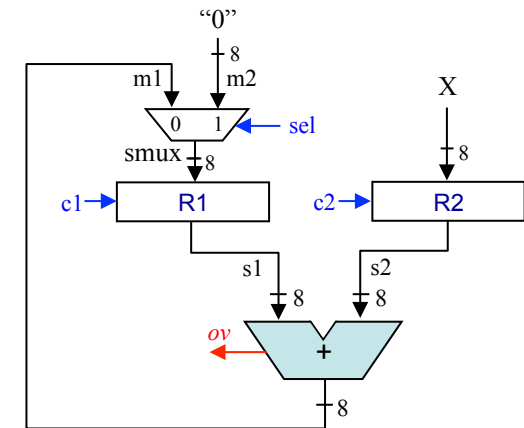
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$tsu$	1 ns
Registadores R1, R2	tempo de hold	$th$	0,5 ns
Registadores R1, R2	tempo de carga	$tco$	1 ns
Somador completo (full adder)	atraso	$tds$	0,25 ns
Mux 2:1	atraso	$tdmux$	1 ns

E supondo que "sel" tenha atraso=0





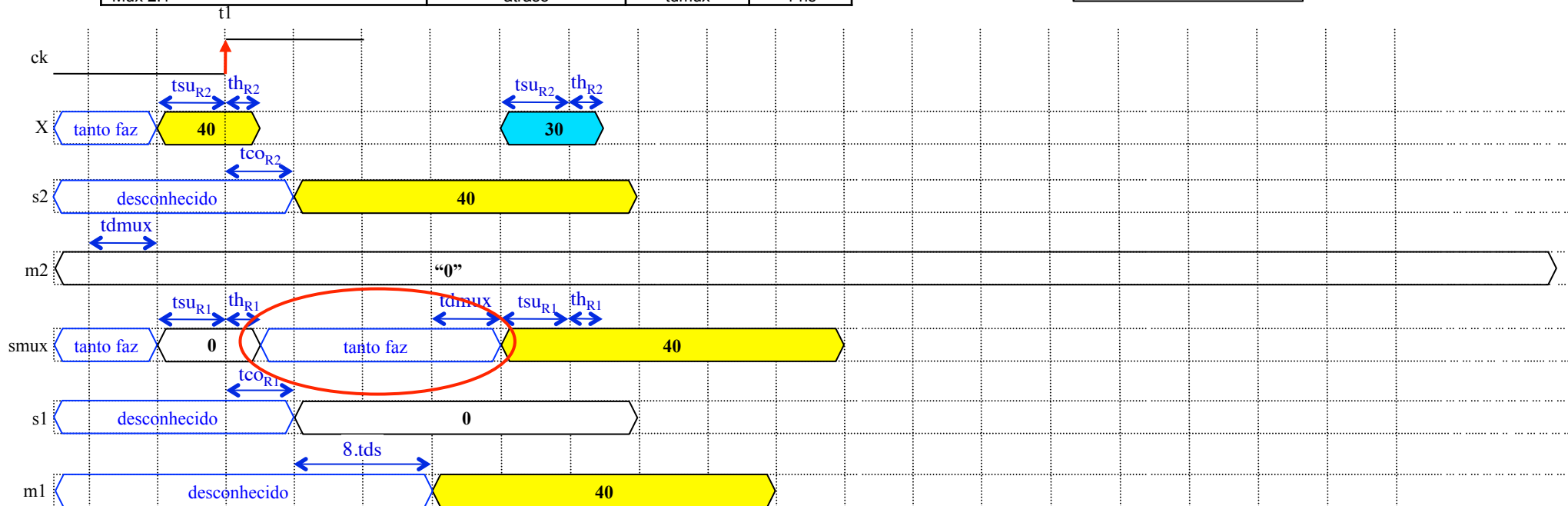
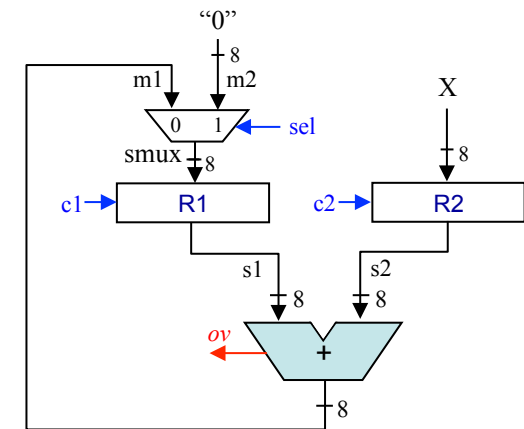
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registradores R1, R2	tempo de setup	tsu	1 ns
Registradores R1, R2	tempo de hold	th	0,5 ns
Registradores R1, R2	tempo de carga	tco	1 ns
Somador completo (full adder)	atraso	tds	0,25 ns
Mux 2:1	atraso	tdmux	1 ns

E supondo que "sel" tenha atraso=0



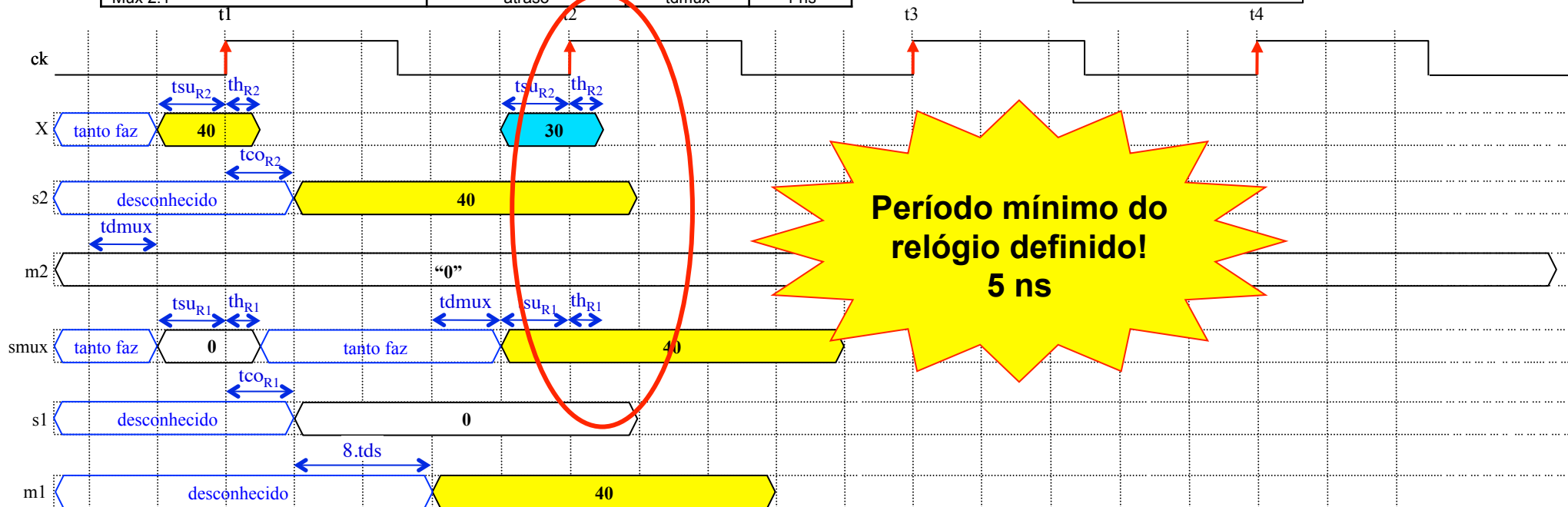
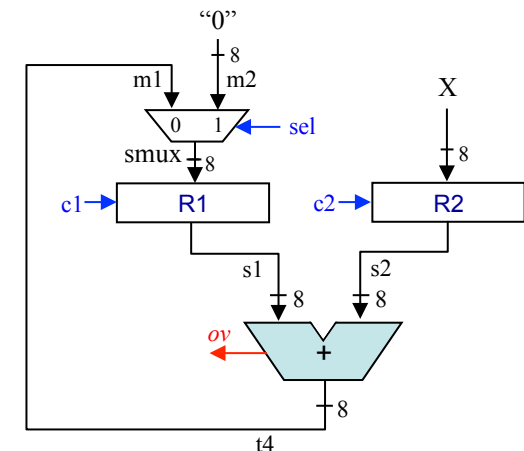
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registradores R1, R2	tempo de setup	$t_{su}$	1 ns
Registradores R1, R2	tempo de hold	$t_h$	0,5 ns
Registradores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0



**Período mínimo do relógio definido! 5 ns**

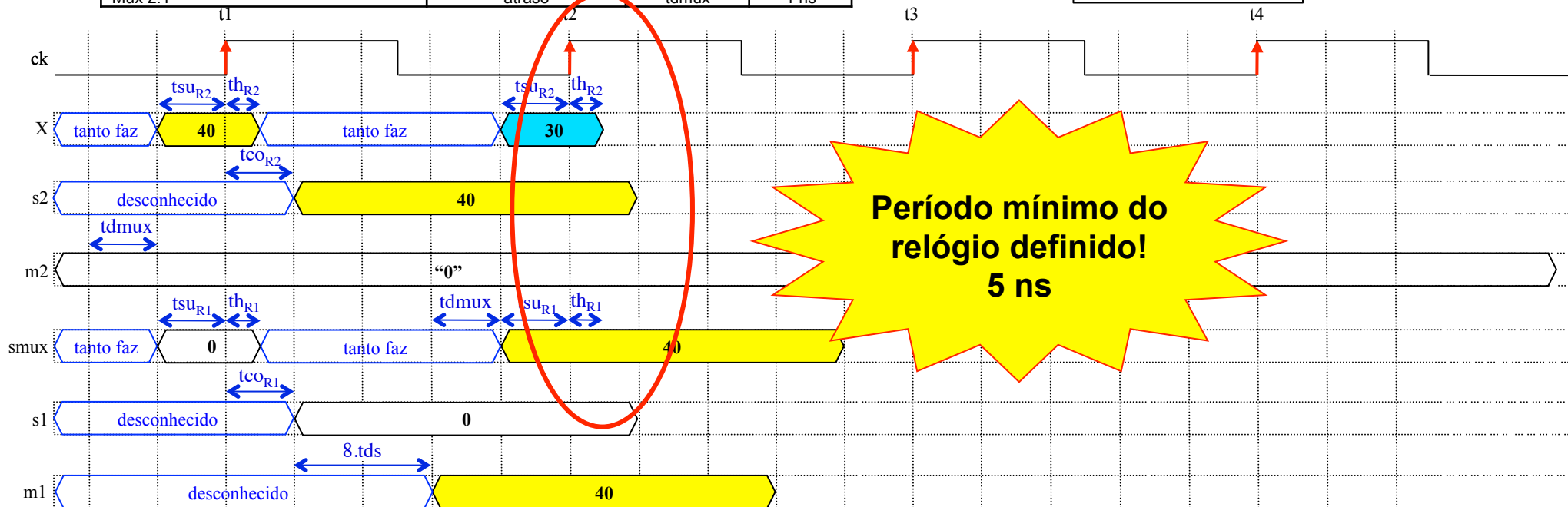
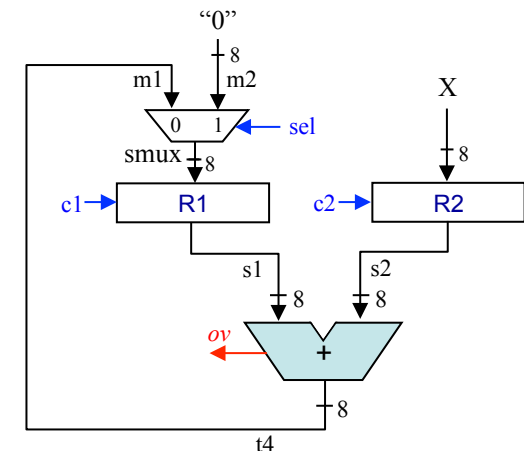
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0



**Período mínimo do relógio definido! 5 ns**

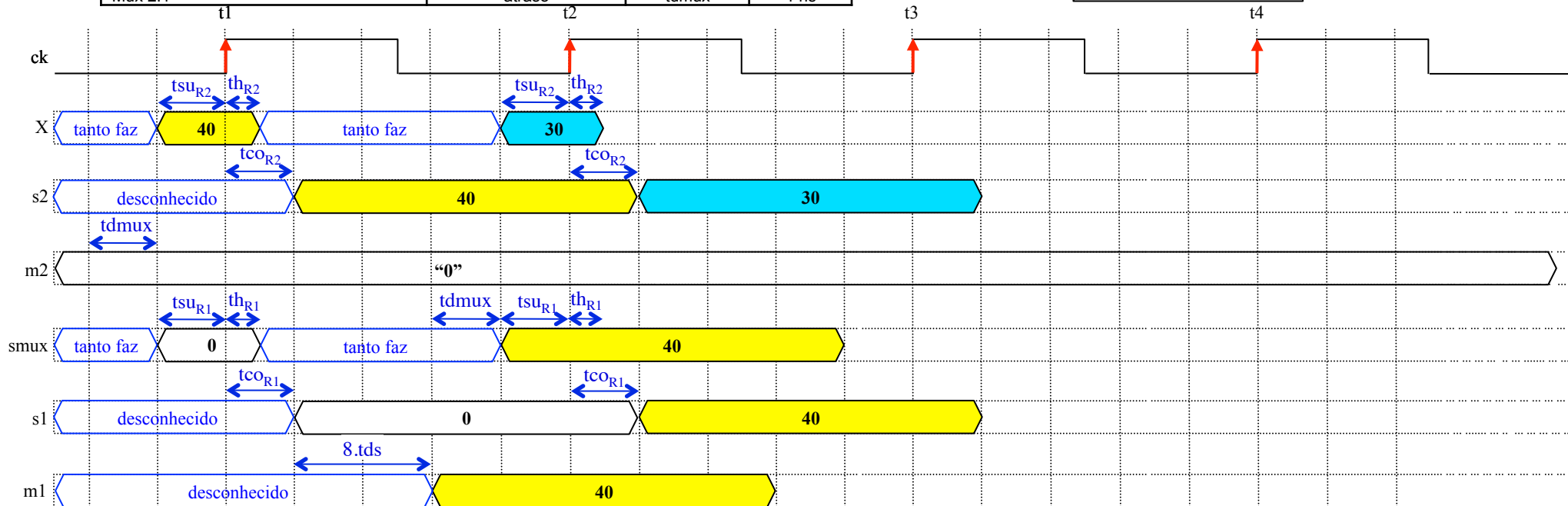
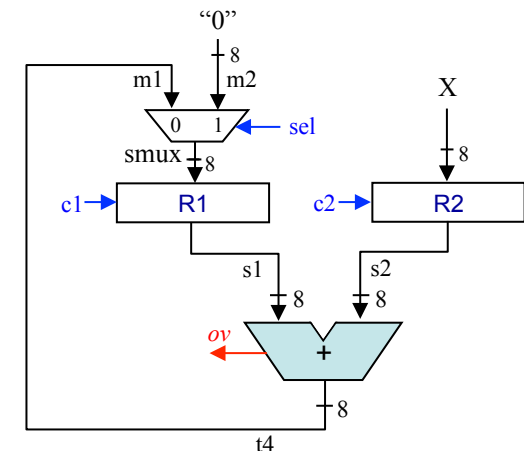
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0



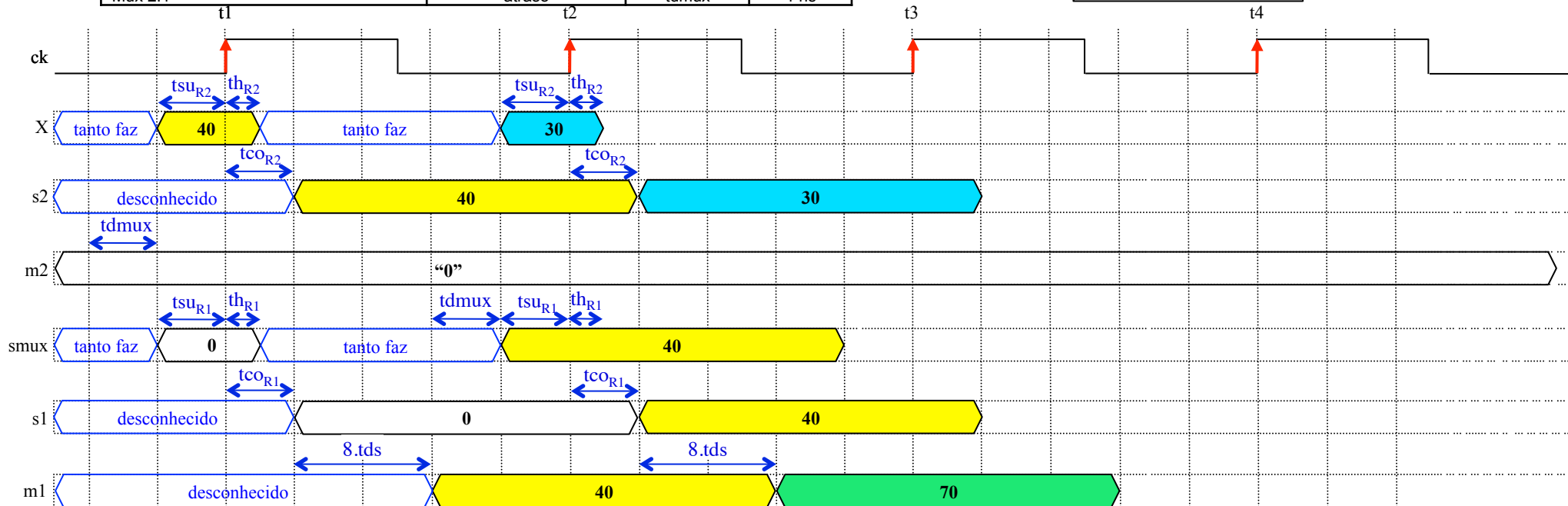
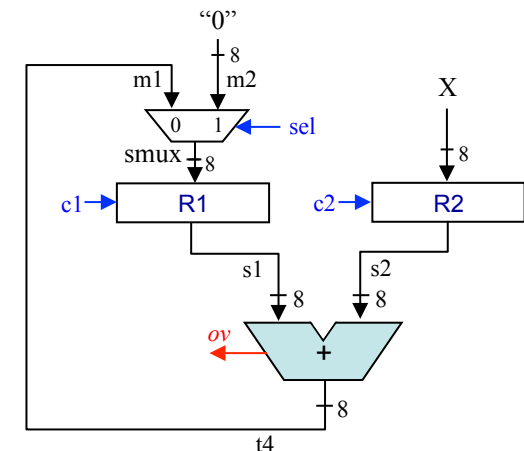
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registradores R1, R2	tempo de setup	$t_{su}$	1 ns
Registradores R1, R2	tempo de hold	$t_h$	0,5 ns
Registradores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0



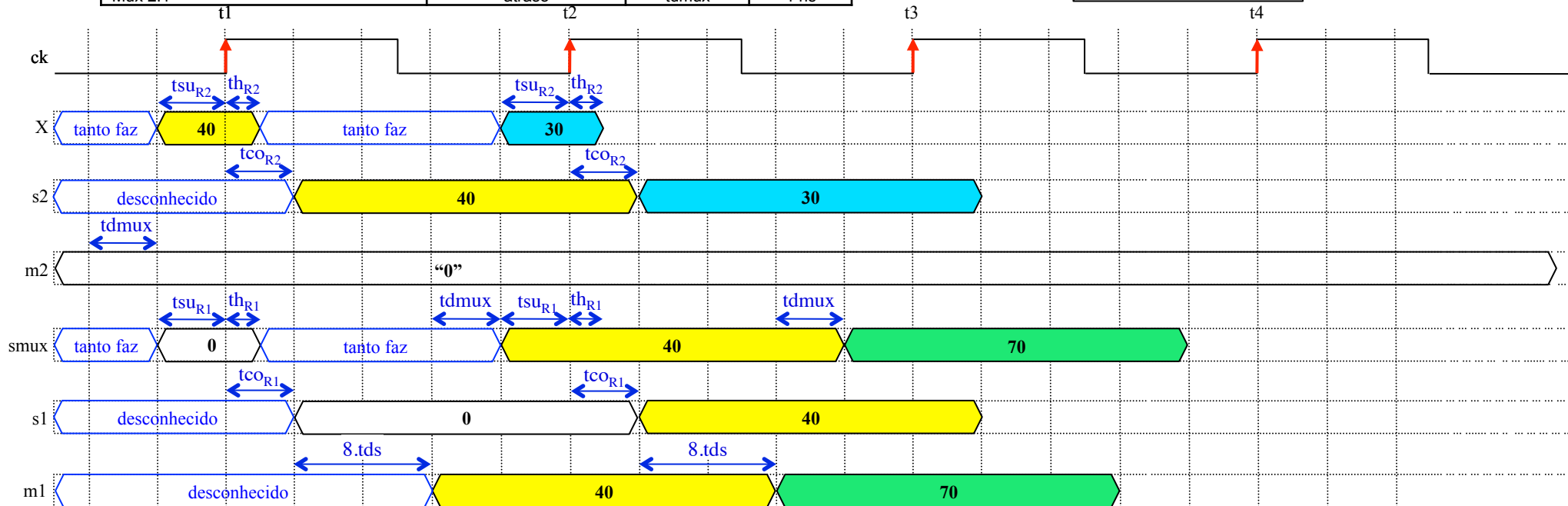
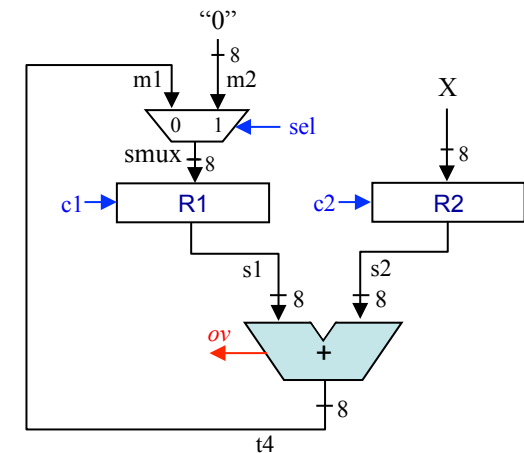
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0



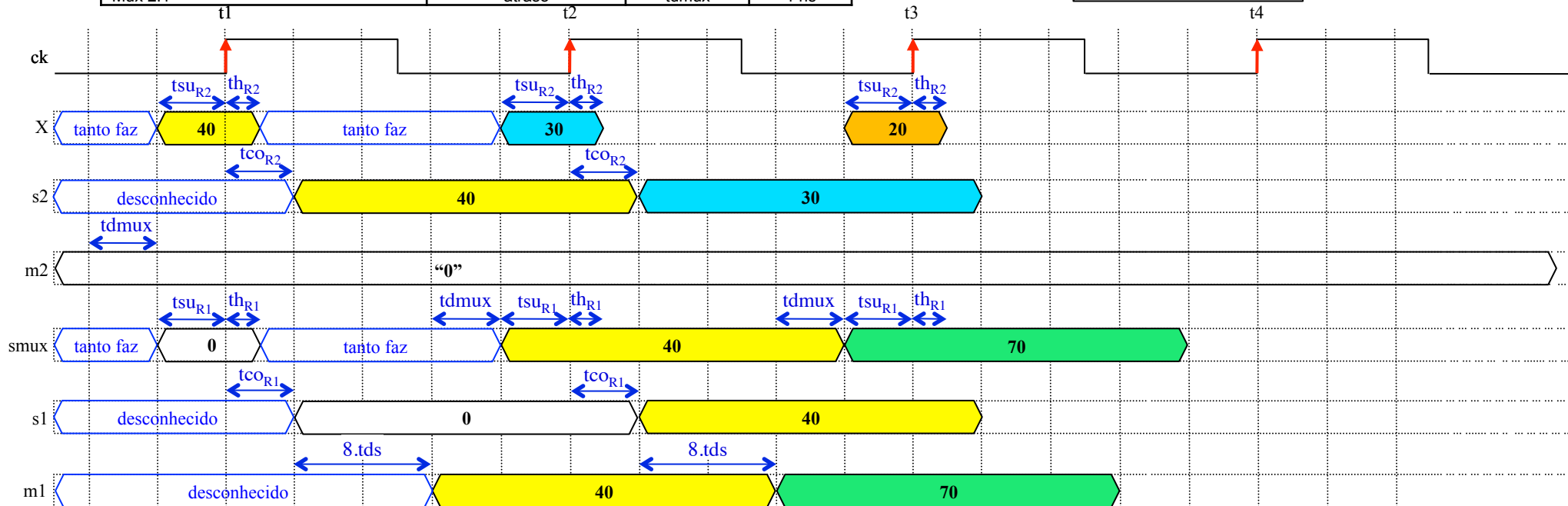
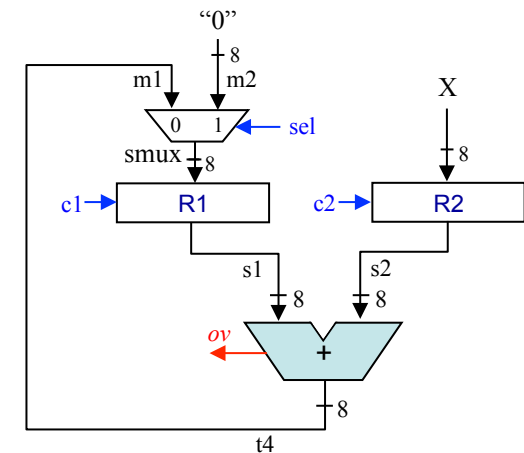
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	tsu	1 ns
Registadores R1, R2	tempo de hold	th	0,5 ns
Registadores R1, R2	tempo de carga	tco	1 ns
Somador completo (full adder)	atraso	tds	0,25 ns
Mux 2:1	atraso	tdmux	1 ns

E supondo que "sel" tenha atraso=0



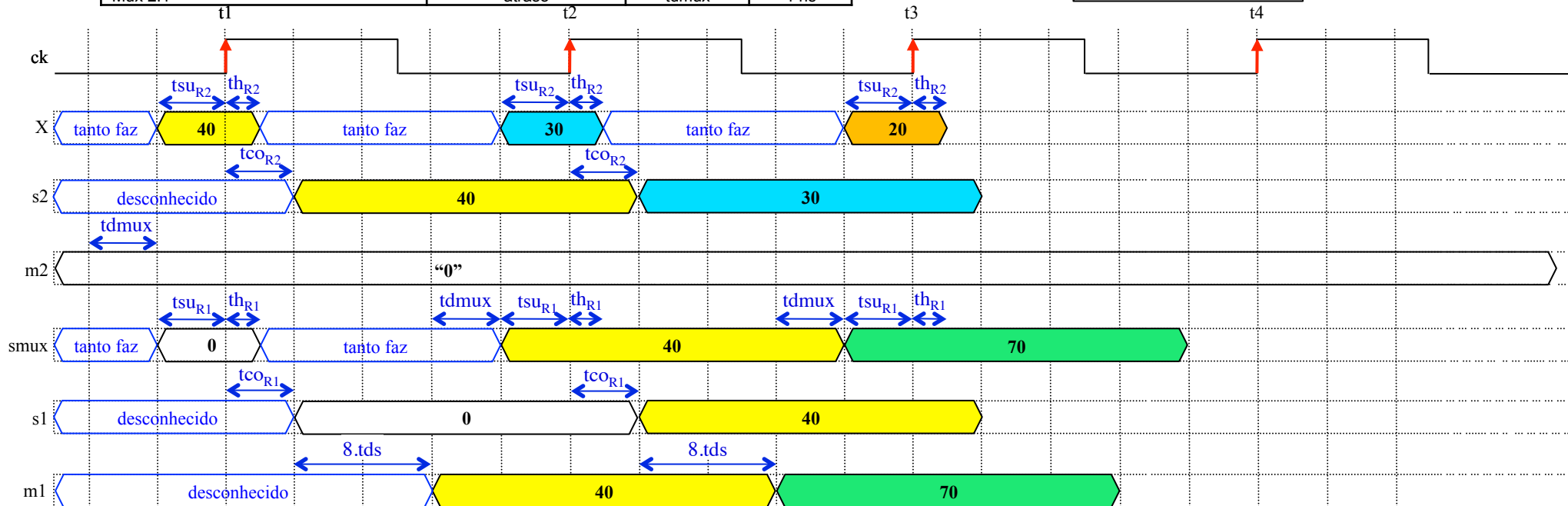
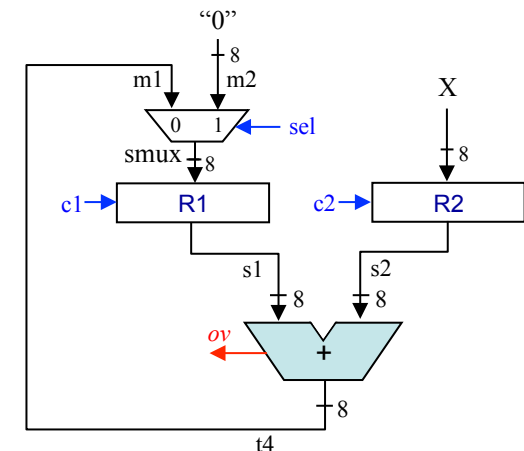
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que "sel" tenha atraso=0





## Solução: Análise Parcial do Funcionamento

- | Componente                             | Característica | Símbolo | Valor   |
|--|----------------|---------|---------|
| Registradores R1, R2                   | tempo de setup | tsu     | 1 ns    |
| Registradores R1, R2                   | tempo de hold  | th      | 0,5 ns  |
| Registradores R1, R2                   | tempo de carga | tco     | 1 ns    |
| Somador completo ( <i>full adder</i> ) | atraso         | tds     | 0,25 ns |
| Mux 2:1                                | atraso         | tdmux   | 1 ns    |



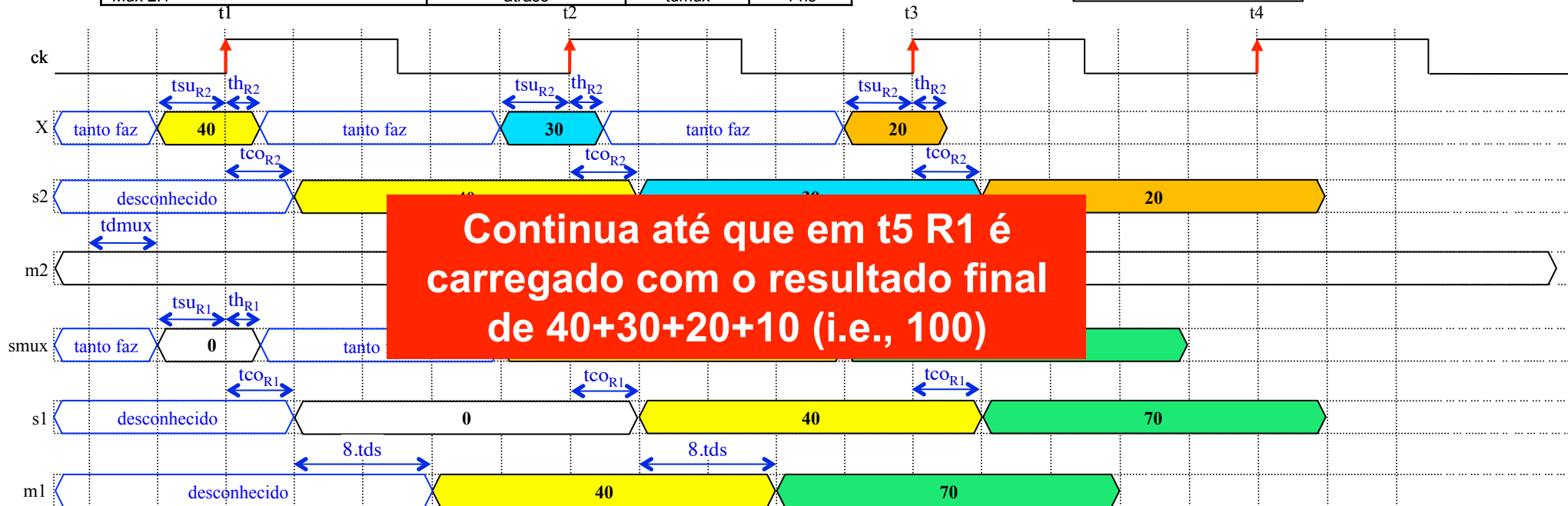
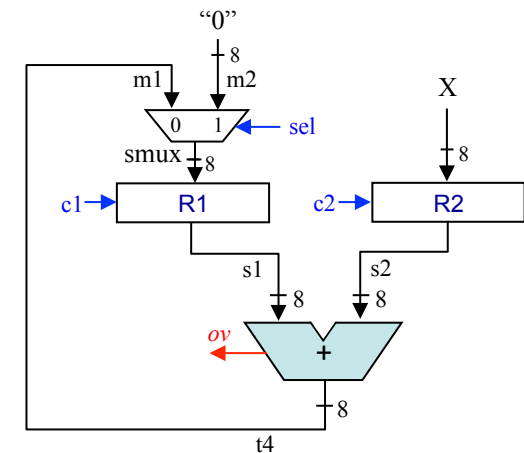
# Máquinas Sequenciais Síncronas

## Solução: Análise Parcial do Funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

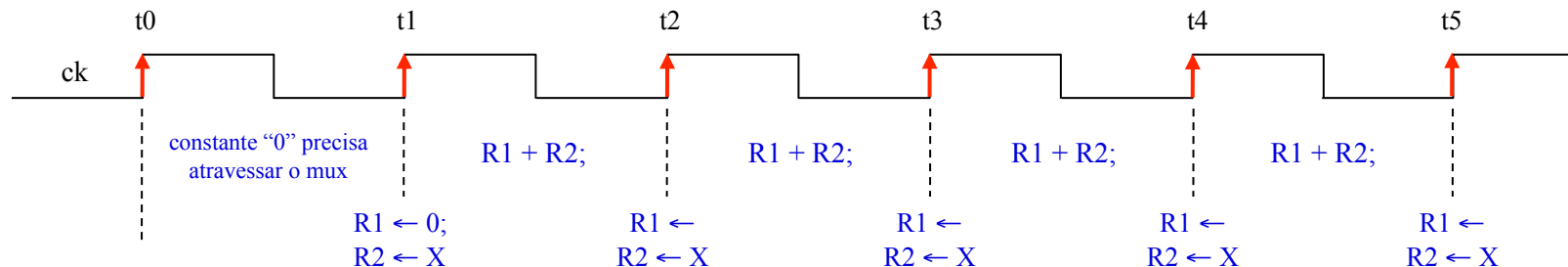
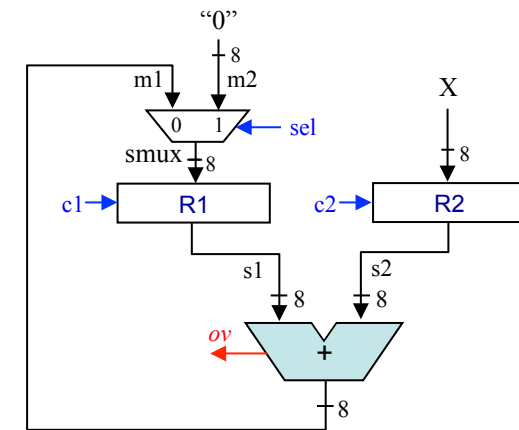
E supondo que "sel" tenha atraso=0



# Máquinas Sequenciais Síncronas

## Solução: Resumo do funcionamento

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1



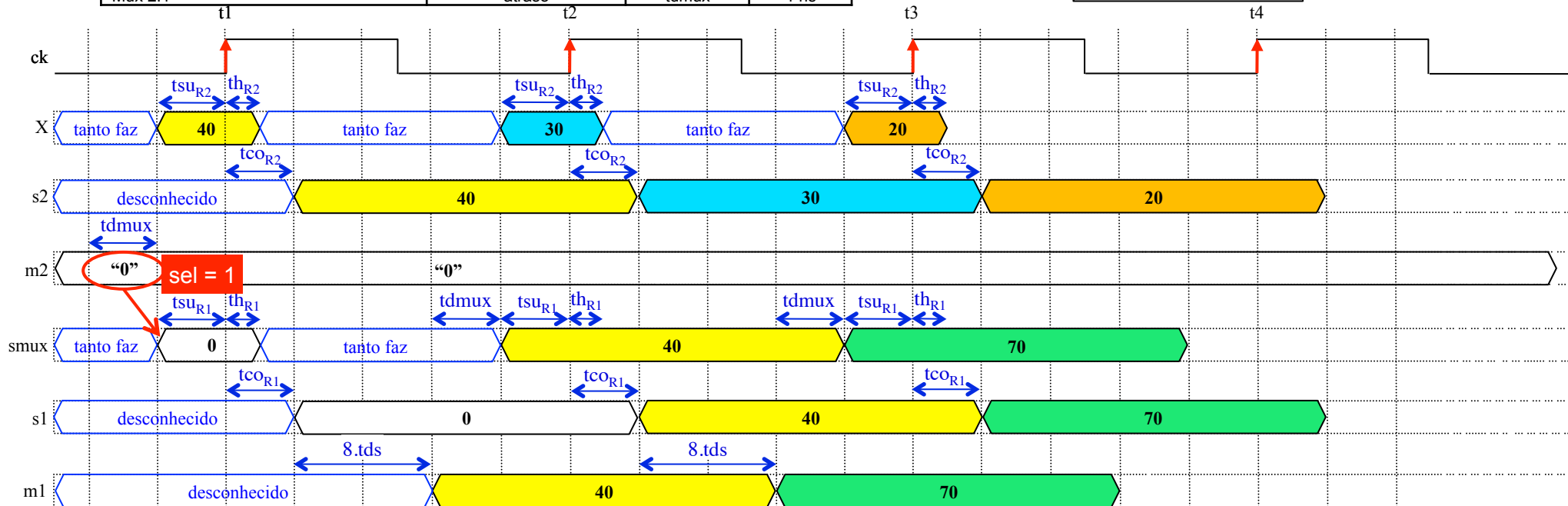
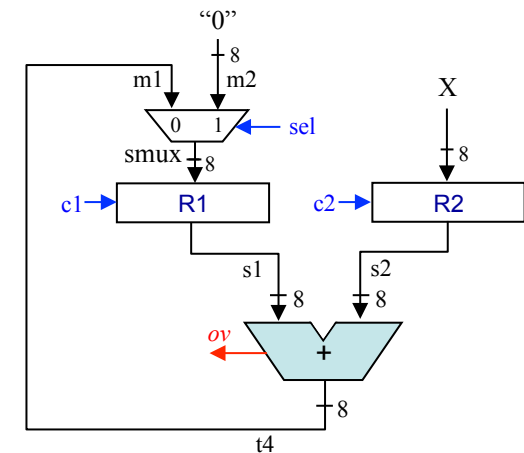
# Máquinas Sequenciais Síncronas

## Solução: Análise do sinal de controle “sel”

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$tsu$	1 ns
Registadores R1, R2	tempo de hold	$th$	0,5 ns
Registadores R1, R2	tempo de carga	$tco$	1 ns
Somador completo (full adder)	atraso	$tds$	0,25 ns
Mux 2:1	atraso	$tdmux$	1 ns

E supondo que “sel” tenha atraso=0



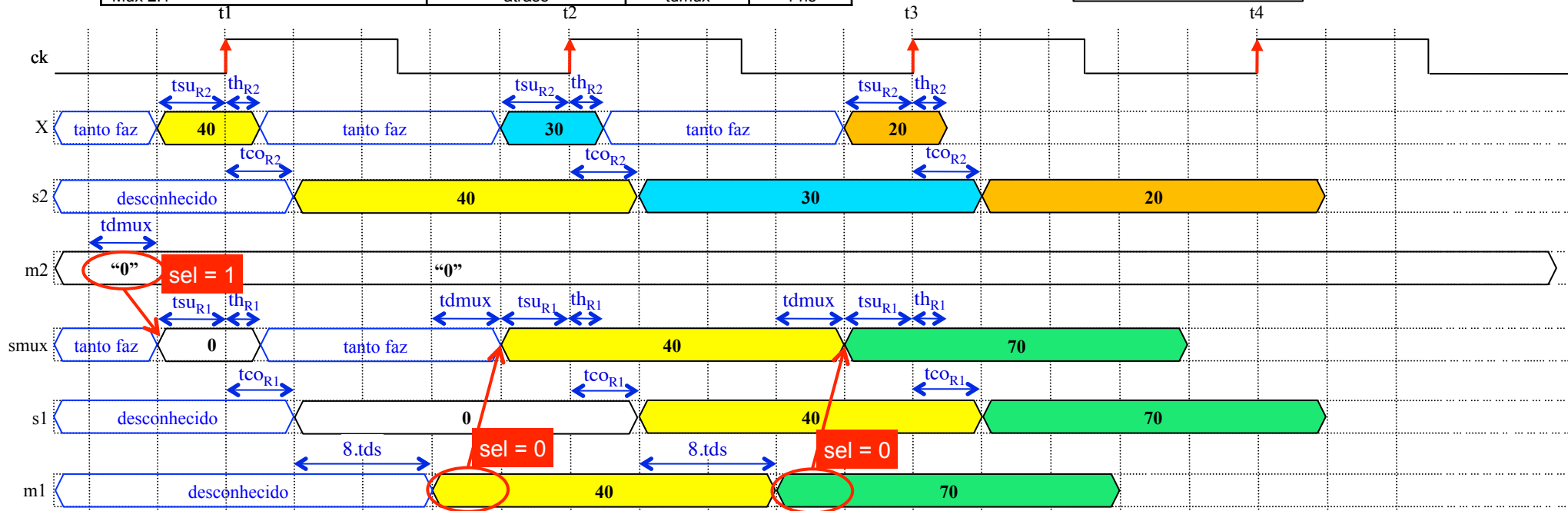
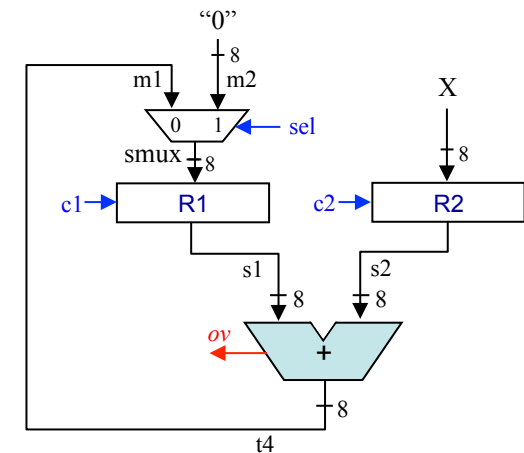
# Máquinas Sequenciais Síncronas

## Solução: Análise do sinal de controle “sel”

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$t_{su}$	1 ns
Registadores R1, R2	tempo de hold	$t_h$	0,5 ns
Registadores R1, R2	tempo de carga	$t_{co}$	1 ns
Somador completo (full adder)	atraso	$t_{ds}$	0,25 ns
Mux 2:1	atraso	$t_{dmux}$	1 ns

E supondo que “sel” tenha atraso=0



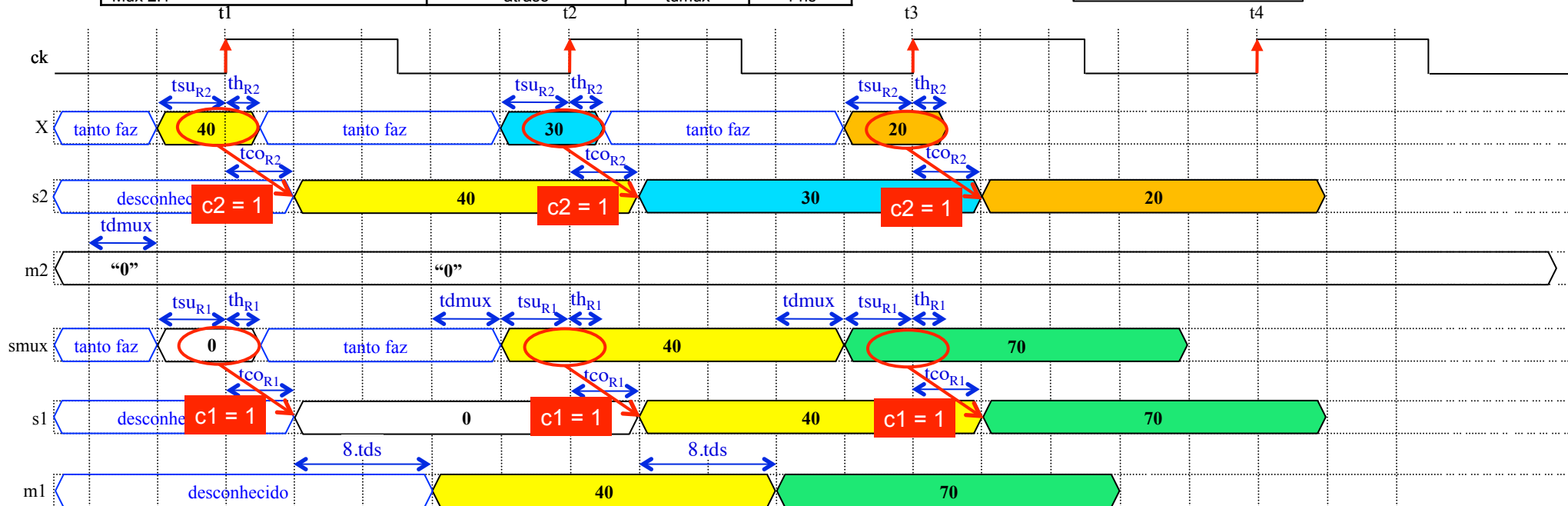
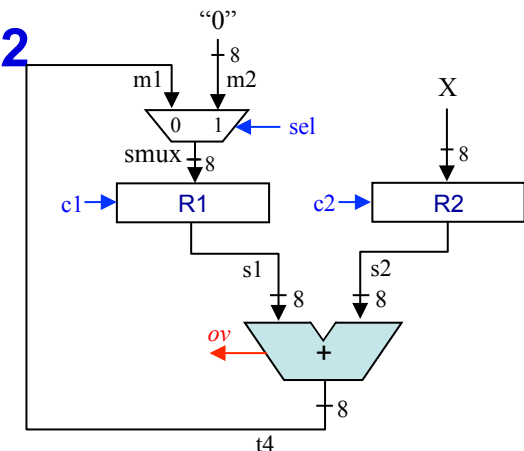
# Máquinas Sequenciais Síncronas

## Solução: Análise dos sinais de Controle c1 e c2

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1

Componente	Característica	Símbolo	Valor
Registadores R1, R2	tempo de setup	$tsu$	1 ns
Registadores R1, R2	tempo de hold	$th$	0,5 ns
Registadores R1, R2	tempo de carga	$tco$	1 ns
Somador completo (full adder)	atraso	$tds$	0,25 ns
Mux 2:1	atraso	$tdmux$	1 ns

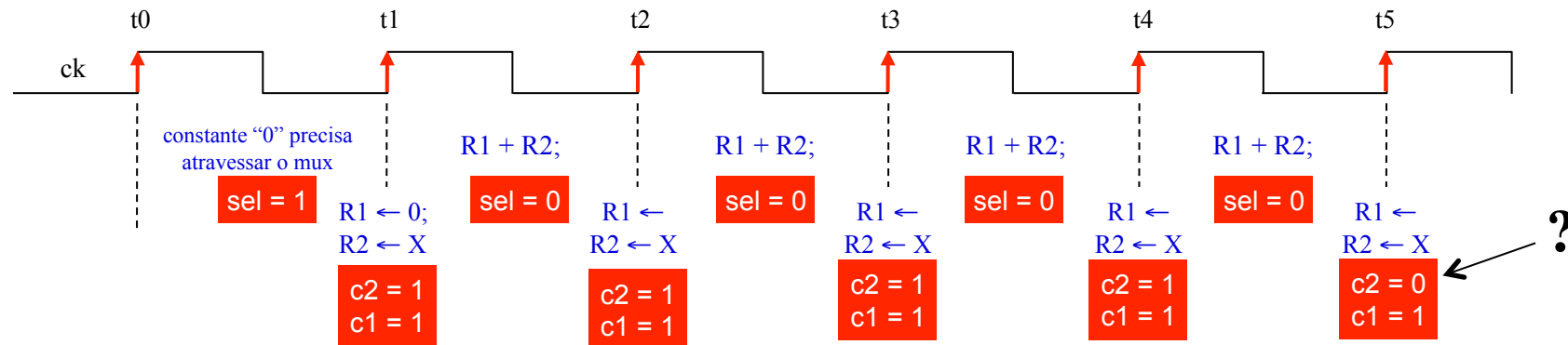
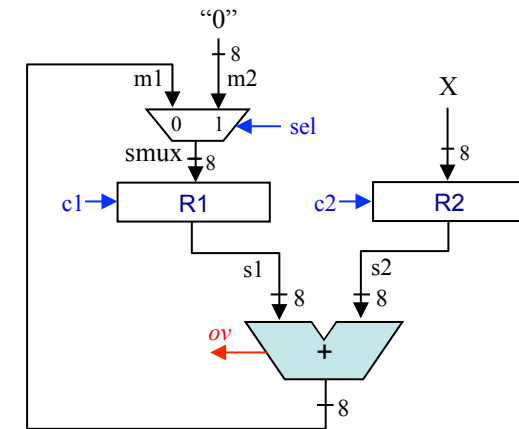
E supondo que "sel" tenha atraso=0



# Máquinas Sequenciais Síncronas

## Solução: Resumo dos valores para os sinais de controle

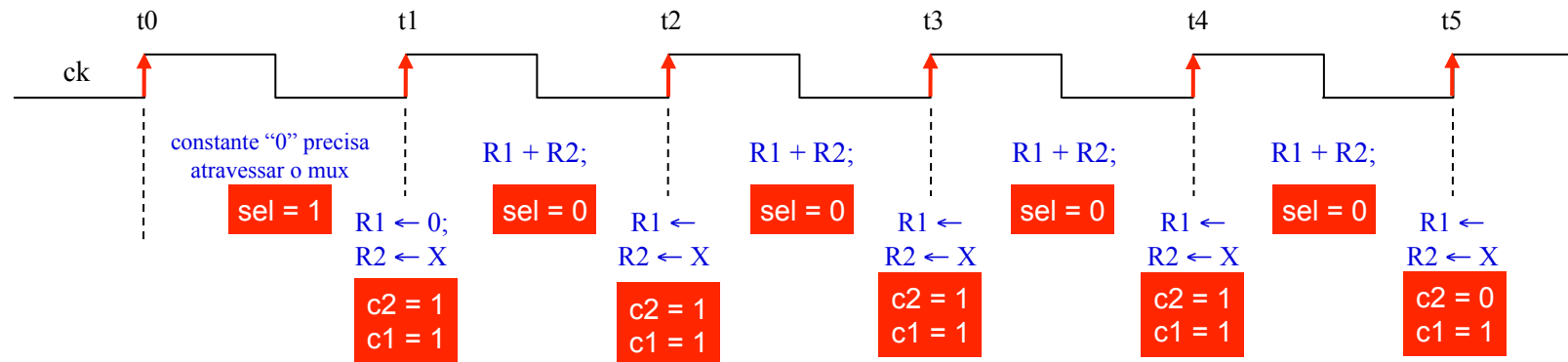
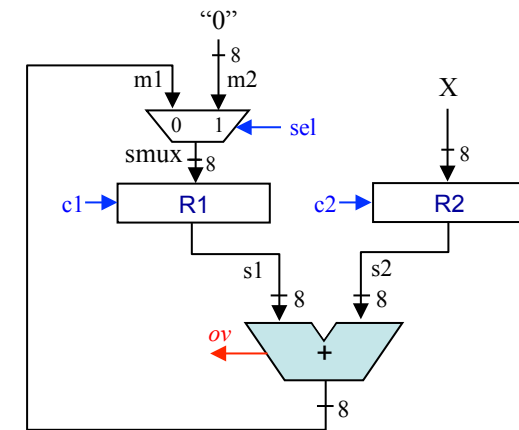
1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1



# Máquinas Sequenciais Síncronas

## Solução: Resumo dos valores para os sinais de controle

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1



Nesta disciplina assumiremos a seguinte assertiva conservadora:  
“toda vez que um registrador não é carregado explicitamente, o  
**valor de seu sinal de habilitação de carga deve ser “0” (zero)”**”



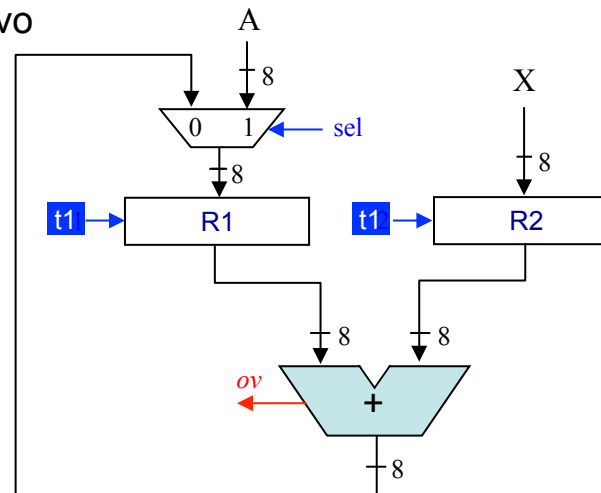
# Máquinas Sequenciais Síncronas

## Tempos de Estabilização

Supondo as seguintes características temporais dos componentes

Componente	Característica	Símbolo	Valor
Registradores R1, R2	tempo de setup	tsu	1 ns
Registradores R1, R2	tempo de hold	th	0,5 ns
Registradores R1, R2	tempo de carga	tco	1 ns
Somador completo ( <i>full adder</i> )	atraso	tds	0,25 ns
Mux 2:1	atraso	tdmux	1 ns
Sinal de controle (sel)	atraso	tdcontrol	0 ns

A análise inicia assumindo-se uma primeira borda de relógio aplicada a R1 e R2. Vamos assumir  $t_1$ , pois o ciclo de relógio  $t_1$ - $t_2$  é mais elucidativo

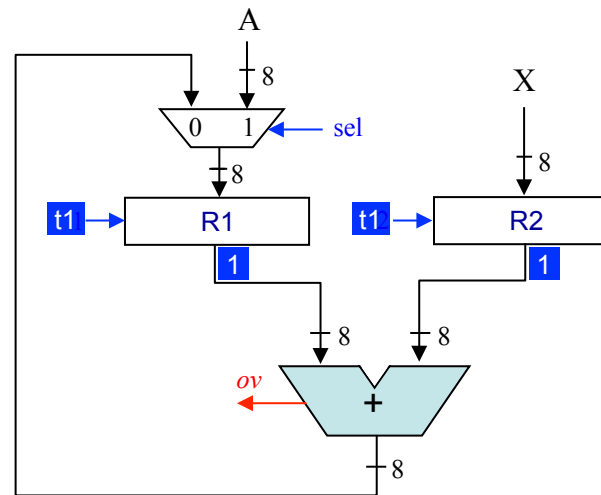


# Máquinas Sequenciais Síncronas

## Tempos de Estabilização

Supondo as seguintes características temporais dos componentes

Componente	Característica	Símbolo	Valor
Registradores R1, R2	tempo de setup	tsu	1 ns
Registradores R1, R2	tempo de hold	th	0,5 ns
Registradores R1, R2	tempo de carga	tco	1 ns
Somador completo ( <i>full adder</i> )	atraso	tds	0,25 ns
Mux 2:1	atraso	tdmux	1 ns
Sinal de controle (sel)	atraso	tdcontrol	0 ns

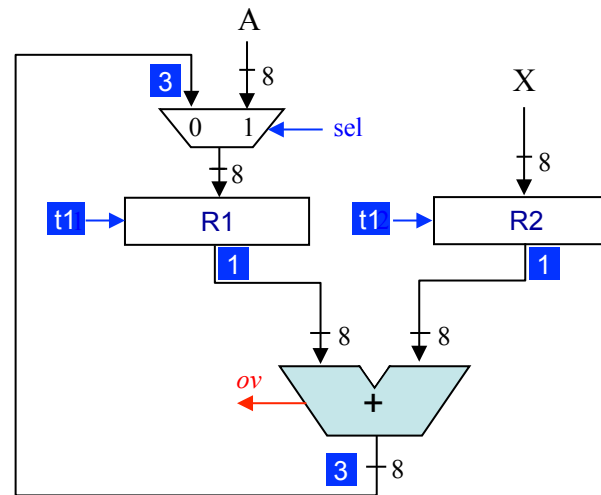


# Máquinas Sequenciais Síncronas

## Tempos de Estabilização

Supondo as seguintes características temporais dos componentes

Componente	Característica	Símbolo	Valor
Registradores R1, R2	tempo de setup	tsu	1 ns
Registradores R1, R2	tempo de hold	th	0,5 ns
Registradores R1, R2	tempo de carga	tco	1 ns
Somador completo ( <i>full adder</i> )	atraso	tds	0,25 ns
Mux 2:1	atraso	tdmux	1 ns
Sinal de controle (sel)	atraso	tdcontrol	0 ns

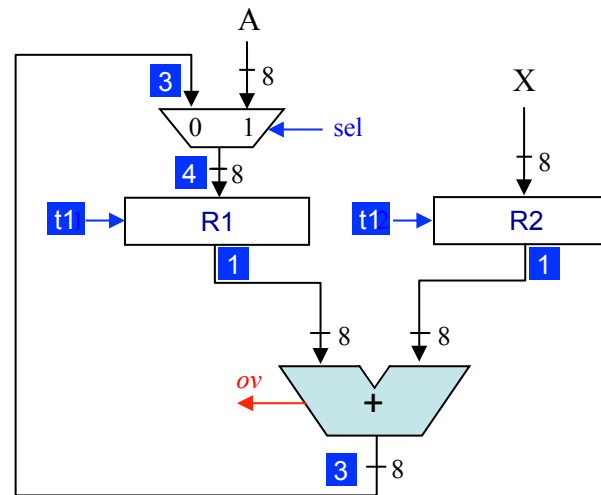


# Máquinas Sequenciais Síncronas

## Tempos de Estabilização

Supondo as seguintes características temporais dos componentes

Componente	Característica	Símbolo	Valor
Registradores R1, R2	tempo de setup	tsu	1 ns
Registradores R1, R2	tempo de hold	th	0,5 ns
Registradores R1, R2	tempo de carga	tco	1 ns
Somador completo ( <i>full adder</i> )	atraso	tds	0,25 ns
Mux 2:1	atraso	tdmux	1 ns
Sinal de controle (sel)	atraso	tdcontrol	0 ns



# Máquinas Sequenciais Síncronas

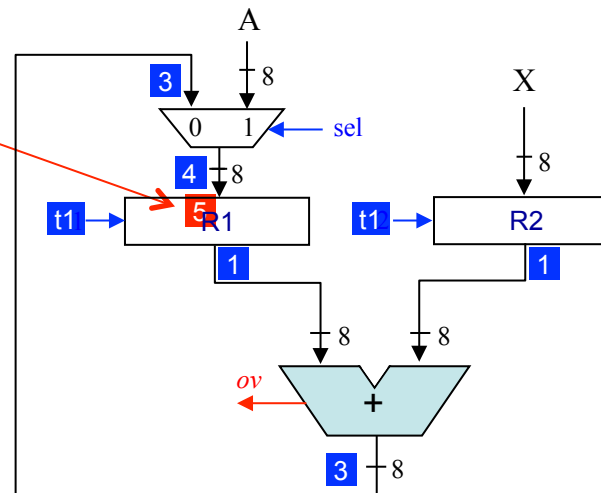
## Tempos de Estabilização

Supondo as seguintes características temporais dos componentes

Componente	Característica	Símbolo	Valor
Registradores R1, R2	tempo de setup	tsu	1 ns
Registradores R1, R2	tempo de hold	th	0,5 ns
Registradores R1, R2	tempo de carga	tco	1 ns
Somador completo ( <i>full adder</i> )	atraso	tds	0,25 ns
Mux 2:1	atraso	tdmux	1 ns
Sinal de controle (sel)	atraso	tdcontrol	0 ns

Considerando o tempo de setup do registrador R1

**Logo, atraso crítico =  $D = 5$  ns**



# Máquinas Sequenciais Síncronas

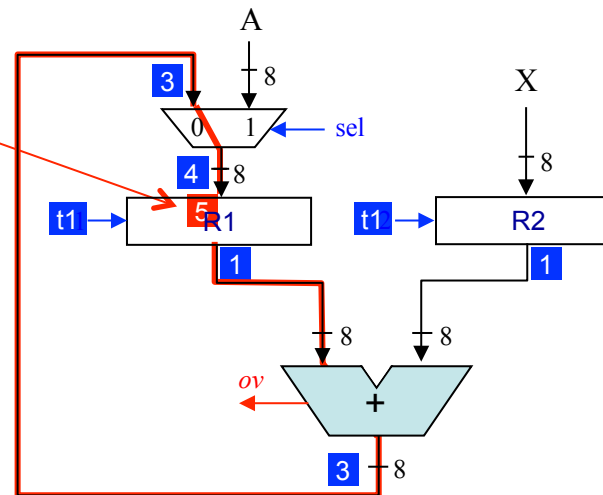
## Caminho Crítico e Atraso Crítico

Supondo as seguintes características temporais dos componentes

Componente	Característica	Símbolo	Valor
Registradores R1, R2	tempo de setup	tsu	1 ns
Registradores R1, R2	tempo de hold	th	0,5 ns
Registradores R1, R2	tempo de carga	tco	1 ns
Somador completo ( <i>full adder</i> )	atraso	tds	0,25 ns
Mux 2:1	atraso	tdmux	1 ns
Sinal de controle (sel)	atraso	tdcontrol	0 ns

Considerando o tempo de setup do registrador R1

Logo, atraso crítico =  $D = 5 \text{ ns}$

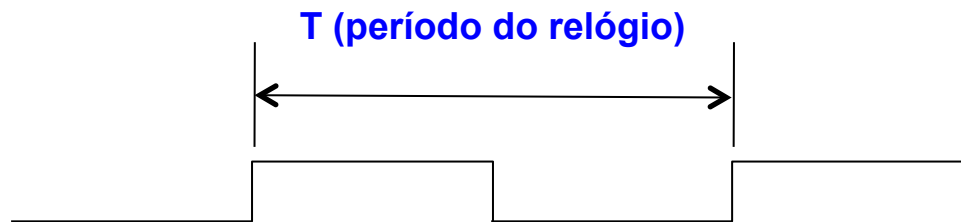
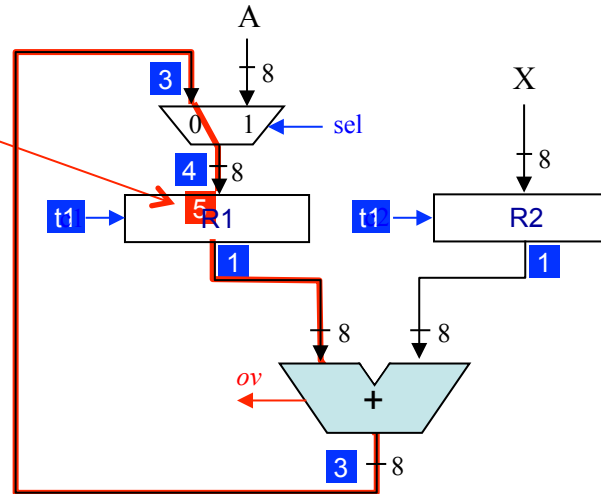


# Máquinas Sequenciais Síncronas

## Estimativa do Período (Mínimo) do Relógio

Considerando o tempo de setup do registrador R1

Logo, atraso crítico =  $D = 5 \text{ ns}$



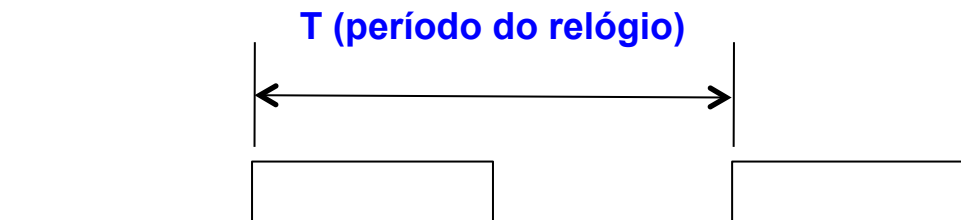
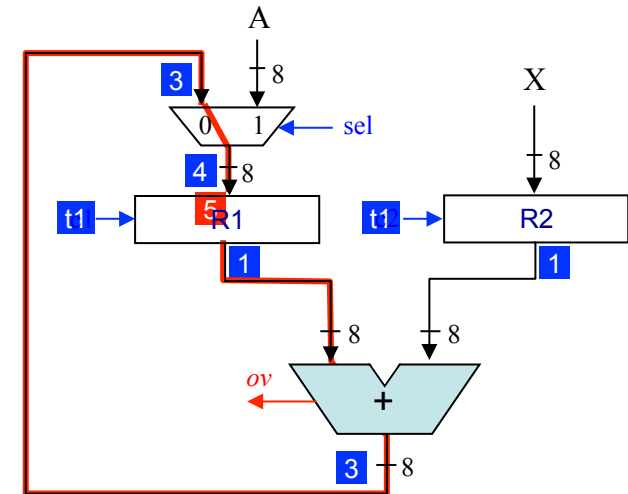
sendo  $T \geq D$

# Máquinas Sequenciais Síncronas

## Estimativa do Período (Mínimo) do Relógio (2)

Pode-se pular a etapa de determinação dos TEs, calculando o atraso crítico diretamente a partir do(s) caminho(s) crítico(s). Exemplo:

$$\begin{aligned} D &= t_{co_{R1}} + 8 \cdot t_{ds} + t_{mux} + t_{su_{R1}} = \\ &= t_{co_{R2}} + 8 \cdot t_{ds} + t_{mux} + t_{su_{R1}} = \\ &= 1 \text{ ns} + 8 \cdot 0,25 \text{ ns} + 1 \text{ ns} + 1 \text{ ns} = \mathbf{5 \text{ ns}} \end{aligned}$$



**sendo  $T \geq D$**

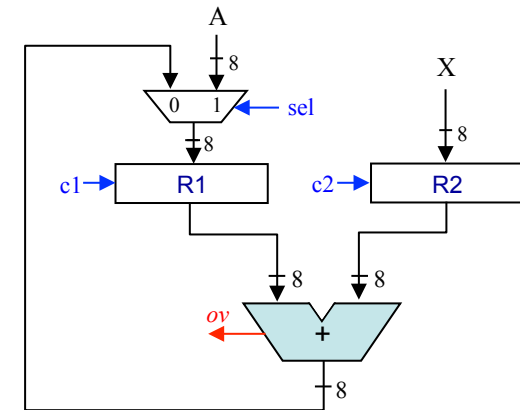


# Máquinas Sequenciais Síncronas

## Tempo de Execução

No exemplo em questão, o processamento requer 4 ciclos de relógio, com  $T \geq D = 5 \text{ ns}$ .

1.  $R1 \leftarrow 0$ ;  $R2 \leftarrow X$ ; // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2$ ;  $R2 \leftarrow X$ ; // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2$ ; // O resultado final S estará em R1



$$\text{Tempo de execução} = n^{\circ} \text{ ciclos} \times T$$

Onde:

$n^{\circ}$  de ciclos que o processamento demora

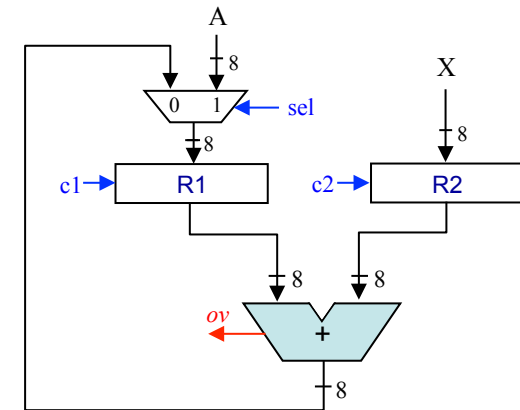
$T$  = período do relógio, ( $f = 1/T$  é a frequência do relógio)

# Máquinas Sequenciais Síncronas

## Tempo de Execução

No exemplo em questão, o processamento requer 4 ciclos de relógio, com  $T \geq D = 5 \text{ ns}$ .

1.  $R1 \leftarrow 0; R2 \leftarrow X;$  // A deve estar estável na entrada X
2.  $R1 \leftarrow R1 + R2; R2 \leftarrow X;$  // B deve estar estável na entrada X
3.  $R1 \leftarrow R1 + R2; R2 \leftarrow X;$  // C deve estar estável na entrada X
4.  $R1 \leftarrow R1 + R2; R2 \leftarrow X;$  // D deve estar estável na entrada X
5.  $R1 \leftarrow R1 + R2;$  // O resultado final S estará em R1



$$\begin{aligned}\text{Tempo de execução} &= n^{\circ} \text{ ciclos} \times T = \\ &= 5 \times 5 \text{ ns} = 25 \text{ ns}\end{aligned}$$

# Máquinas Sequenciais Síncronas

## Comparação com Versão Monociclo

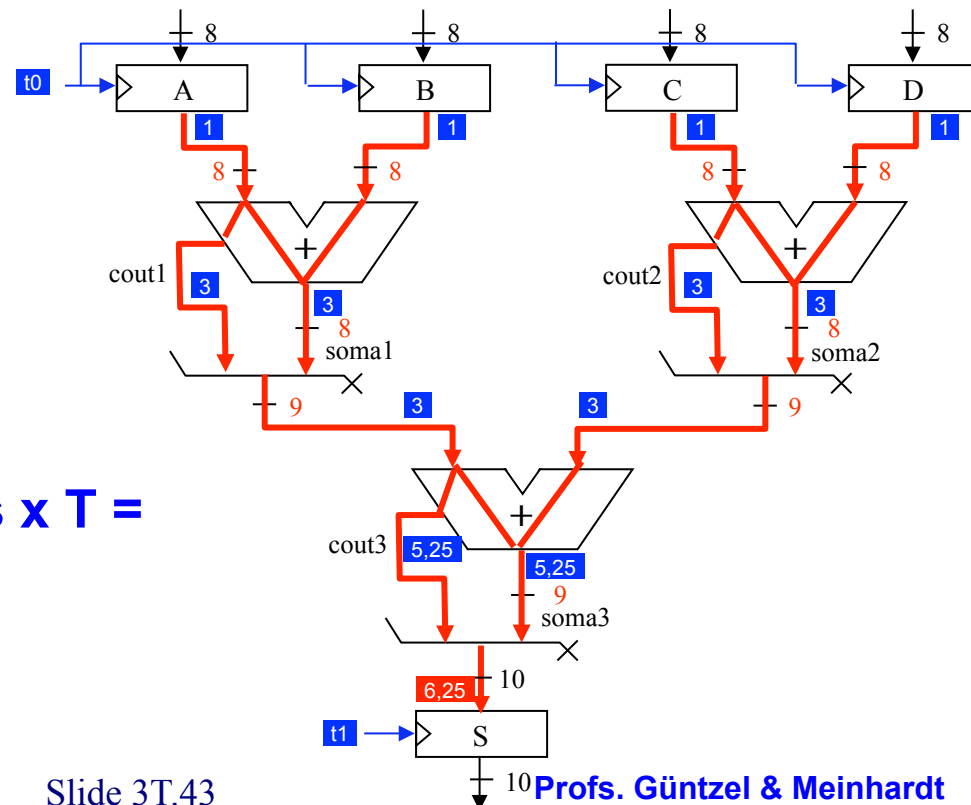
Componente	Característica	Símbolo	Valor
Registadores A, B, C, D, S	tempo de setup	tsu	1 ns
Registadores A, B, C, D, S	tempo de hold	th	0,5 ns
Registadores A, B, C, D, S	tempo de carga	tco	1 ns
Somador completo ( <i>full adder</i> )	atraso	tds	0,25 ns
Sinal de controle (sel)	atraso	tdcontrol	0 ns

Nesta versão monociclo, não há

Calculando o atraso crítico diretamente a partir do(s) caminho(s) crítico(s):

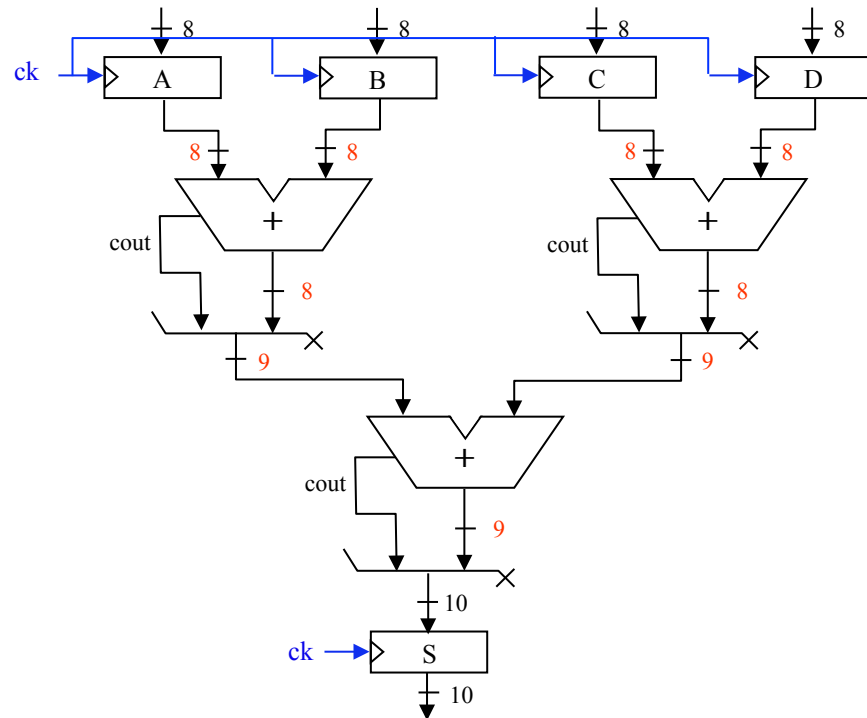
$$\begin{aligned}
 D = T &= t_{co} + t_{dC} + t_{su}(S) = \\
 &= 1 + 8 \times 0,25 + 9 \times 0,25 + 1 = \\
 &= \mathbf{6,25 \text{ ns}}
 \end{aligned}$$

$$\begin{aligned}
 \text{Tempo de execução} &= n^{\circ} \text{ ciclos} \times T = \\
 &= 1 \times 6,25 \text{ ns} = \mathbf{6,25 \text{ ns}}
 \end{aligned}$$

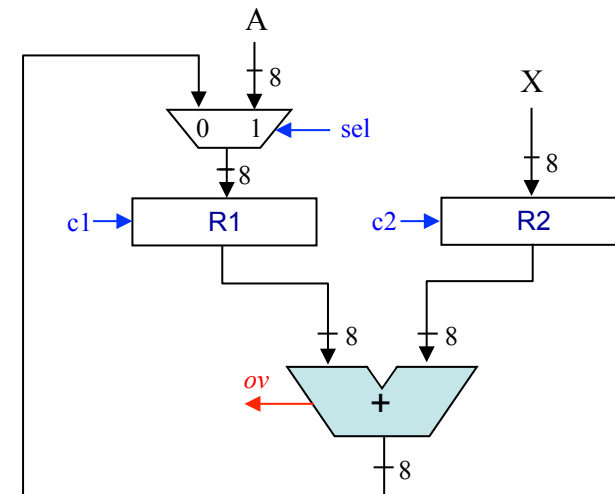


# Máquinas Sequenciais Síncronas

## Tempo de Execução: Multiciclo x Monociclo



Tempo de execução = 6,25 ns

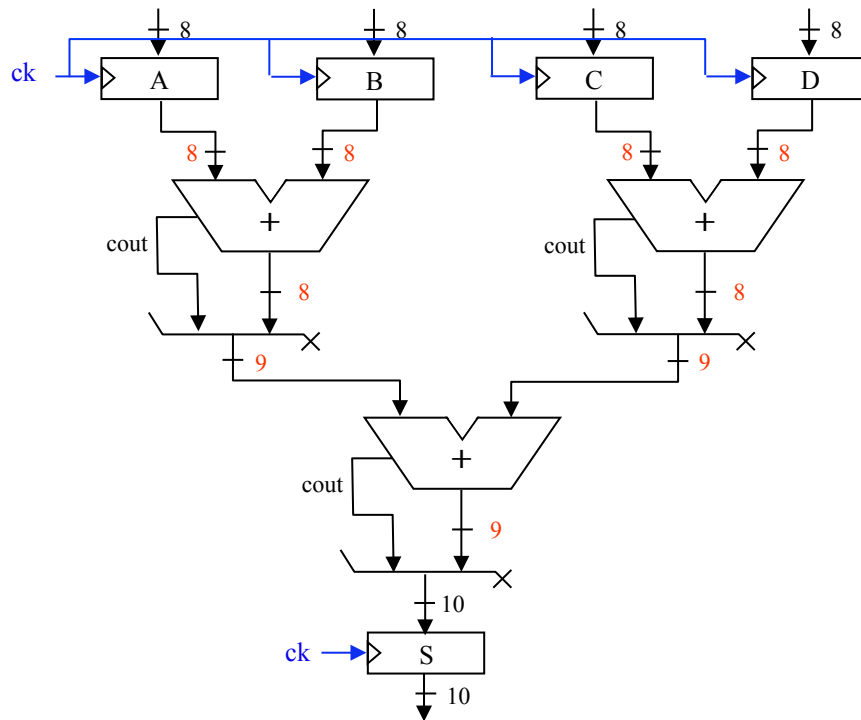


Tempo de execução = 25 ns

Tempo de execução(monociclo) / Tempo de execução(multiciclo) =  $6,25/25 = 0,25$   
Versão monociclo leva  $\frac{1}{4}$  do tempo que a versão multiciclo para concluir o cálculo

# Máquinas Sequenciais Síncronas

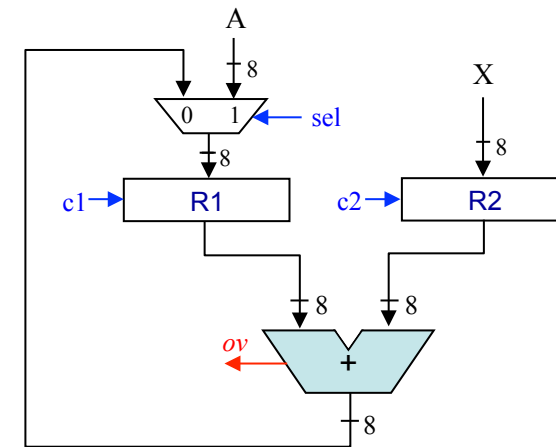
## Custo do Hardware: Multiciclo x Monociclo



$$\begin{aligned} \text{Custo} &= 4 \times 18 \times 8 + 2 \times 24 \times 8 + 1 \times 24 \times 9 + \\ &\quad 1 \times 18 \times 10 = \\ &= 704 + 384 + 216 + 220 = \mathbf{1356 \text{ transistores}} \end{aligned}$$

Componente RT	Custo, nº de trans
Somador	24n
Mux 2:1	4n
Reg. com carga paralela	18n
Reg. com carga paralela controlada	22n

onde  $n = \text{nº de bits}$

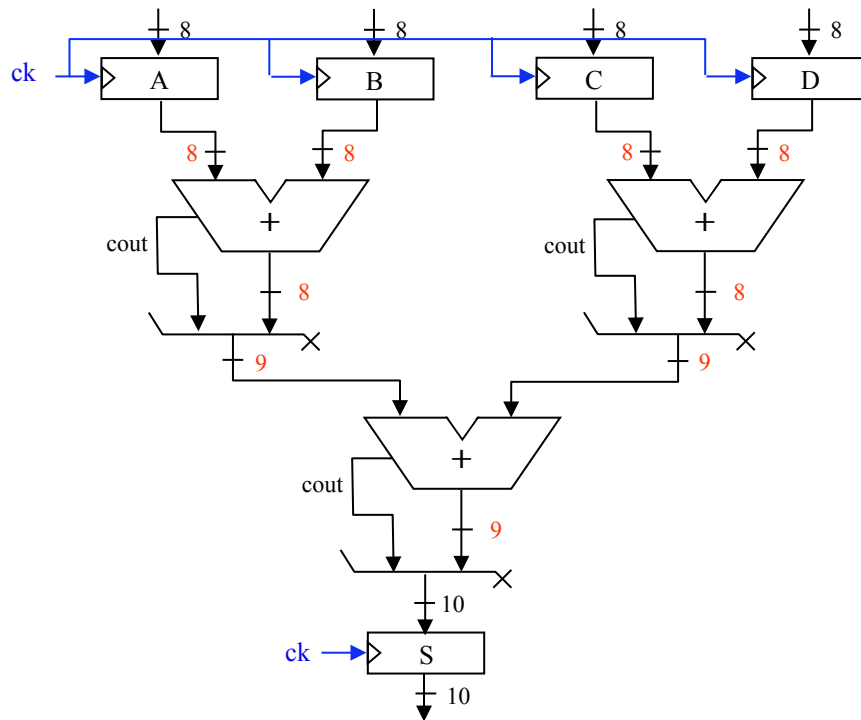


$$\begin{aligned} \text{Custo} &= 8 \times (2 \times 22 + 1 \times 24 + 1 \times 4) = \\ &= 352 + 192 + 32 = \mathbf{576 \text{ transistores}} \end{aligned}$$

**Porém, falta o custo do B.C.**

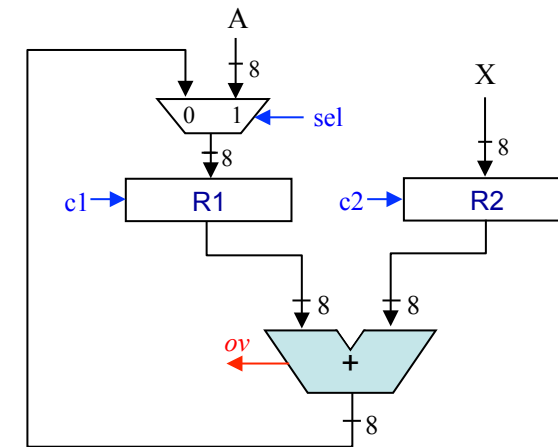
# Máquinas Sequenciais Síncronas

## Custo do Hardware: Multiciclo x Monociclo



Componente RT	Custo, nº de trans
Somador	24n
Mux 2:1	4n
Reg. com carga paralela	18n
Reg. com carga paralela controlada	22n

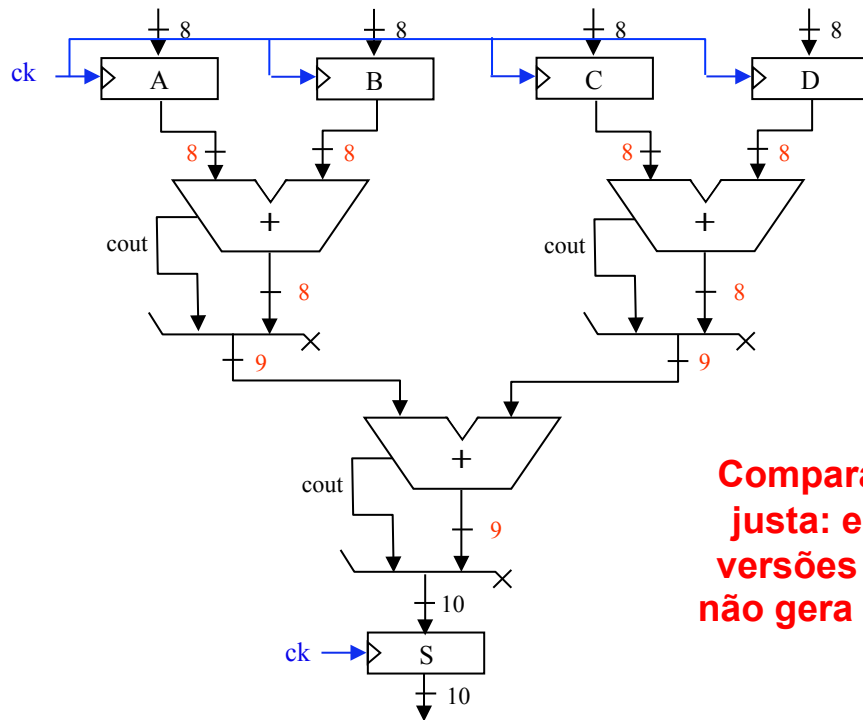
onde  $n$  = nº de bits



**Mas, na versão monociclo nunca ocorre *overflow*!!**  
**Já na versão multiciclo pode ocorrer *overflow*...**  
**Logo, esta comparação não parece “justa”...**

# Máquinas Sequenciais Síncronas

## Custo do Hardware: Multiciclo x Monociclo (2)

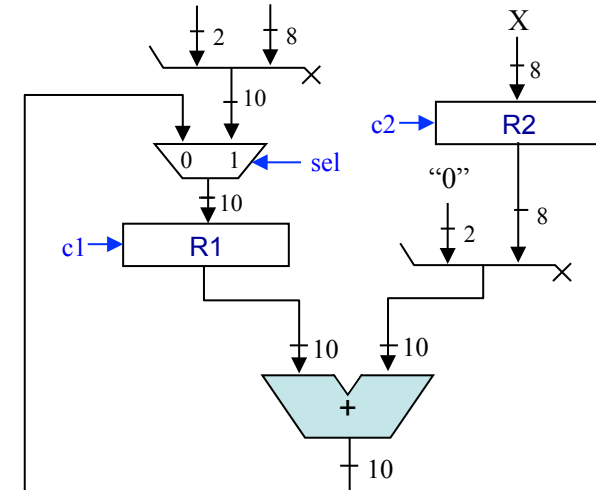


Componente RT	Custo, nº de trans
Somador	24n
Mux 2:1	4n
Reg. com carga paralela	18n
Reg. com carga paralela controlada	22n

onde n = nº de bits

**Comparação mais justa: em ambas versões A+B+C+D não gera overflow ...**

$$\begin{aligned} \text{Custo} &= 4 \times 18 \times 8 + 2 \times 24 \times 8 + 1 \times 24 \times 9 + 1 \times 18 \times 10 = \\ &= 704 + 384 + 216 + 220 = \mathbf{1356 \text{ transistores}} \end{aligned}$$



$$\begin{aligned} \text{Custo} &= 8 \times 22 + 10 \times (1 \times 22 + 1 \times 24 + 1 \times 4) = \\ &= 352 + 192 + 32 = \mathbf{676 \text{ transistores}} \end{aligned}$$

**Porém, ainda falta o custo do B.C.**

# Processadores Dedicados

## Classificação dos Sistemas Digitais Quanto à Aplicação

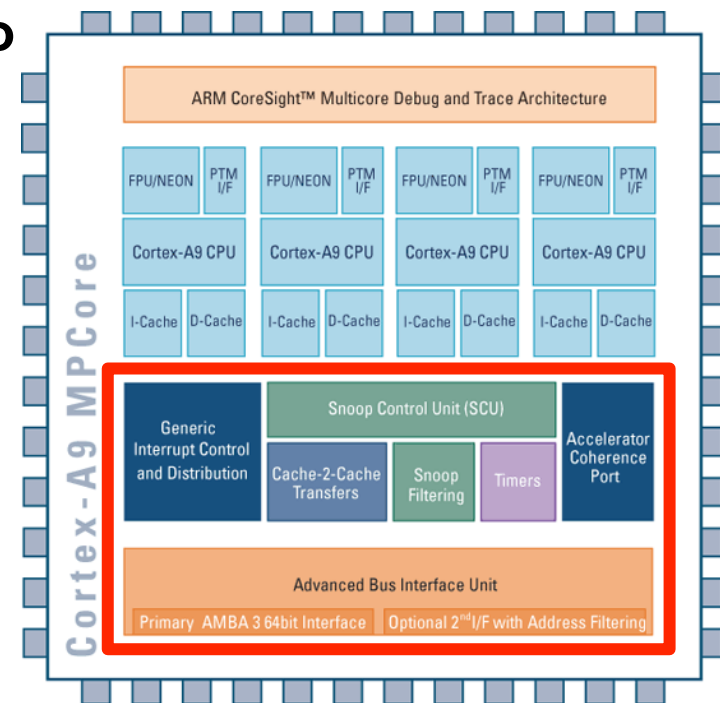
### 3. Processadores Dedicados ou Blocos Aceleradores ou Blocos IPs\* :

- Projetados para **uma aplicação ou algoritmo específico**.
- Exemplos: controladores de I/O, controladores de memória, codecs\*\* de imagens (jpeg, MPEG, H.264/AVC, VP9, HEVC, AV1), codecs\*\* de áudio, cifradores/decifradores\*\*\* etc
- Geralmente, usados onde há necessidade de altíssimo desempenho e/ou alta eficiência energética (por exemplo, dispositivos pessoais, alimentados por bateria).

\* *Intellectual Property Blocks*

\*\* codificadores/decodificadores

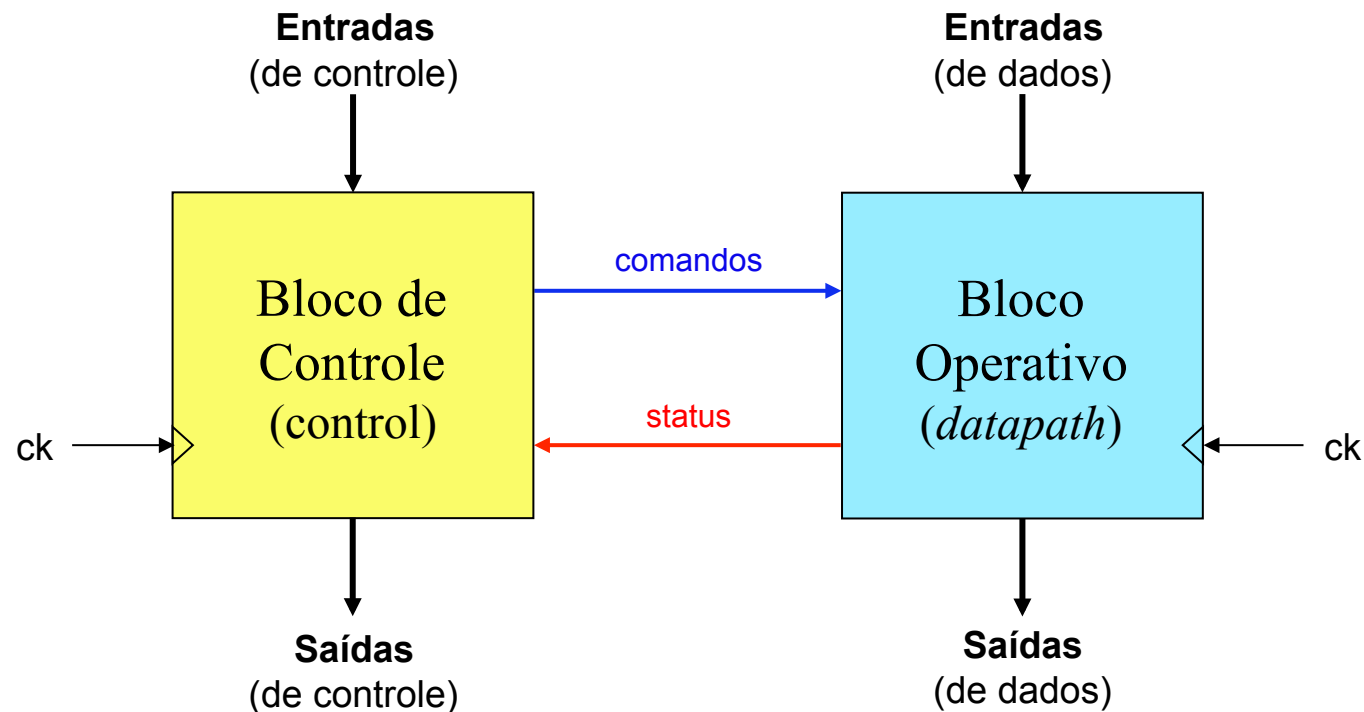
\*\*\* podem usar mais de um algoritmo de criptografia





# O Modelo BO/BC

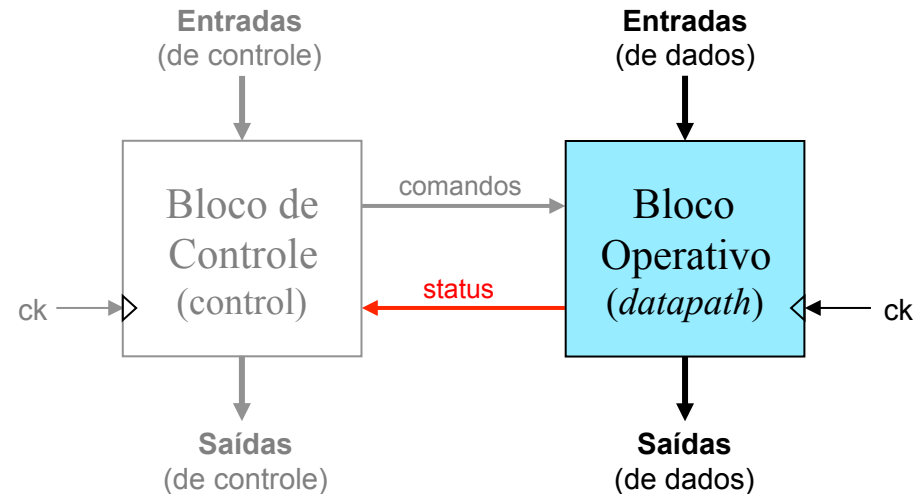
## O Modelo Bloco Operativo / Bloco de Controle



- Este é um modelo simples de sistemas digitais, porém didático e aplicável na maioria dos casos.

# O Modelo BO/BC

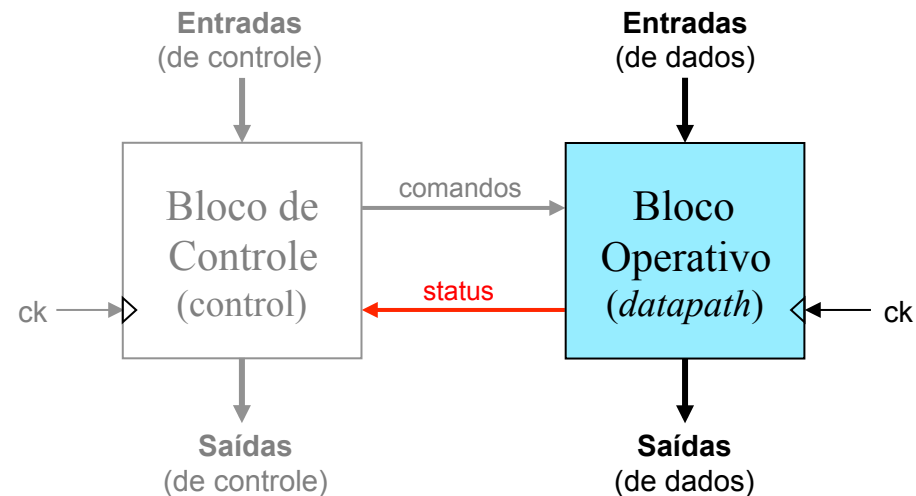
## Bloco Operativo: funções



- Realiza transformações sobre dados, geralmente provenientes do ambiente externo
- As transformações são realizadas em um ou mais passos, cada passo demorando um ciclo de relógio
- Gera sinais de “status” que são usados pelo Bloco de Controle para definir a sequência de operações a serem realizadas (às vezes são chamados de “*flags*”)

# O Modelo BO/BC

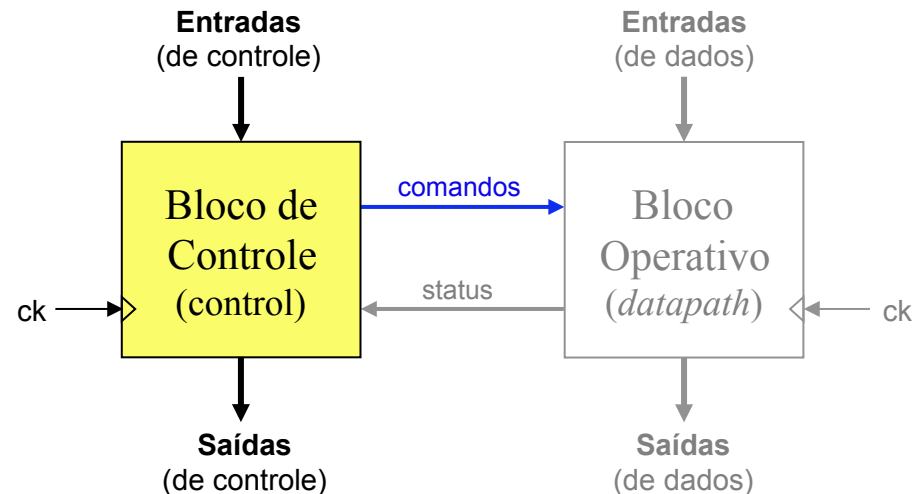
## Bloco Operativo: componentes



- Unidades Funcionais (UFs): somadores, subtratores, deslocadores, multiplicadores, UFs combinadas (somadores/subtratores, ULAs)
- Elementos de armazenamento: registradores ou banco de registradores, memórias (SRAM)
- Rede de interconexão: fios, multiplexadores, barramentos + *buffers tri-state*

# O Modelo BO/BC

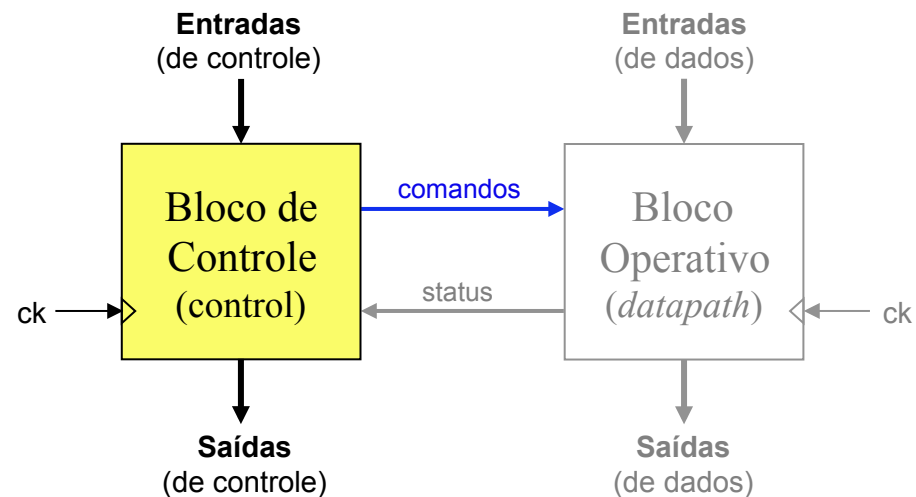
## Bloco de Controle: funções



- Gera comandos, que são sinais de controle na ordem necessária para que o bloco operativo realize os passos desejados
- Recebe sinais de controle do ambiente externo: opcode, no caso de CPUs, sinais específicos (por exemplo, “iniciar”), no caso de CPUs dedicadas e de blocos aceleradores
- Pode gerar uma ou mais saídas de controle para se comunicar com outros sistemas digitais (p. ex.: “done”, “bus request”, “ack”)

# O Modelo BO/BC

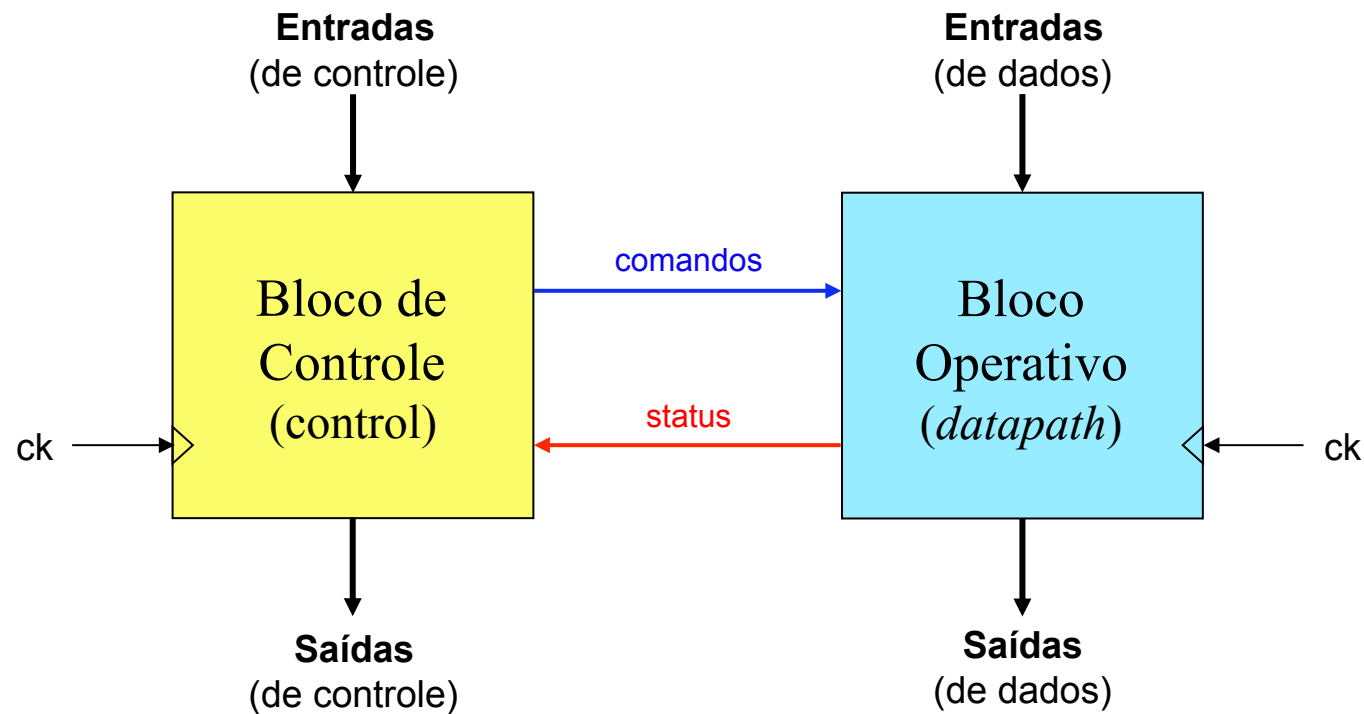
## Bloco de Controle: componentes



- Em um esquema monociclo: é um bloco combinacional
- Em um esquema multiciclo: é uma máquina de estados (FSM – Finite State Machine)

# O Modelo BO/BC

## O Modelo Bloco Operativo / Bloco de Controle

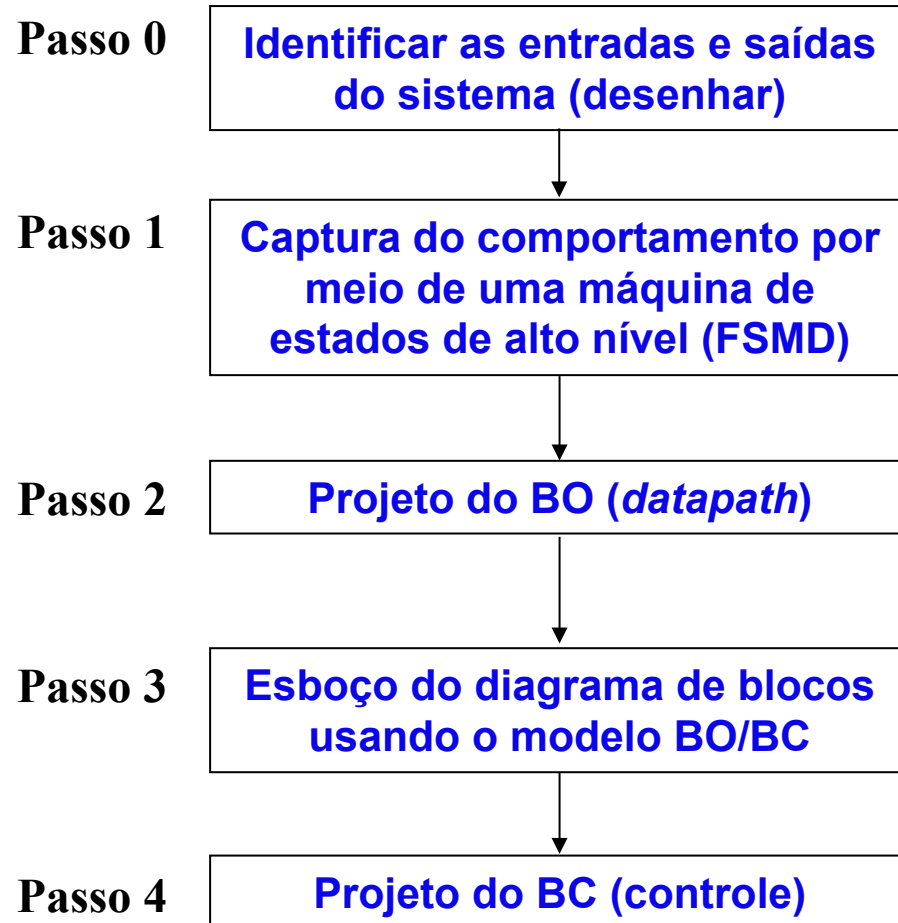


- Este é um modelo simples de sistemas digitais, porém didático e aplicável na maioria dos casos.

# Projeto no Nível RT

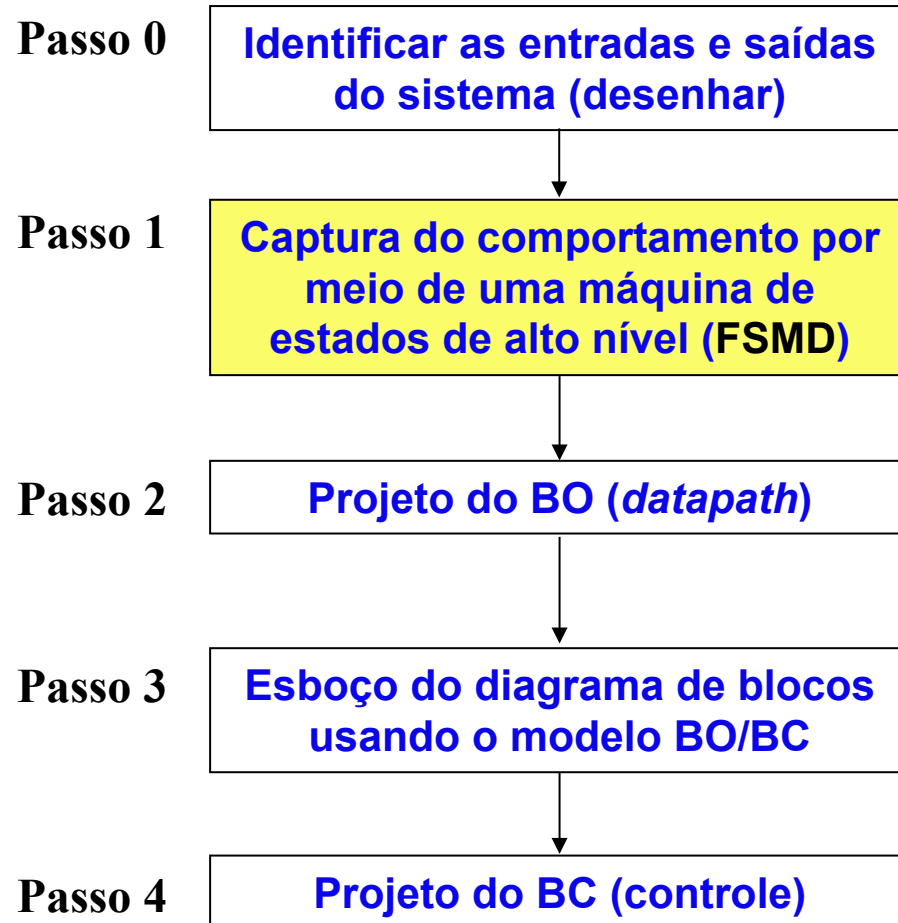
---

## Método de Projeto



# Projeto no Nível RT

## Método de Projeto



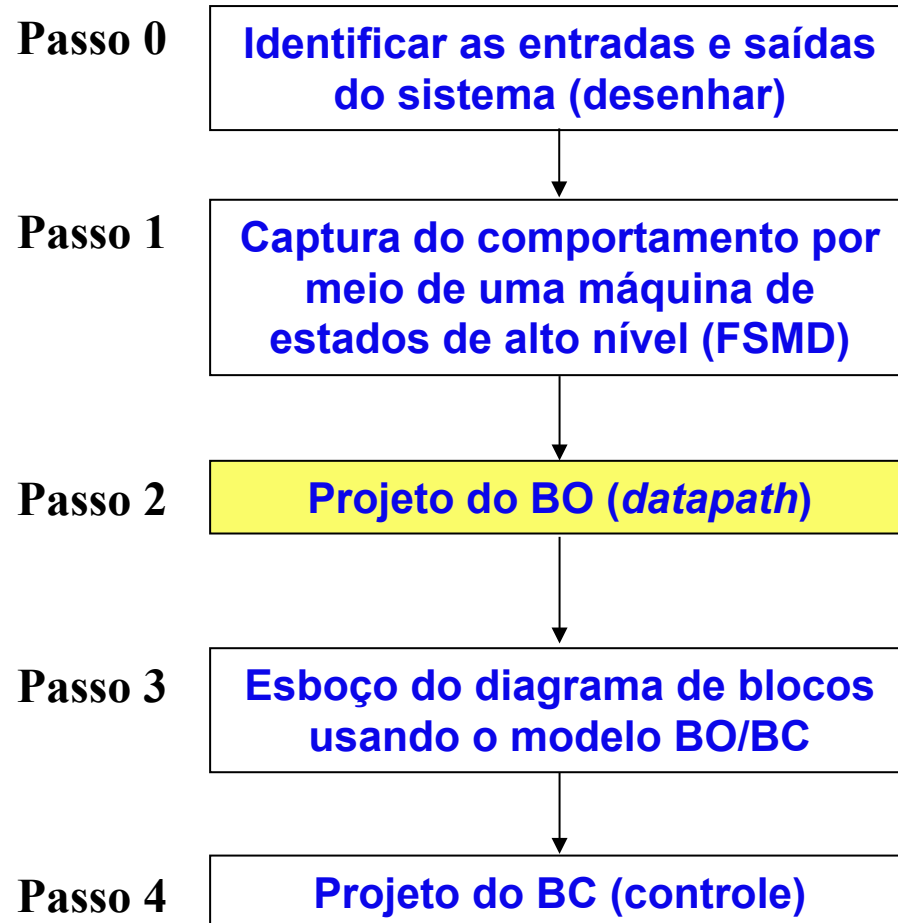
FSMD é uma extensão de uma máquina de estados, na qual:

- Entradas e saídas correspondem a dados com mais de um bit.
- Há variáveis locais para armazenar dados temporários e entradas/saídas (se necessário)
- Ações e condições podem envolver equações e expressões aritméticas (ao invés de apenas equações e expressões Booleanas).



# Projeto no Nível RT

## Método de Projeto

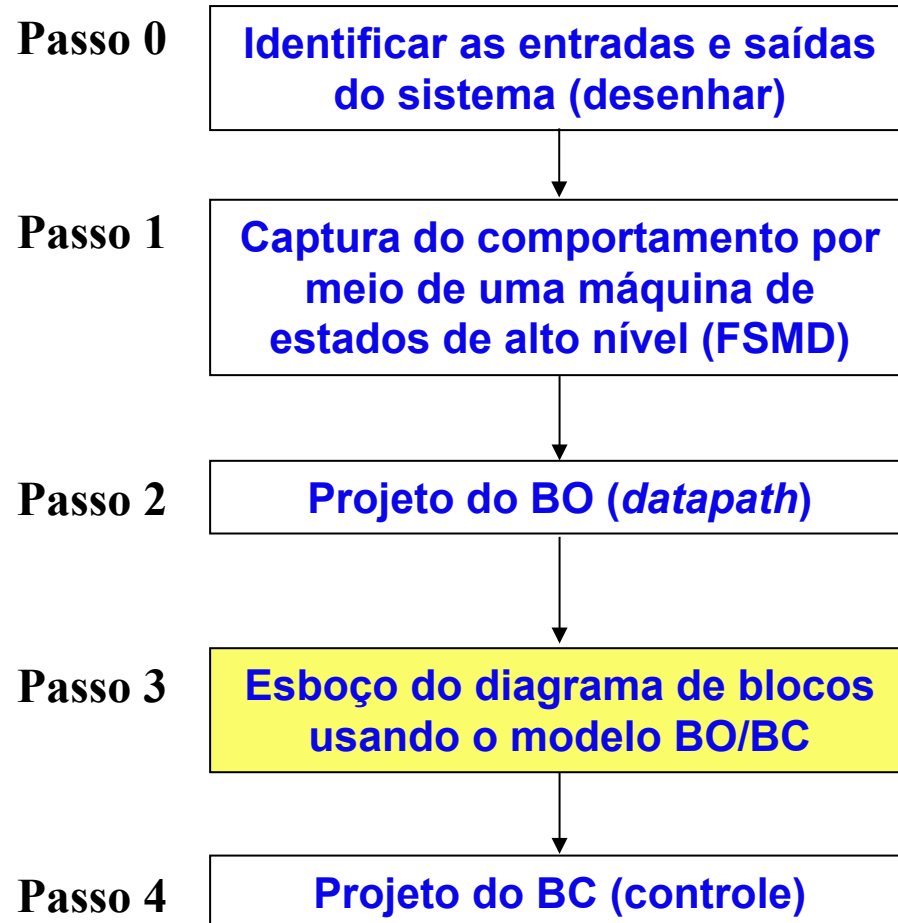


Analizando a FSMD, identificar:

- Os registradores para armazenar dados.
- As operações aritméticas (e lógicas) necessárias para operar os dados e para as expressões a serem usadas como condições de troca de estados.
- Selecionar os componentes do nível RT para implementar, conforme identificado no passo anterior.
- Conectar os componentes do nível RT selecionados no passo anterior.

# Projeto no Nível RT

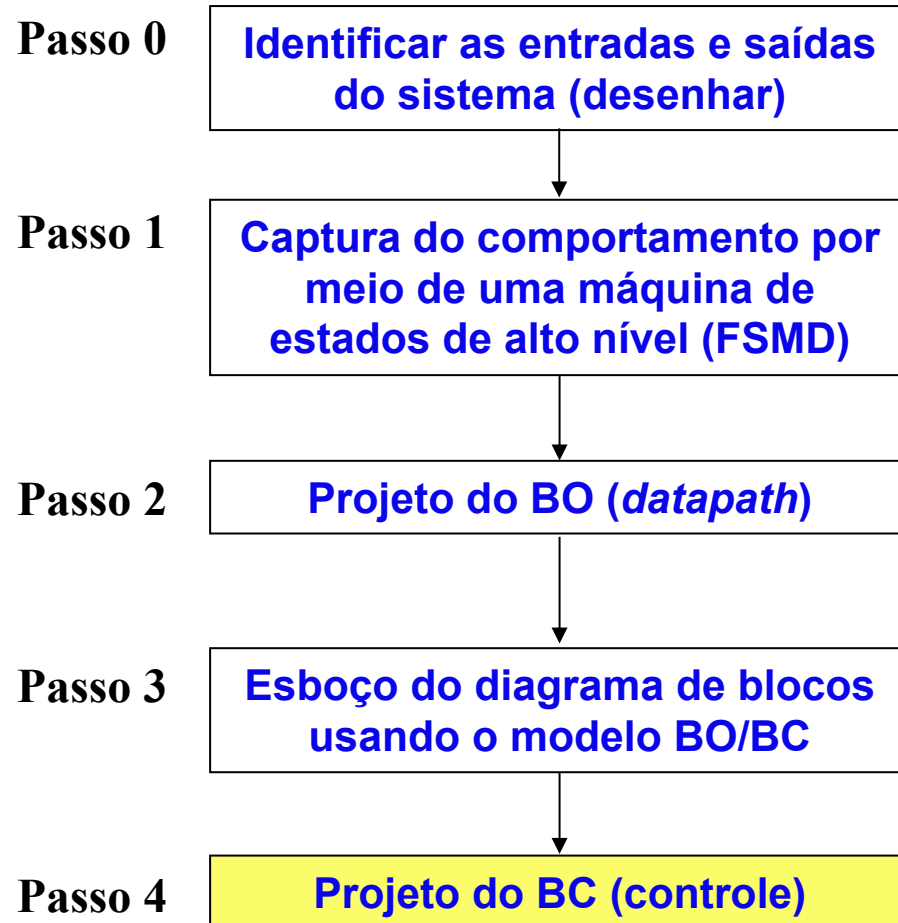
## Método de Projeto



- Desenhar o diagrama de blocos segundo o modelo BO/BC.
- No desenho, identificar todos os sinais (nome e número de bits): entradas, saídas, sinais de status, sinais de comando.

# Projeto no Nível RT

## Método de Projeto



- A partir da FSMD inicial e observando os nomes dos sinais definidos no passo 3, projetar a FSM que deve controlar o BO projetado (conforme visto na parte anterior desta disciplina).

# Projeto no Nível RT

---

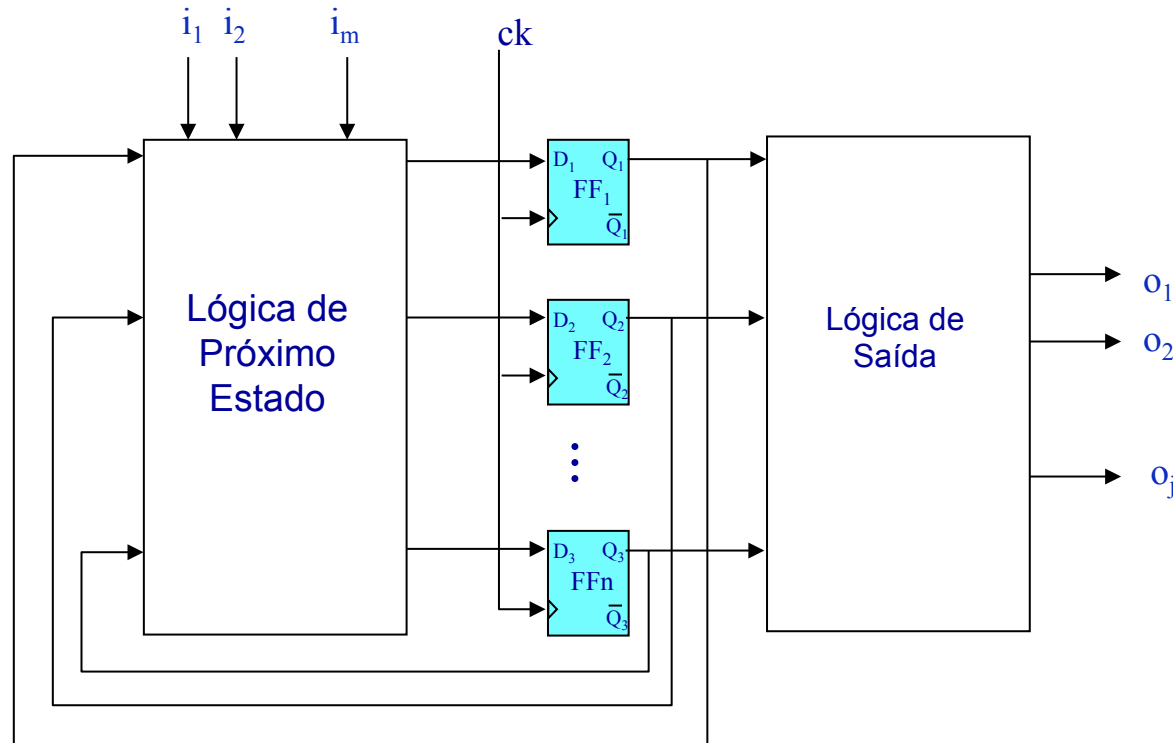
## Máquinas de Estados Finitos (FSM)

- Podem ser síncronas (cadenciadas por um sinal monótono chamado relógio ou *clock*) ou assíncronas (sem relógio).
- Máquina Sequenciais Síncronas são mais utilizadas porque:
  - São mais fáceis de projetar e de validar.
  - Têm operação mais segura, são mais robustas.
- Há dois modelos: Moore e Mealy.
- Registradores podem ser vistos como Máquina Sequenciais Síncronas.

# Máquinas Sequenciais Síncronas

## Modelo de Moore

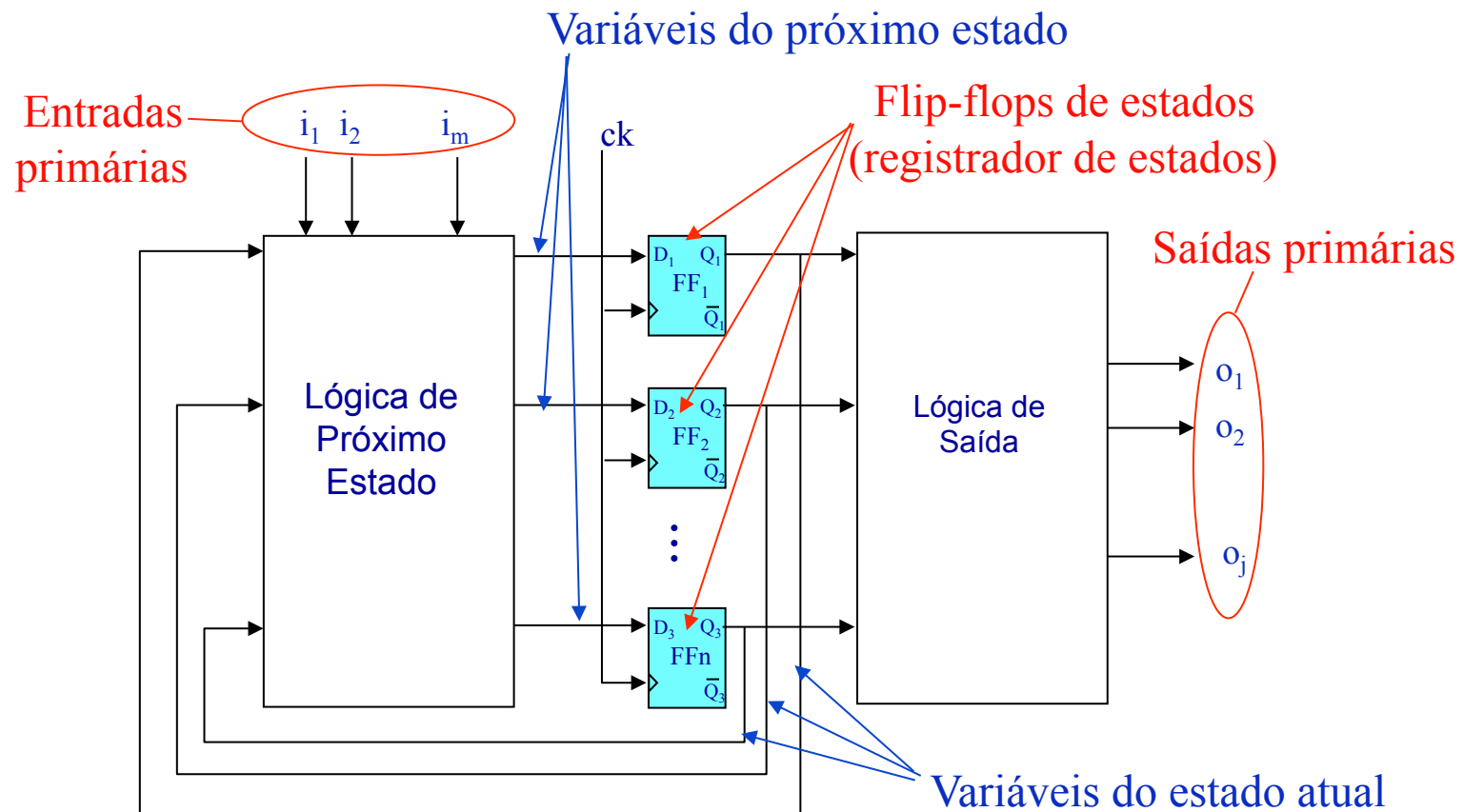
Característica principal: as saídas dependem apenas do estado atual.



# Máquinas Sequenciais Síncronas

## Modelo de Moore

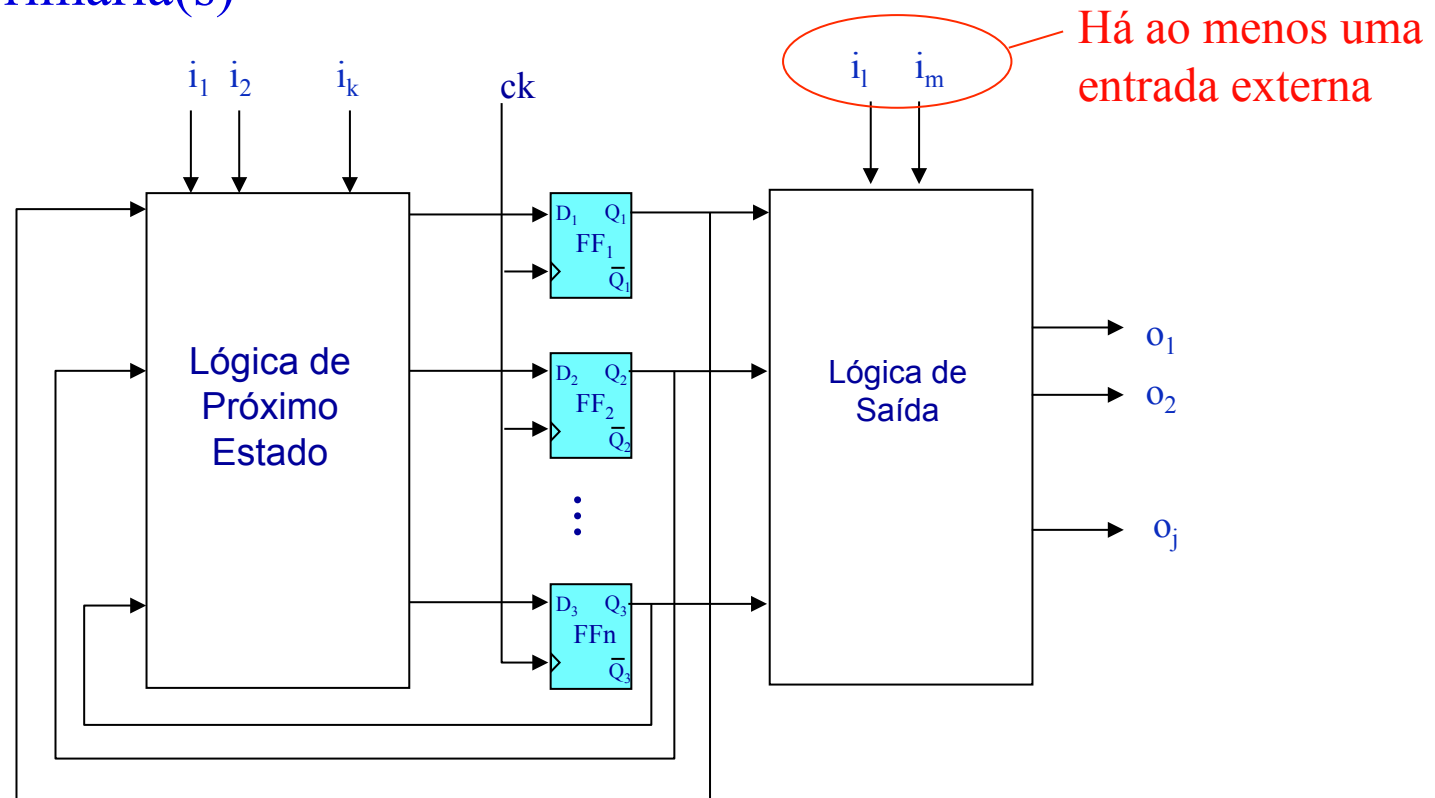
Característica principal: as saídas dependem apenas do estado atual.



# Máquinas Sequenciais Síncronas

## Modelo de Mealy

Característica principal: as saídas dependem do estado atual e de entrada(s) primária(s)



# Máquinas Sequenciais Síncronas

---

## Síntese de Circuitos Sequenciais

Roteiro para a Síntese (=Projeto)

1. Determinar quantos estados são necessários (e o nº de variáveis de estado)
2. Construir o diagrama de estados, observando com cuidado o comportamento solicitado para a FSM e adotando um modelo de FSM (Moore ou Mealy):
  - Determinar as transições entre estados necessárias
  - Selecionar um estado para servir como estado inicial
3. Construir a tabela de próximo estado e a tabela das saídas
4. Escolher uma codificação para os estados e definir o tipo de flip-flops para compor o registrador de estados.
5. Sintetizar (projetar) os circuitos combinacionais: lógica de próximo estado e lógica de saída.