



**Universidade Federal de Santa Catarina**  
**Centro Tecnológico**  
Departamento de Informática e Estatística  
Curso de Graduação em Ciências da Computação



# **Sistemas Digitais**

**INE 5406**

## **Aula 12-P**

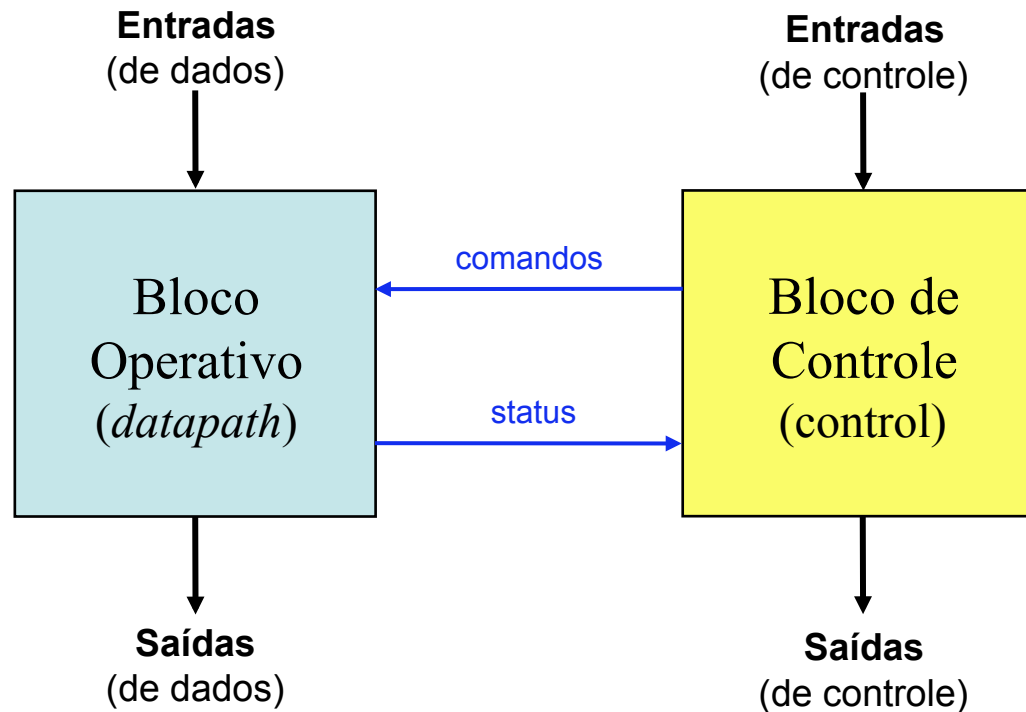
**Descrição em VHDL, síntese e simulação de máquinas de estados finitos (FSMs). FSM como Bloco de Controle.**

**Prof. José Luís Güntzel      Est. Doc. Eberle Rambo**  
**{guntzel, eberle18}@inf.ufsc.br**

**[www.inf.ufsc.br/~guntzel/ine5406/ine5406.html](http://www.inf.ufsc.br/~guntzel/ine5406/ine5406.html)**

# FSM como Bloco de Controle

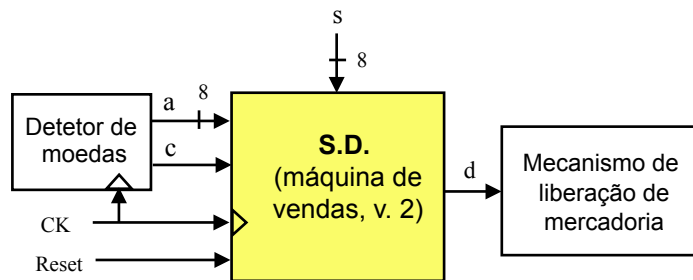
## ▶ O Modelo Bloco Operativo / Bloco de Controle



# FSM como Bloco de Controle

## ▶ Projetando um Sistema Digital

### Exemplo 1: Passo 1 (captura do comportamento com FSM)

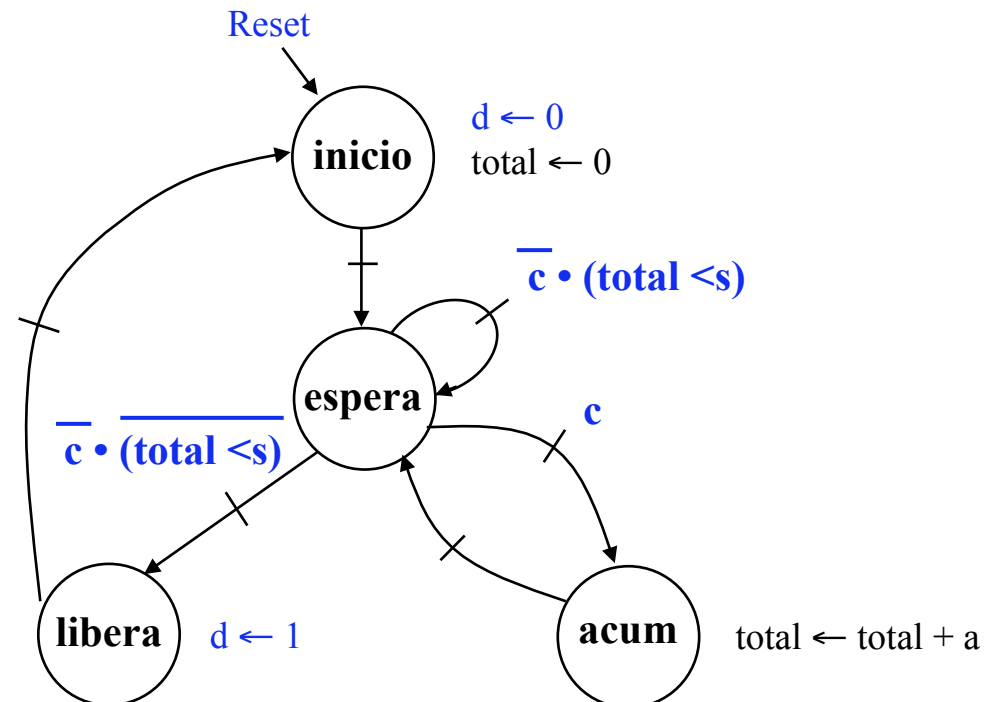


**Entradas:**  $c$  (1 bit),  $a$  (8 bits),  $s$  (8 bits)

**Saídas:**  $d$  (1 bit)

**Variáveis internas ou locais:**  $total$  (8 bits)

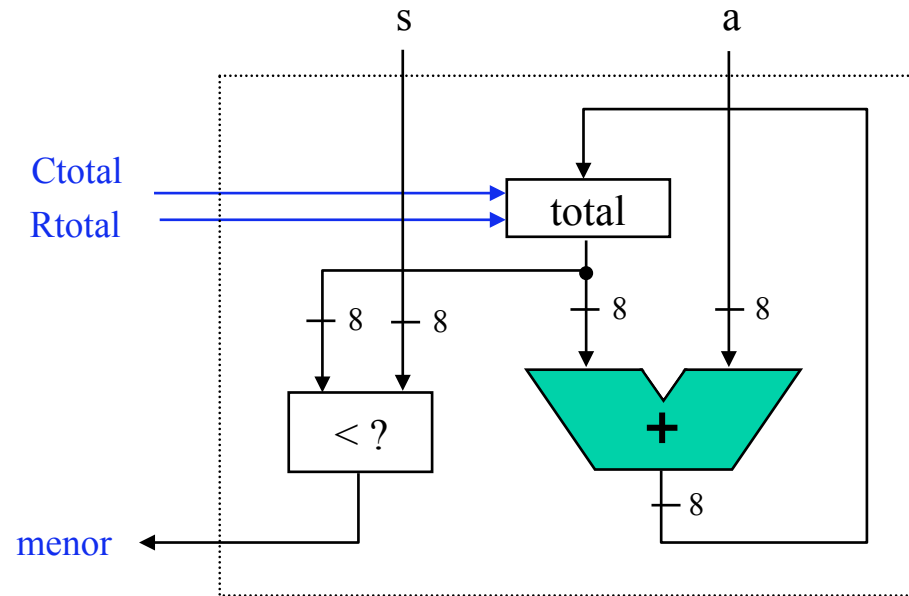
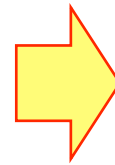
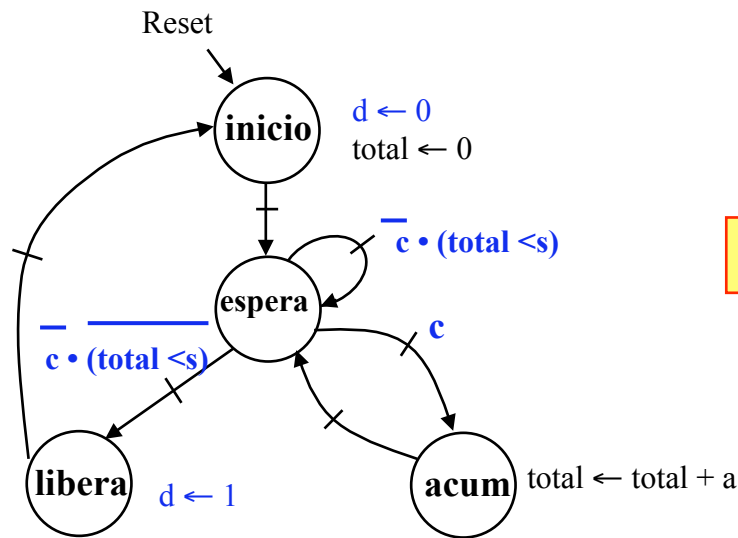
**OBS:** em primeira aproximação, pode-se assumir que cada variável interna é mapeada para um registrador.



# FSM como Bloco de Controle

## ▶ Projetando um Sistema Digital

### Exemplo 1: Passo 2 (projeto do BO)



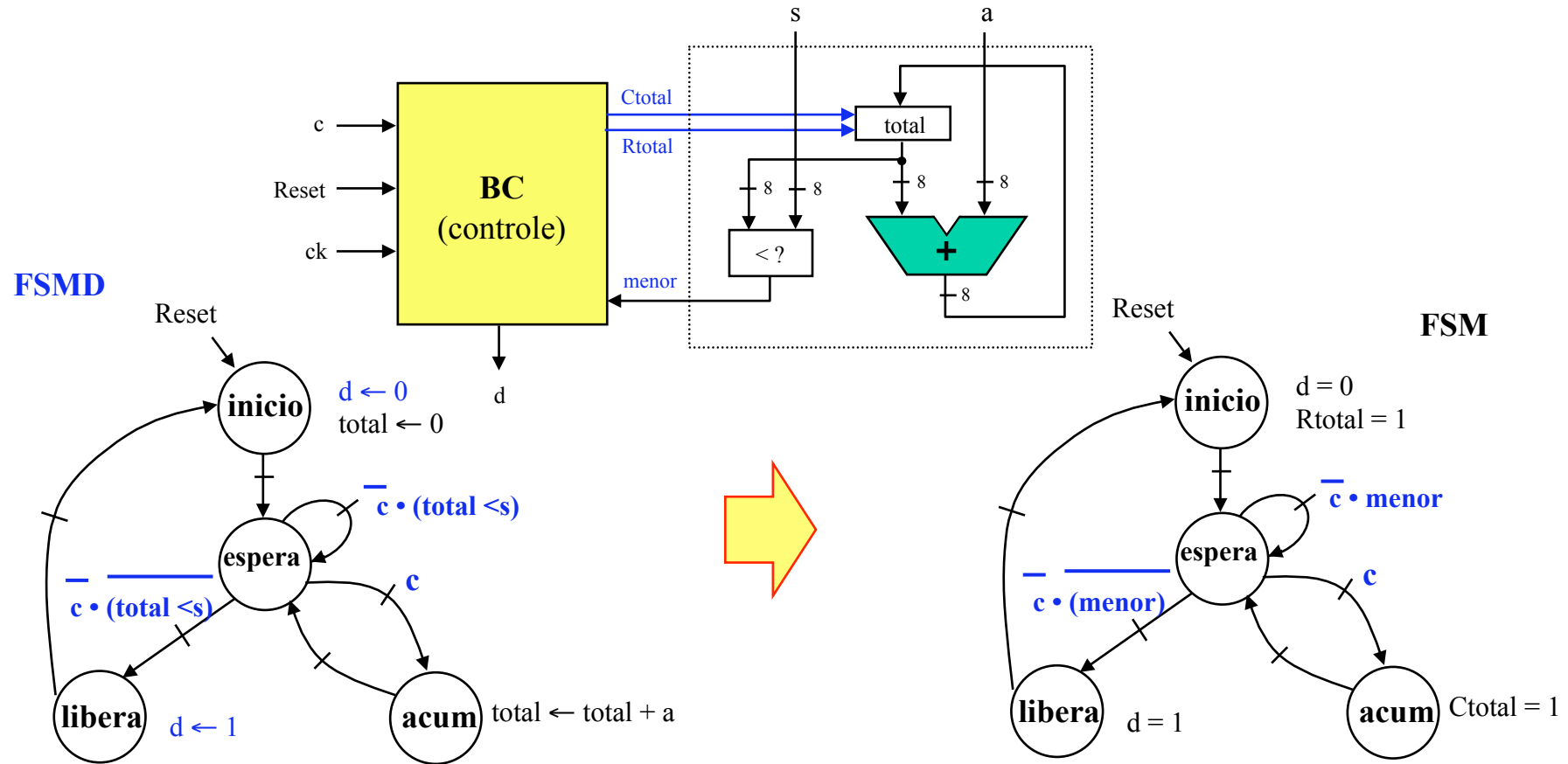
### Convenção:

- Todos os registradores são cadenciados pelo sinal de relógio (ck). Entretanto, para simplificar o desenho, o sinal de relógio está omitido.

# FSM como Bloco de Controle

## ▶ Projetando um Sistema Digital

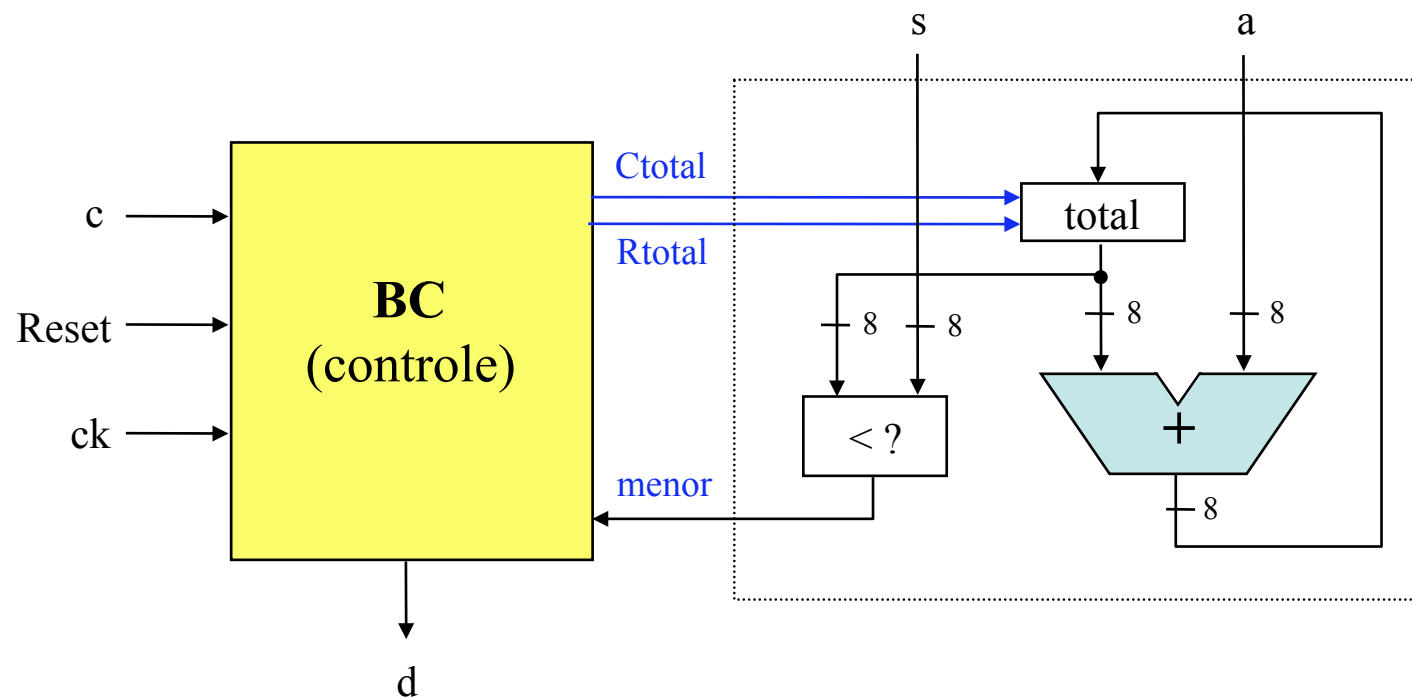
### Exemplo 1: Passo 4 (Derivando a FSM a partir do BO e da FSMD)



# FSM como Bloco de Controle

## ▶ Projetando um Sistema Digital

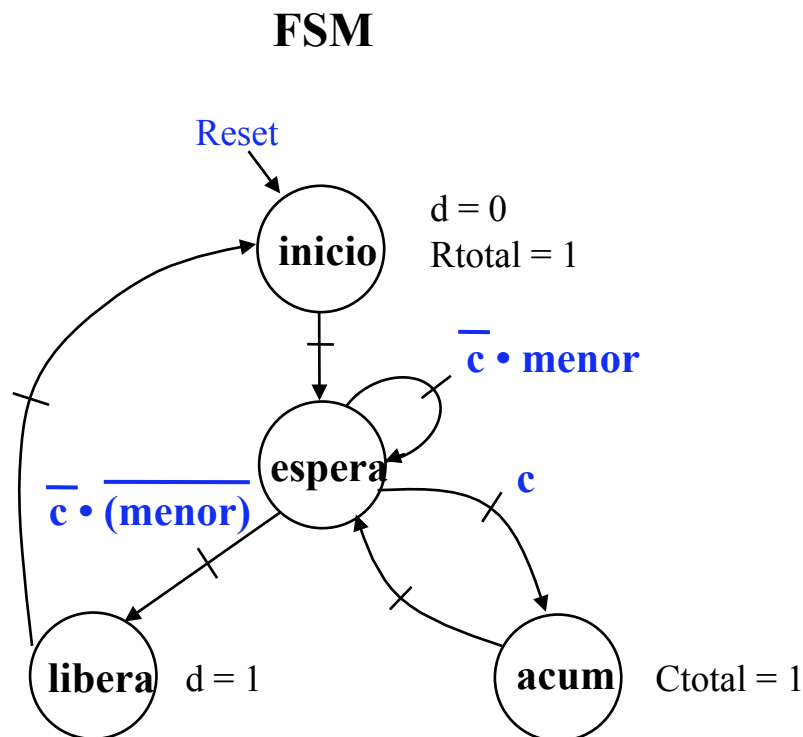
### Exemplo 1: Diagrama BO/BC



# FSM como Bloco de Controle

## ▶ Projetando um Sistema Digital

### Exemplo 1: Projeto do BC



**Tabela de Transição de Estados**

Estado atual	c	menor	Próximo estado
início	X	X	espera
espera	0	0	libera
espera	0	1	espera
espera	1	X	acum
acum	X	X	espera
libera	X	X	início

# FSM como Bloco de Controle

## ▶ Projetando um Sistema Digital

### Exemplo 1: Projeto do BC

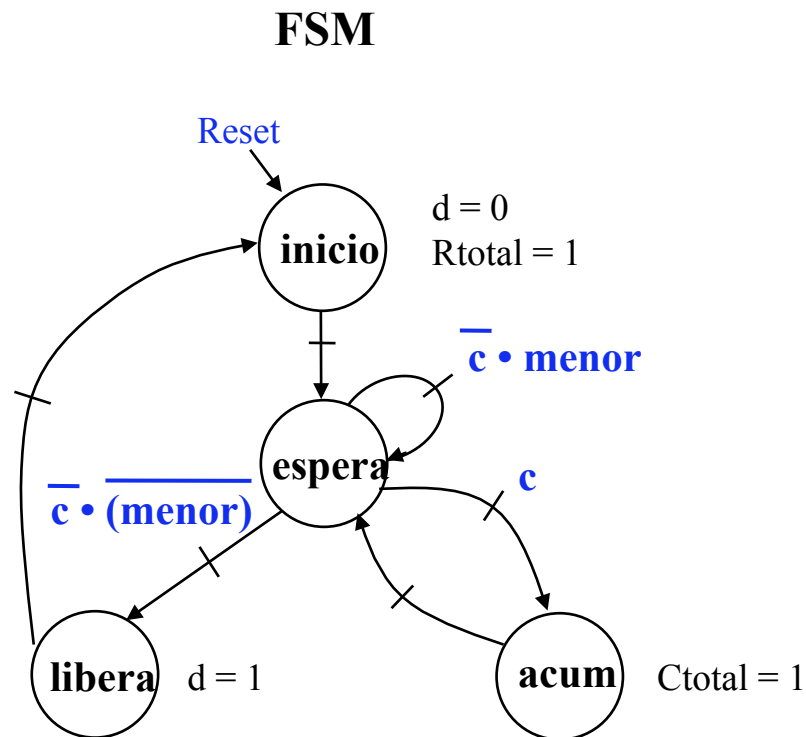


Tabela de Saídas

Estado	$R_{total}$	$C_{total}$	$d$
início	1	0	0
espera	0	0	0
acum	0	1	0
libera	0	0	1

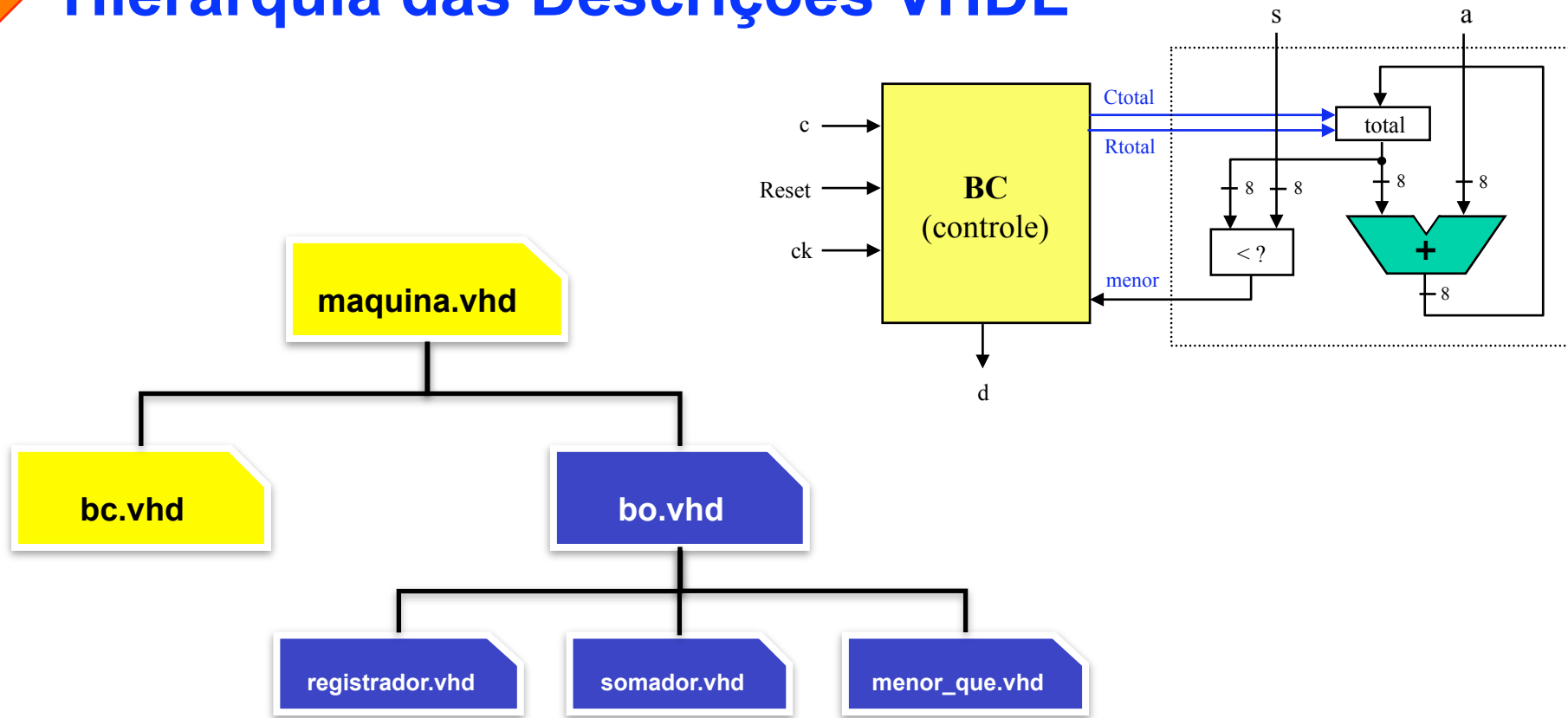
#### Importante:

- O sinal de carga de um registrador só deve estar ativado (=1) quando este for carregado com um novo valor! Caso contrário, o sinal de carga deve permanecer em desativado (=0).



# FSM como Bloco de Controle

## ► Hierarquia das Descrições VHDL



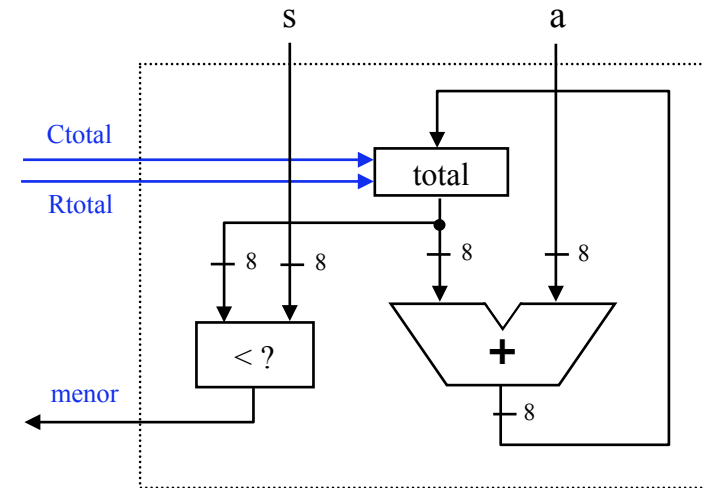
Arquivos VHDL disponíveis em <http://www.inf.ufsc.br/~guntzel/ine5406/aula12P/acompletar>

# FSM como Bloco de Controle

## ► Bloco Operativo

```
ENTITY bo IS
  PORT (clk, Ctotal, Rtotal : IN STD_LOGIC;
        s, a : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        menor : OUT STD_LOGIC);
END bo;
```

```
ARCHITECTURE estrutura OF bo IS
  -- components
  SIGNAL soma, total: STD_LOGIC_VECTOR (7 DOWNTO 0);
BEGIN
  reg: registrador PORT MAP (clk, Ctotal, Rtotal, soma, total);
  som: somador PORT MAP (total, a, soma);
  men: menor_que PORT MAP (total, s, menor);
END estrutura;
```



# FSM como Bloco de Controle

## ▶ Registrador

```
ENTITY registrador IS
  PORT (clk, carga, reset : IN STD_LOGIC;
        d : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        q : OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END registrador;
```

```
ARCHITECTURE estrutura OF registrador IS
BEGIN
```

```
  PROCESS(clk, reset)
```

```
  BEGIN
```

```
    IF(reset = '1') THEN
```

```
      q <= "00000000";
```

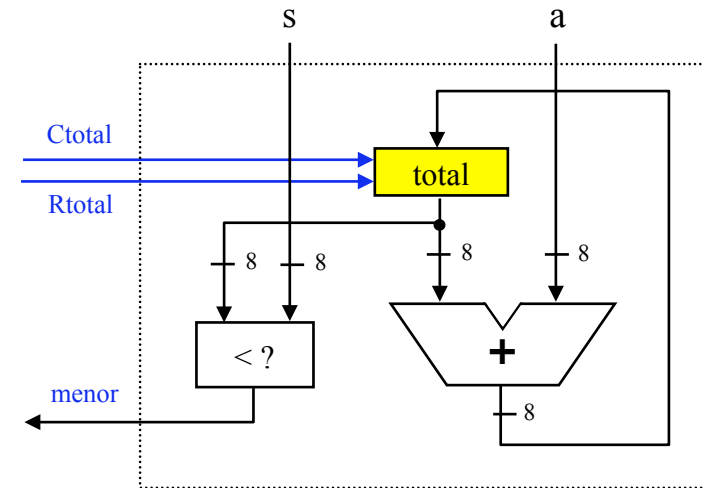
```
    ELSIF(clk'EVENT AND clk = '1' AND carga = '1') THEN
```

```
      q <= d;
```

```
    END IF;
```

```
  END PROCESS;
```

```
END estrutura;
```

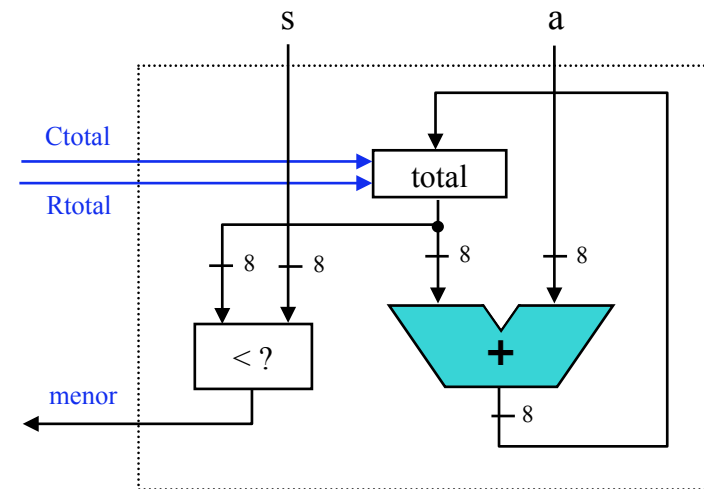


# FSM como Bloco de Controle

## ▶ Somador

```
ENTITY somador IS  
  PORT (a, b : IN STD_LOGIC_VECTOR(7 DOWNTO 0);  
        s : OUT STD_LOGIC_VECTOR(7 DOWNTO 0));  
END somador;
```

```
ARCHITECTURE estrutura OF somador IS  
BEGIN  
  s <= a + b;  
END estrutura;
```

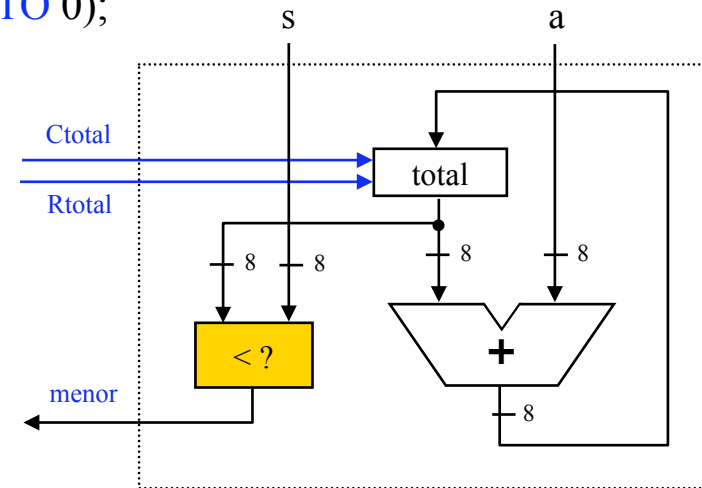


# FSM como Bloco de Controle

## ▶ Menor\_que

```
ENTITY menor_que IS  
  PORT (a, b : IN STD_LOGIC_VECTOR(7 DOWNTO 0);  
        menor : OUT STD_LOGIC);  
END menor_que;
```

```
ARCHITECTURE estrutura OF menor_que IS  
BEGIN  
  menor <= '1' WHEN A < B ELSE '0';  
END estrutura;
```



# FSM como Bloco de Controle

## ▶ O Bloco de Controle

```
ENTITY bc IS
    PORT (Reset, clk, c, menor : IN STD_LOGIC;
          d, Ctotal, Rtotal : OUT STD_LOGIC);
END bc;
ARCHITECTURE estrutura OF bc IS
    TYPE tipo_estado IS (INICIO, ESPERA, ACUM, LIBERA);
    SIGNAL estado: tipo_estado;
BEGIN
    -- Logica de proximo estado (e registrador de estado)
    PROCESS (clk, Reset)
    BEGIN
        -- completar
    END PROCESS;

    -- Logica de saida
    PROCESS (estado)
    BEGIN
        CASE estado IS
            -- completar
        END CASE;
    END PROCESS;
END estrutura;
```

Estado atual	c	menor	Próximo estado
início	X	X	espera
espera	0	0	libera
espera	0	1	espera
espera	1	X	acum
acum	X	X	espera
libera	X	X	início

Estado	Rtotal	Ctotal	d
início	1	0	0
espera	0	0	0
acum	0	1	0
libera	0	0	1

# FSM como Bloco de Controle

## ▶ Máquina de Vendas

ENTITY maquina IS

PORT (Reset, clk, c : IN STD\_LOGIC;

s, a : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

d : OUT STD\_LOGIC);

END maquina;

ARCHITECTURE estrutura OF maquina IS

COMPONENT bo IS

-- completar

END COMPONENT;

COMPONENT bc IS

-- completar

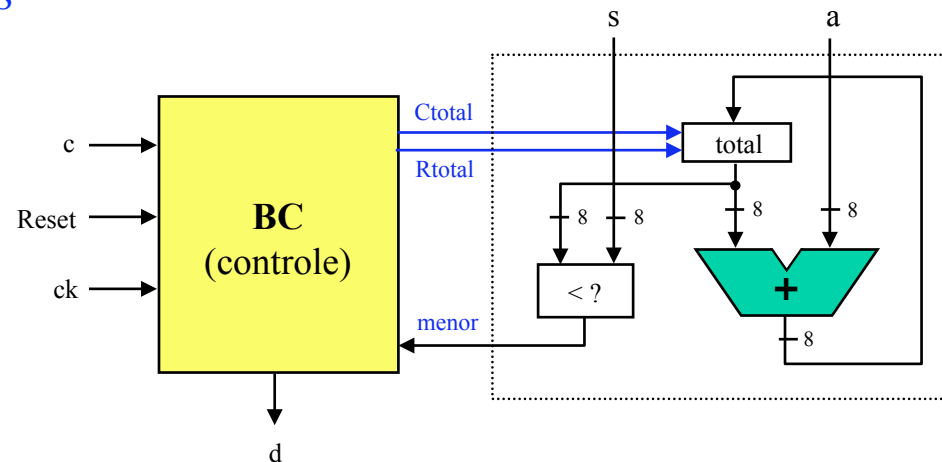
END COMPONENT;

SIGNAL -- completar

BEGIN

-- completar

END estrutura;



# FSM como Bloco de Controle

---

## ▶ Simulação

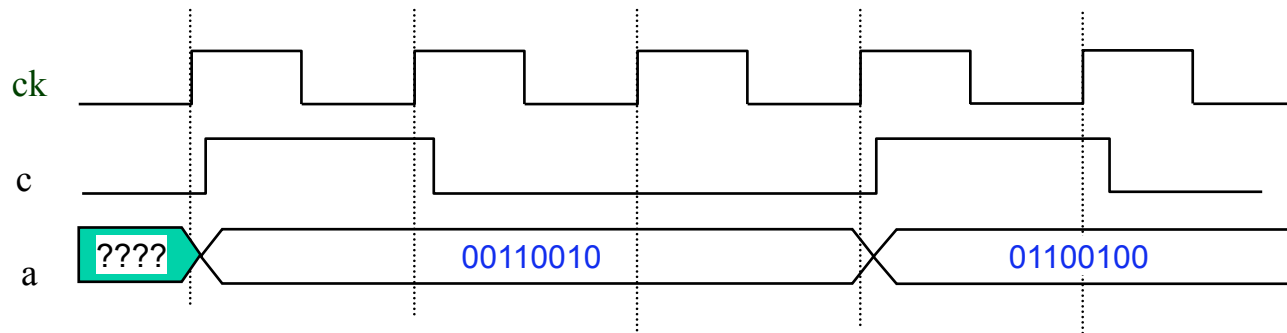
1. Na pasta Meus\_documentos, criar uma pasta com o seu nome (p. ex., “Paulo”). Na pasta “Paulo”, criar uma pasta com nome de “maquina”.
2. Acessar o sítio “[www.inf.ufsc.br/~guntzel/ine5406/aula12P/acompletar](http://www.inf.ufsc.br/~guntzel/ine5406/aula12P/acompletar)” e baixar para a pasta os arquivos “.vhd”
3. Abrir o Quartus II e criar um projeto na pasta “maquina”, selecionando “maquina.vhd” como toplevel. Escolher o dispositivo FPGA EP2C35F672C6 e selecionar o ModelSim-Altera como EDA Simulation Tool.
4. Completar os arquivos “[bc.vhd](#)” e “[maquina.vhd](#)”.
5. Compilar o projeto criado.
6. A partir do Quartus II, chame o ModelSim-Altera e inicie uma simulação com atrasos (“Gate-level Simulation”).
7. Crie um arquivo de estímulos, nomeando-o “estimulos.do”. Preparar os estímulos seguindo as especificações da próxima transparência.



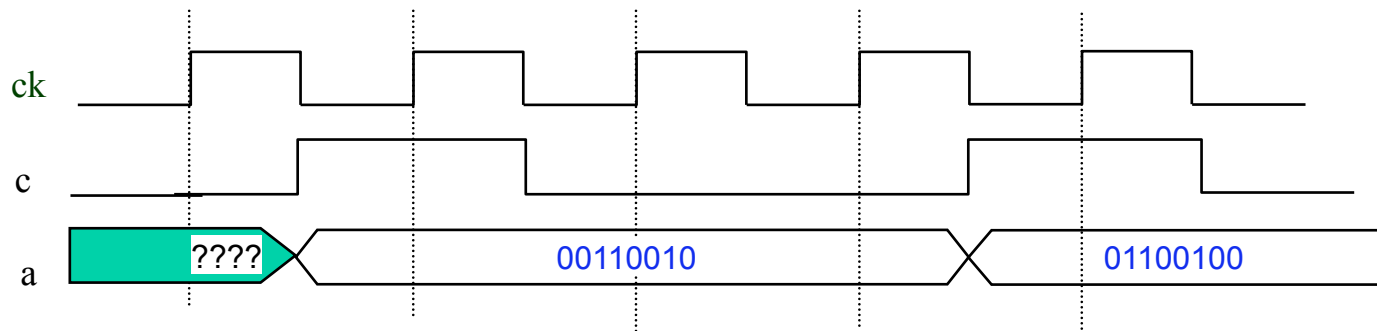
# FSM como Bloco de Controle

## ▶ Simulação

Formas de onda da especificação (aula teórica):



Formas de onda para o ModelSim-Altera (para aula prática):

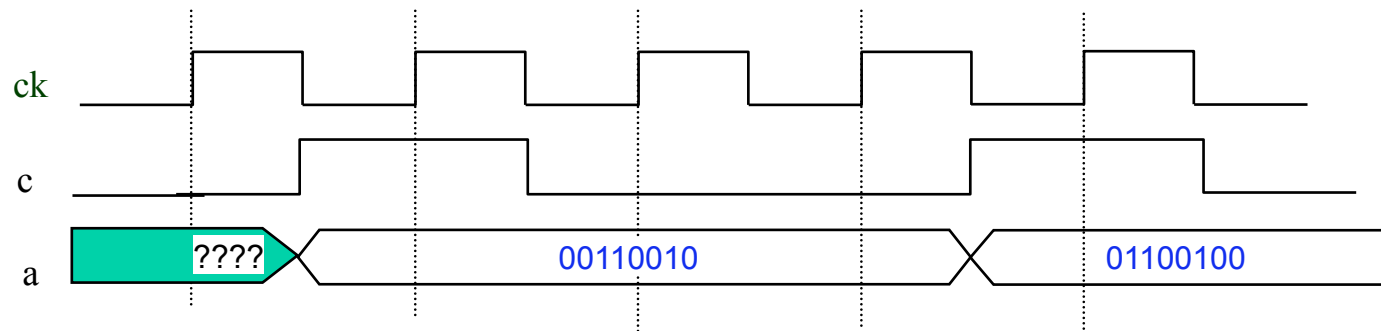


# FSM como Bloco de Controle

## ▶ Simulação

- Escolher um período de relógio maior que o atraso crítico do circuito “maquina”. Exemplo: 10 ns
- Fixar a entrada “s” em um valor, que representa o preço unitário do item vendido. Por exemplo: 160 (centavos)
- Simular sequências de moedas sendo inseridas. Por exemplo:
  - 100, 50, 25
  - 100, 50, 60

Formas de onda para o ModelSim-Altera (para aula prática):



# FSM como Bloco de Controle

---

## ▶ Simulação

- Deixar ao menos 5 ciclos de relógio entre cada nova moeda inserida
- Lembrar que “a” fica estável até que uma nova moeda seja inserida (ver slides da aula 12T)