



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



Sistemas Digitais

INE 5406

Aula 8-T

4. Projeto de Sistemas Digitais no Nível RT. Classificação dos Sistemas Digitais. O modelo BO / BC (*datapath* x controle). Componentes do Nível RT. Método de Projeto no Nível RT. Exemplo.

Prof. José Luís Güntzel
guntzel@inf.ufsc.br

www.inf.ufsc.br/~guntzel/ine5406/ine5406.html

4. Projeto de Sistemas Digitais no Nível RT

► Tipos e Características dos Sistemas Digitais

Classificação Segundo a Temporização

1. SDs Assíncronos:

- Não possuem um sinal de relógio para prover o cadenciamento das operações
- As operações são vistas como eventos encadeados (ou independentes)
- Possui um custo maior em termos de recursos, pois é necessário implementar protocolos de comunicação entre os blocos do sistema
- São tolerantes a variações na temporização
- São mais difíceis de projetar (difícil de garantir o funcionamento, após a fabricação)

4. Projeto de Sistemas Digitais no Nível RT

► Tipos e Características dos Sistemas Digitais

Classificação Segundo a Temporização

2. SDs Síncronos:

- Utilizam um sinal de relógio para prover o cadenciamento das operações
- O funcionamento é quebrado em passos denominados operações
- Cada operação geralmente leva um ciclo de relógio para ser realizada, mas há esquemas alternativos (ex.: *chaining*, *pipelining*)
- Em um ciclo de relógio uma ou mais operações podem ser realizadas simultaneamente
- As técnicas de projeto existentes permitem abstrair-se detalhes do comportamento

4. Projeto de Sistemas Digitais no Nível RT

Tipos e Características dos Sistemas Digitais

Classificação Segundo a Aplicação

1. Sistemas Digitais de Aplicação Específica (ASICs)
2. Sistemas Digitais de Propósito Geral
3. Sistemas Digitais Programáveis para uma Classe de Aplicações

4. Projeto de Sistemas Digitais no Nível RT

► Tipos e Características dos Sistemas Digitais

Classificação Segundo a Aplicação

1. Sistemas Digitais de Aplicação Específica (ASICs):

- Realizam somente um algoritmo (ou parte de um algoritmo)
- Oferecem pouca ou nenhuma programabilidade (i.e., alteração da funcionalidade)
- O projeto é feito de modo a otimizar a execução do algoritmo implementado, visando a **aplicação específica** (máximo desempenho com o mínimo custo e eventualmente, mínimo consumo de energia)
- Exemplos: codificadores/decodificadores de imagens estáticas (jpeg etc) ou dinâmicas (mpeg, H.264/AVC etc)

4. Projeto de Sistemas Digitais no Nível RT

► Tipos e Características dos Sistemas Digitais

Classificação Segundo a Aplicação

2. Sistemas Digitais de Propósito Geral:

- Podem ser programados para executar (virtualmente) **qualquer algoritmo**
- Para tanto, são projetados para realizar um **conjunto de instruções**
- São otimizados para realizar o conjunto de instruções para o qual são projetados (e não um algoritmo ou uma classe de algoritmo)
- Exemplos: microprocessadores (CPUs)

4. Projeto de Sistemas Digitais no Nível RT

► Tipos e Características dos Sistemas Digitais

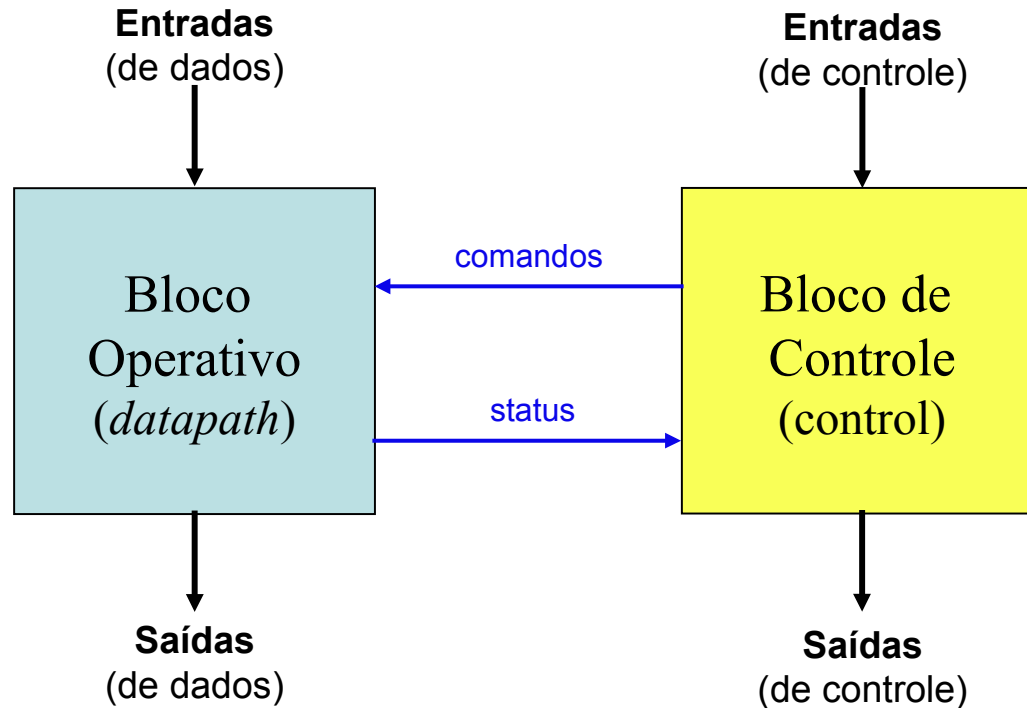
Classificação Segundo a Aplicação

3. Sistemas Digitais Programáveis para uma Classe de Aplicações:

- Podem ser programados para executar uma função ou um algoritmo pertencente a uma determinada classe.
- São projetados e otimizados para realizar um **conjunto de instruções** apropriado à classe de problema a que se destinam
- Exemplos: microcontroladores, DSPs (*Digital Signal Processors*) e GPUs (*Graphic Processing Units*)

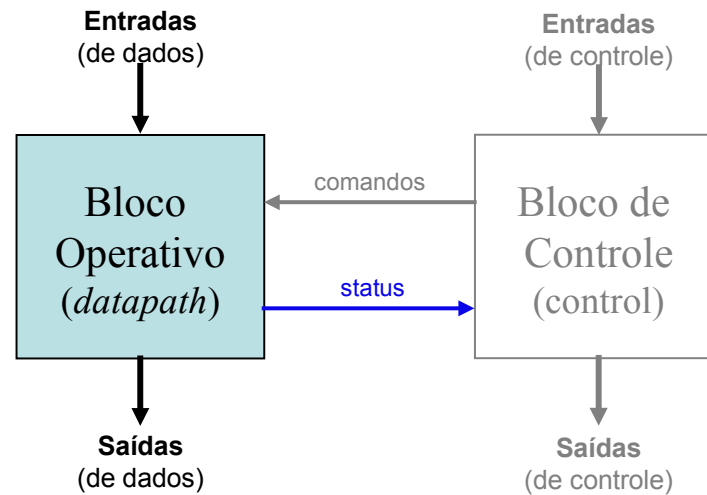
4. Projeto de Sistemas Digitais no Nível RT

▶ O Modelo Bloco Operativo / Bloco de Controle



4. Projeto de Sistemas Digitais no Nível RT

▶ O Modelo Bloco Operativo / Bloco de Controle

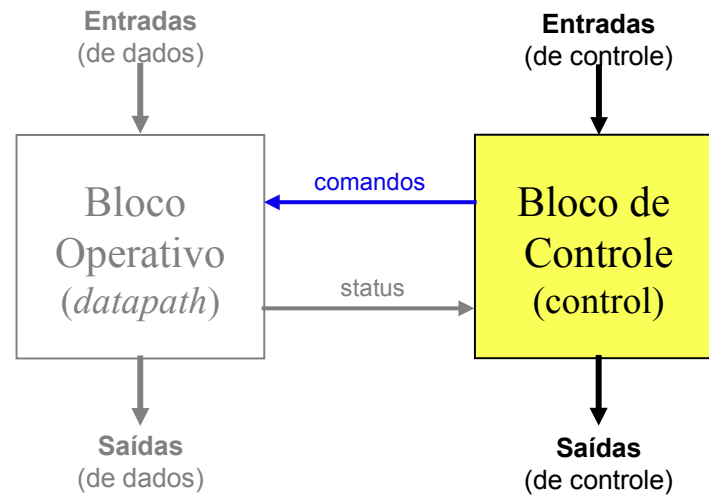


Bloco Operativo:

- 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 seqüência de operações a serem realizadas (às vezes são chamados de “*flags*”)

4. Projeto de Sistemas Digitais no Nível RT

▶ O Modelo Bloco Operativo / Bloco de Controle

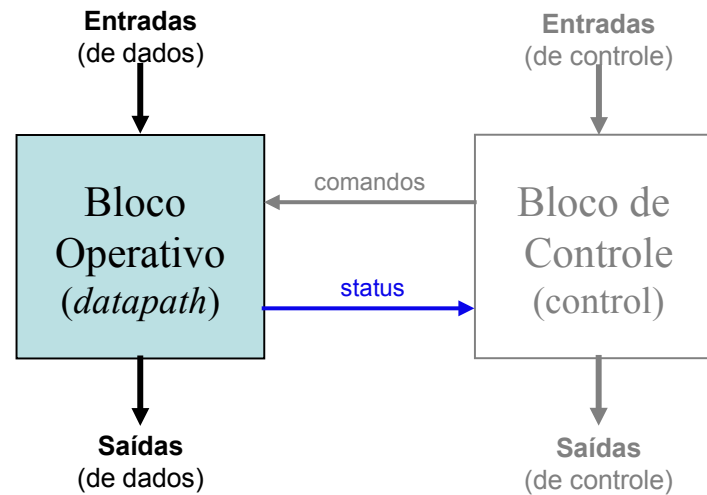


Bloco de Controle:

- 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, podendo ser desde um simples “iniciar” até um código de operação (“opcode”, dos processadores)
- Pode gerar uma ou mais saídas de controle para se comunicar com outros sistemas digitais (p. ex.: “done”, “bus request”, “ack”)

4. Projeto de Sistemas Digitais no Nível RT

► Os Componentes do Nível RT

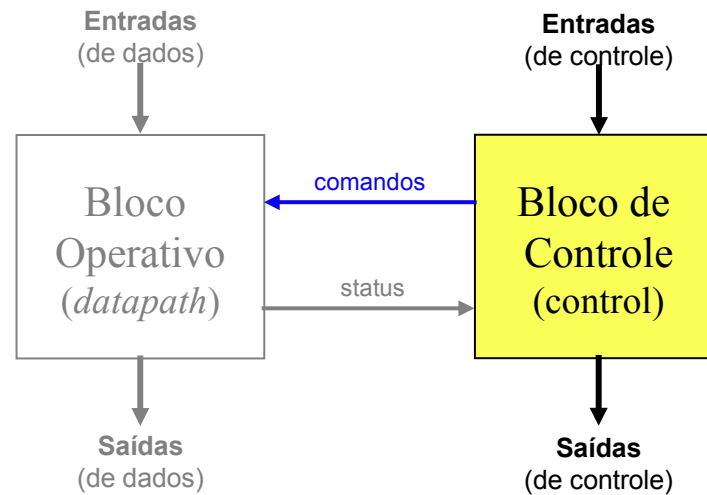


Bloco Operativo:

- Unidades Funcionais (UFs). Exemplos: somadores, subtratores, deslocadores, multiplicadores, UFs combinadas (somadores/subtratores, ULAs)
- Elementos de armazenamento: registradores, banco de registradores (vários registradores, mas com limitação de portas de entrada/saída), memórias RAM (geralmente, SRAM)
- Rede de interconexão: fios, multiplexadores, barramentos + *buffers tri-state*

4. Projeto de Sistemas Digitais no Nível RT

► Os Componentes do Nível RT

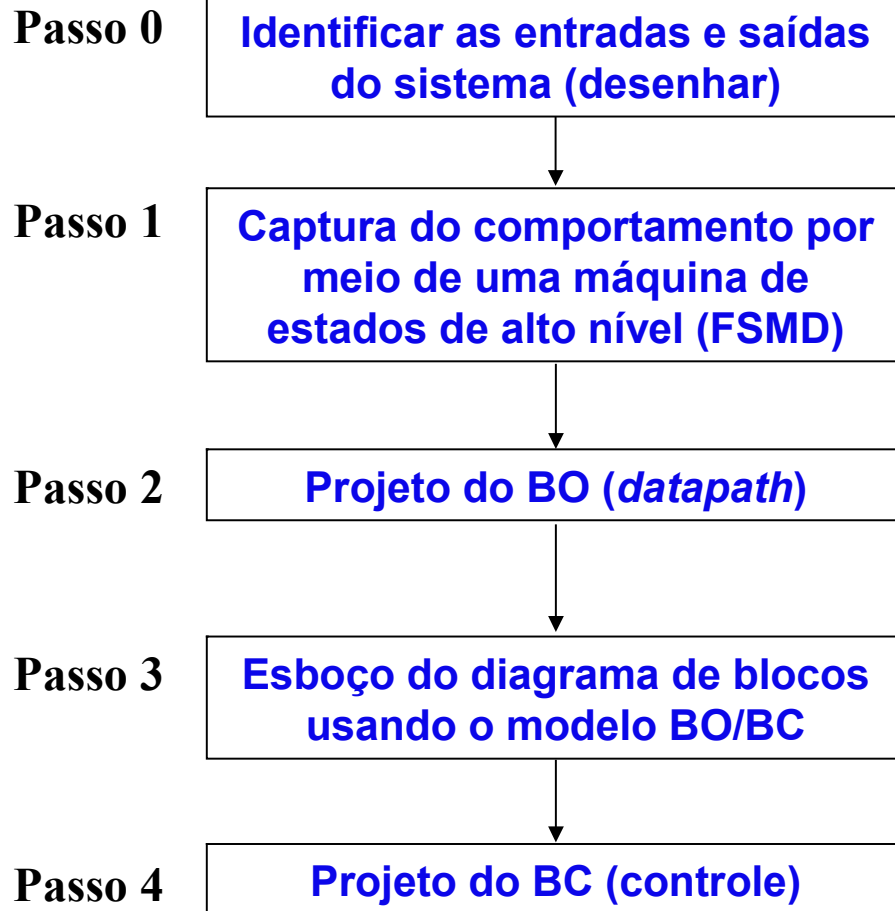


Bloco de Controle:

- Implementado por uma ou mais FSMs usando um dos seguintes métodos:
 - *Hardwired*: registrador de estados + circuitos lógicos ou
 - Usando ROM: registrador de estados + circuito comb. + um ou mais blocos ROM ou
 - Microprogramada: registrador-contador + circuito comb. + um ou mais blocos ROM (subcaso da anterior...)

4. Projeto de Sistemas Digitais no Nível RT

▶ Método de Projeto



4. Projeto de Sistemas Digitais no Nível RT

▶ Método de Projeto

Passo 0

Identificar as entradas e saídas do sistema (desenhar)

Passo 1

Captura do comportamento por meio de uma máquina de estados de alto nível (FSMD)

Passo 2

Projeto do BO (*datapath*)

Passo 3

Esboço do diagrama de blocos usando o modelo BO/BC

Passo 4

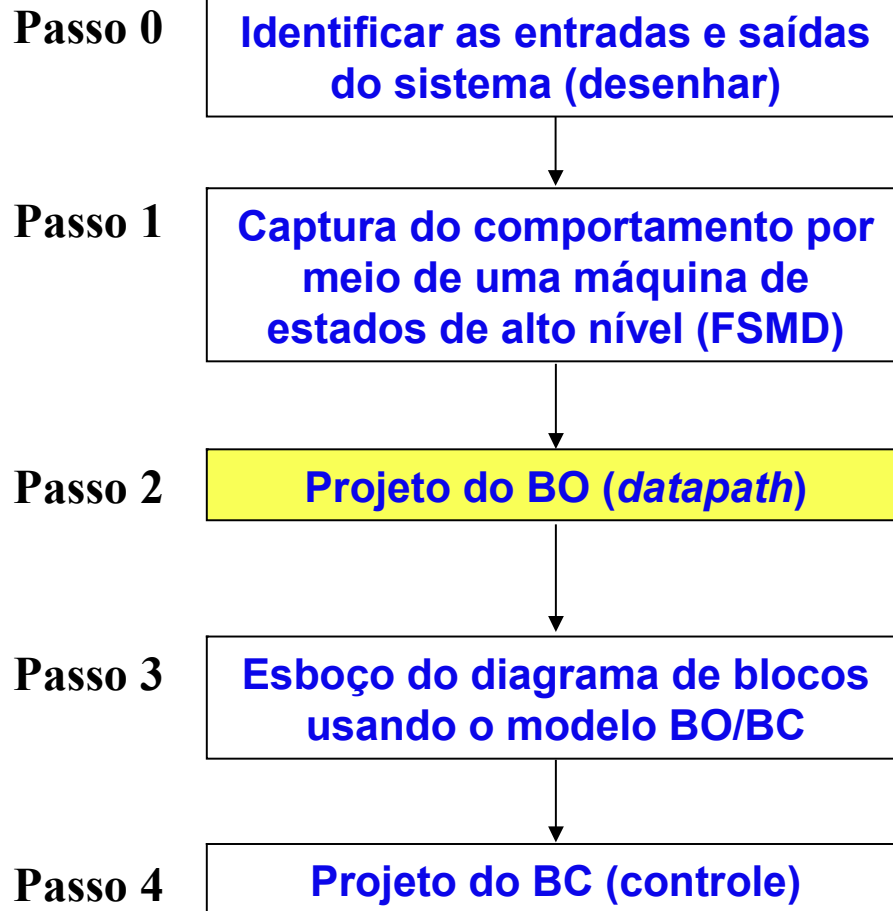
Projeto do BC (controle)

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

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

4. Projeto de Sistemas Digitais no Nível RT

▶ Método de Projeto



Analisando 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.

4. Projeto de Sistemas Digitais no Nível RT

▶ Método de Projeto

Passo 0

Identificar as entradas e saídas do sistema (desenhar)

Passo 1

Captura do comportamento por meio de uma máquina de estados de alto nível (FSMD)

Passo 2

Projeto do BO (*datapath*)

Passo 3

Esboço do diagrama de blocos usando o modelo BO/BC

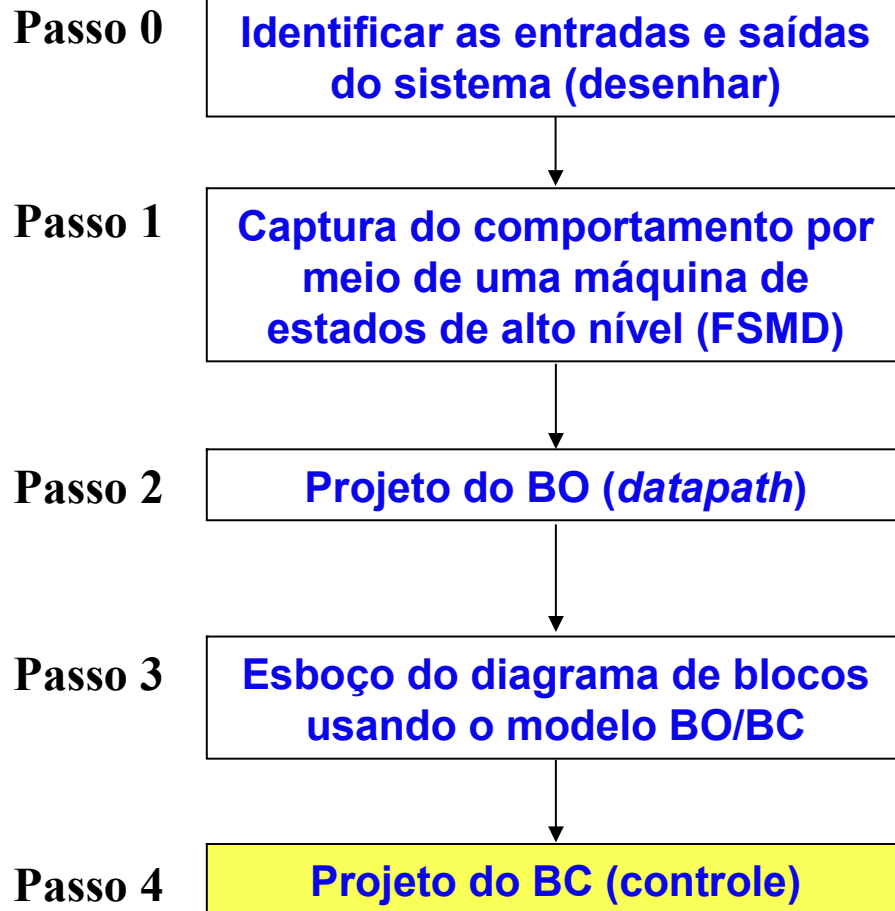
Passo 4

Projeto do BC (controle)

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

4. Projeto de Sistemas Digitais no Nível RT

▶ Método de Projeto



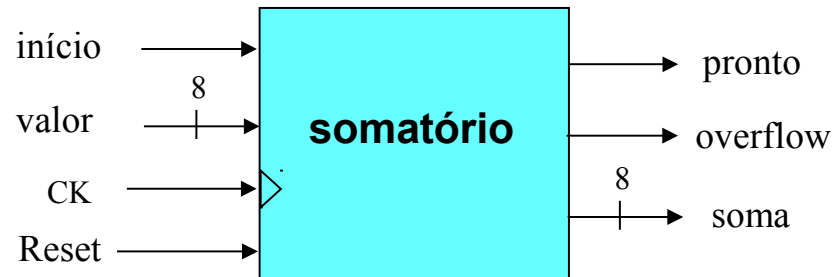
- A partir da FSMD inicial e observando os nomes dos sinais definidos no passo 3, projetar a FSM que deve controlar o BO projetado (conforme visto na parte anterior desta disciplina).

4. Projeto de Sistemas Digitais no Nível RT

▶ **Projetando um Sistema Digital**

Exemplo 1: cálculo de um somatório

Necessita-se de um sistema digital capaz de calcular o somatório de 4 número binários inteiros sem sinal, cada um deles com 8 bits. Este sistema digital, doravante denominado de “somatório”, possui uma entrada de relógio (“CK”), uma entrada de reset assíncrono (“Reset”), uma entrada de dados com 8 bits (“valor”), uma entrada de controle denominada “início”, duas saídas de controle (“pronto” e “overflow”) e uma saída de dados de 8 bits (“soma”).

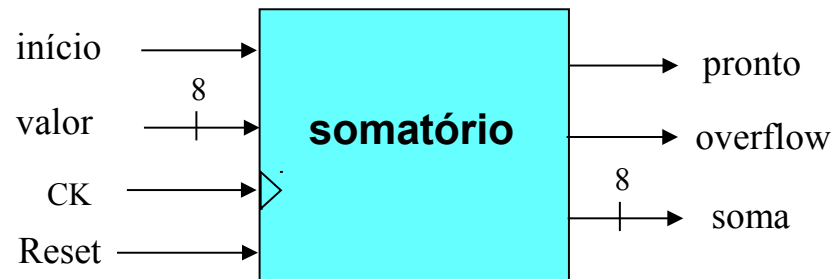


4. Projeto de Sistemas Digitais no Nível RT

▶ **Projetando um Sistema Digital**

Exemplo 1: cálculo de um somatório

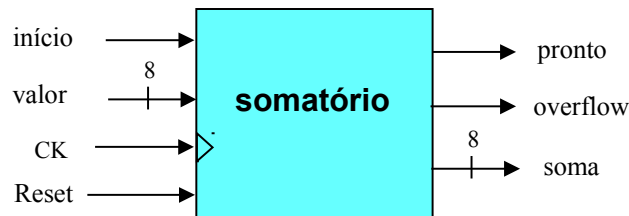
Funcionamento: ao receber um sinal de início (início = 1), “somatório” passa a ler da entrada “valor” os 4 valores a serem somados. (Suponha que os valores sejam fornecidos no momento adequado.) Ao mesmo tempo que os valores são lidos, seu somatório vai sendo calculado. Uma vez calculado o somatório, as saídas “soma”, “pronto” e “overflow” são atualizadas, ficando estáveis por, no mínimo um ciclo de relógio, antes de “somatório” voltar para o estado de reset.



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

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



Nitidamente, este sistema digital deve realizar o seguinte algoritmo (em hardware):

Início

$\text{acum} \leftarrow 0;$

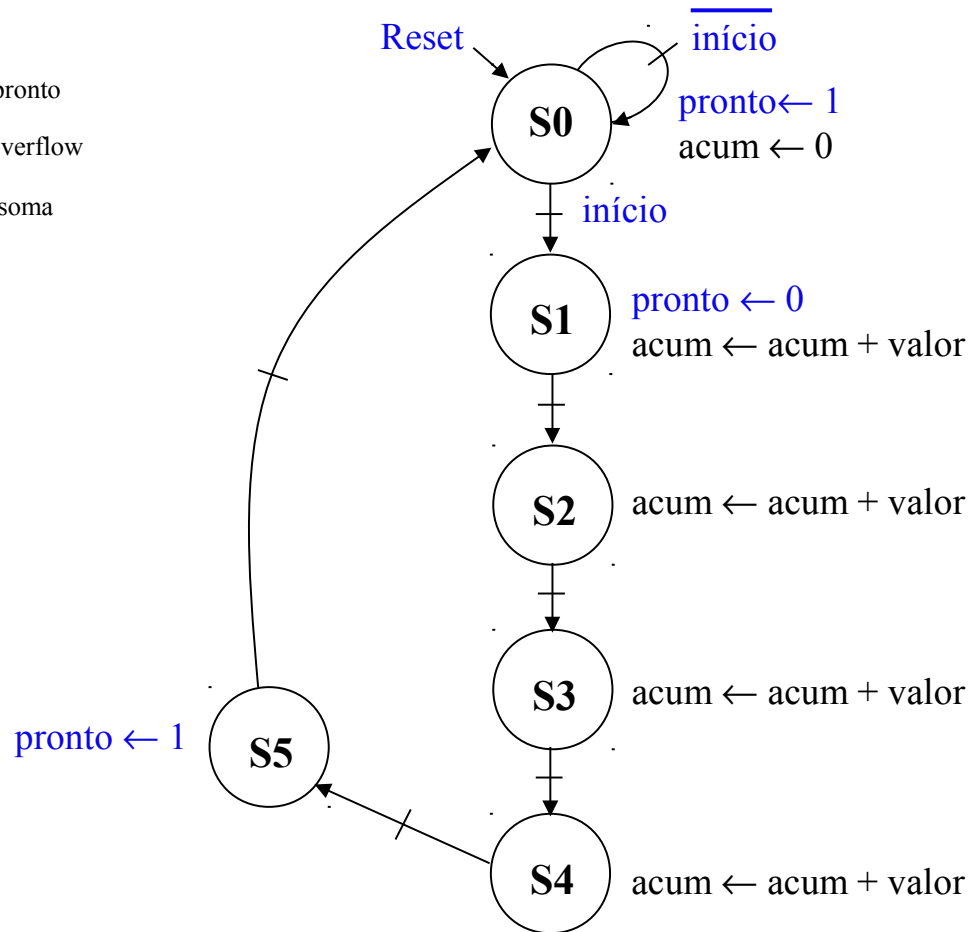
Fazer 4 vezes

{

$\text{acum} \leftarrow \text{acum} + \text{valor};$

}

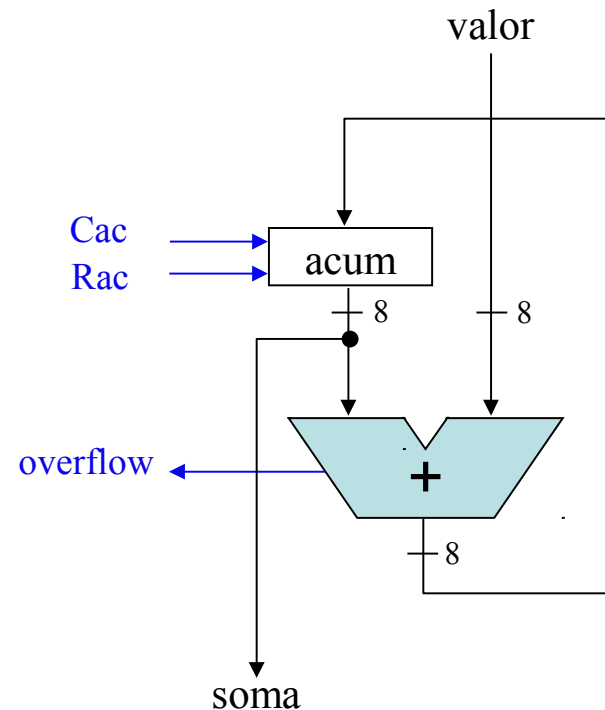
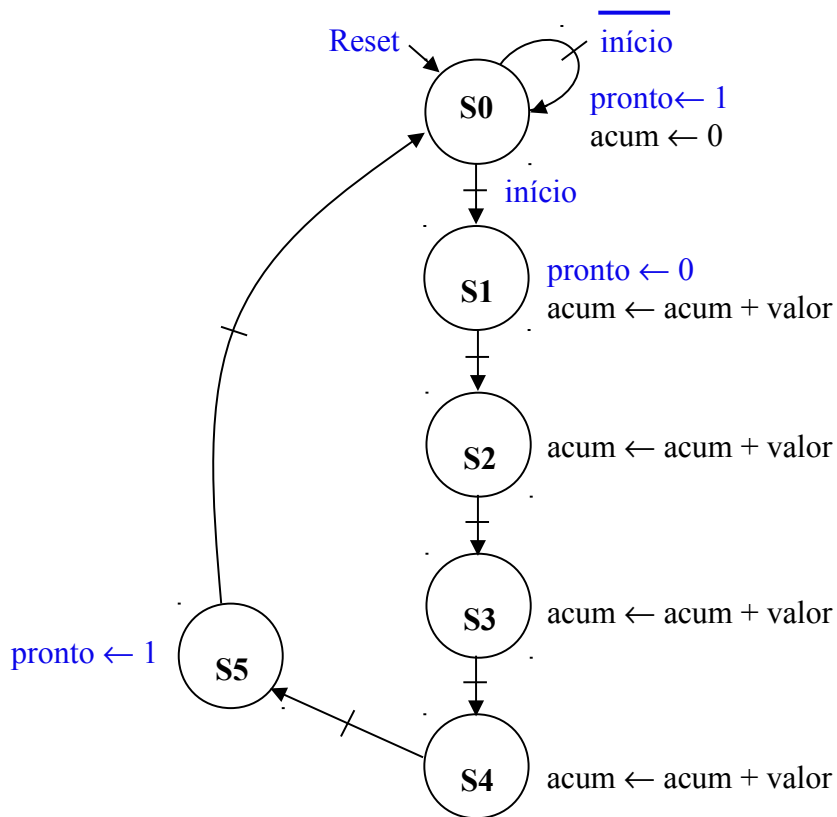
Fim



4. Projeto de Sistemas Digitais no Nível RT

▶ **Projetando um Sistema Digital**

Exemplo 1: Passo 2 (projeto do BO)

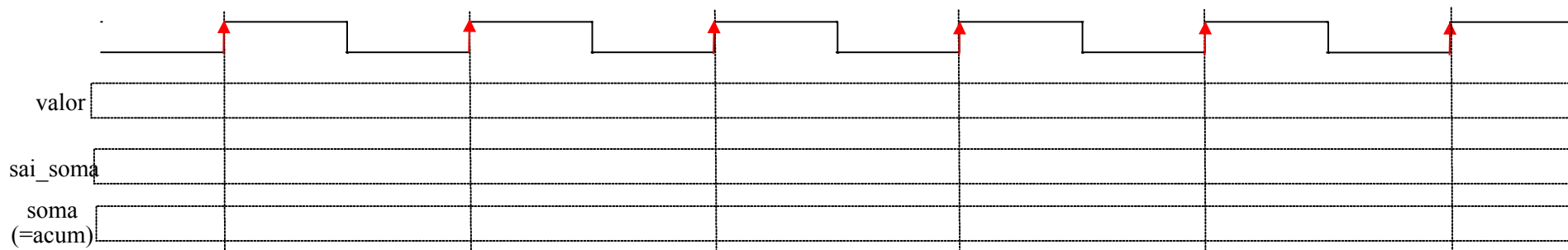
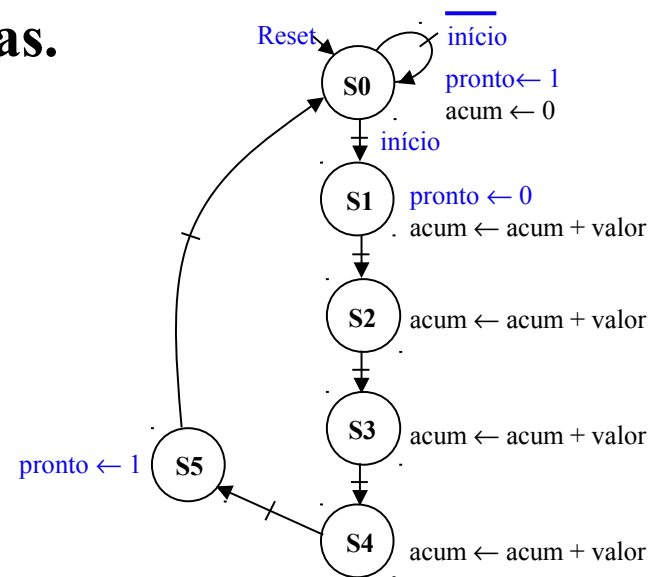
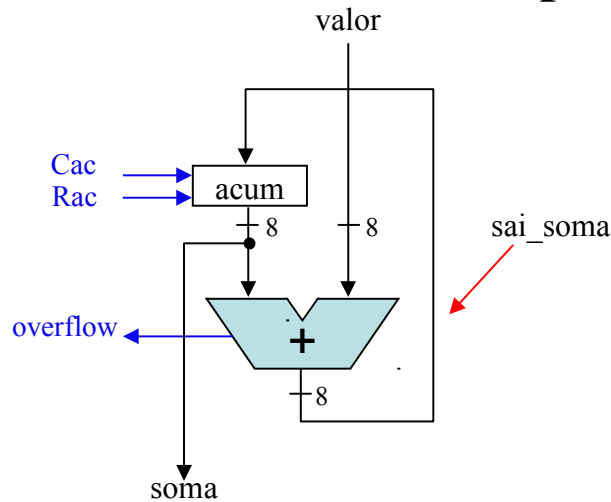


Este é um B.O. típico para o cálculo de um somatório.

4. Projeto de Sistemas Digitais no Nível RT

▶ **Projetando um Sistema Digital**

Exercício: ilustrar o comportamento do S.D. Do exemplo 1 por meio de formas de onda apropriadas.



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

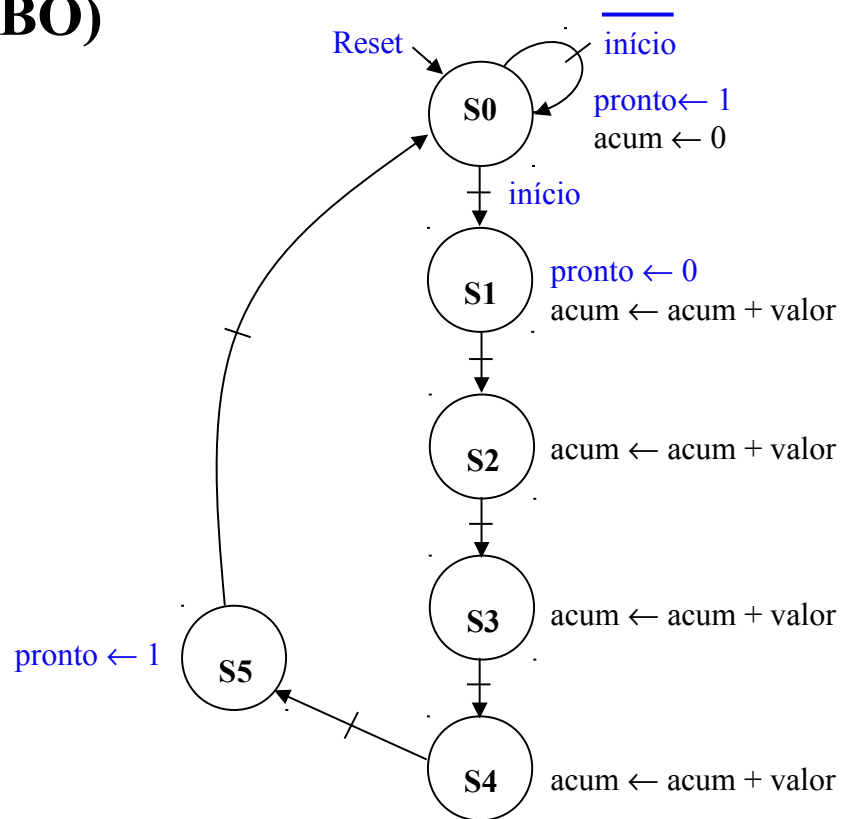
Exemplo 1: Passo 2 (projeto do BO)

Pergunta:

E se fosse necessário realizar a soma sequencial de 1024 parcelas, a FSM teria **1026** estados?

Resp.: sim!

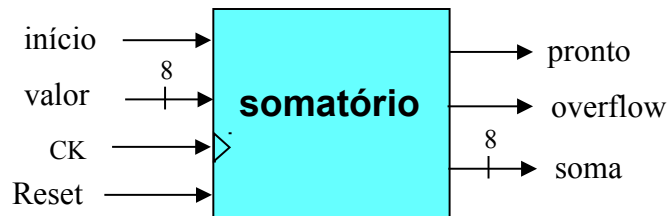
Então é melhor pensarmos em uma solução mais genérica...



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 1a: Passo 1 (captura do comportamento com FSMD)



Uma Solução mais genérica...

Início

$\text{acum} \leftarrow 0; \text{cont} \leftarrow 4;$

Enquanto ($\text{cont} \neq 0$) faça:

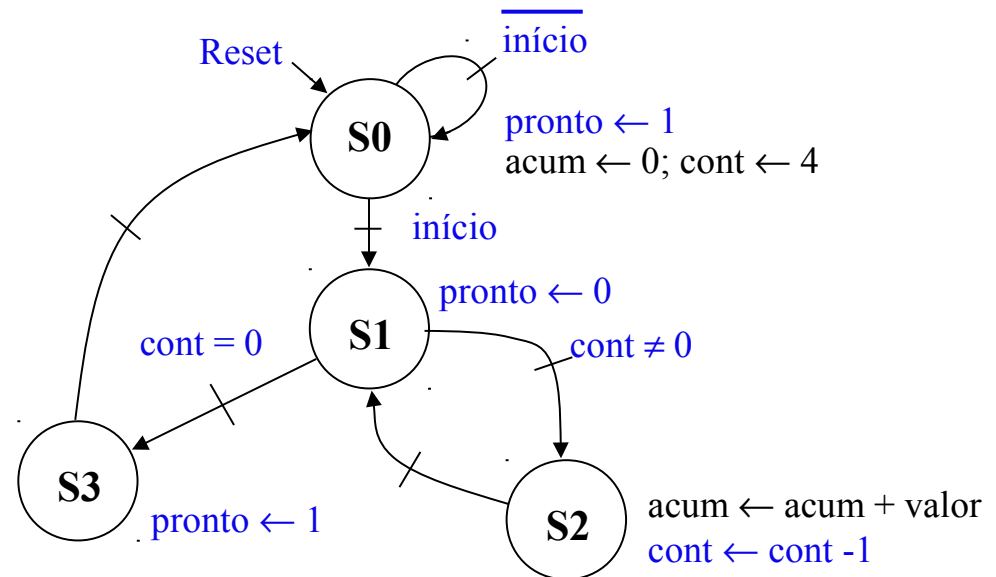
{

$\text{acum} \leftarrow \text{acum} + \text{valor};$

$\text{cont} \leftarrow \text{cont} - 1;$

}

Fim



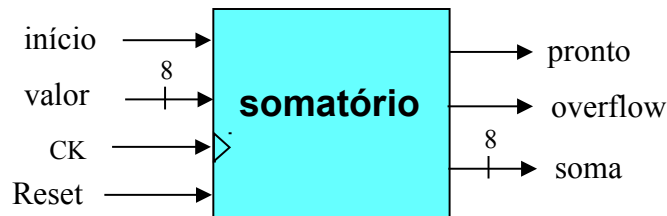
OBS:

- Estamos assumindo que o sinal "pronto" indica quando as saídas "soma" e "overflow" contêm os valores finais do processamento.

4. Projeto de Sistemas Digitais no Nível RT

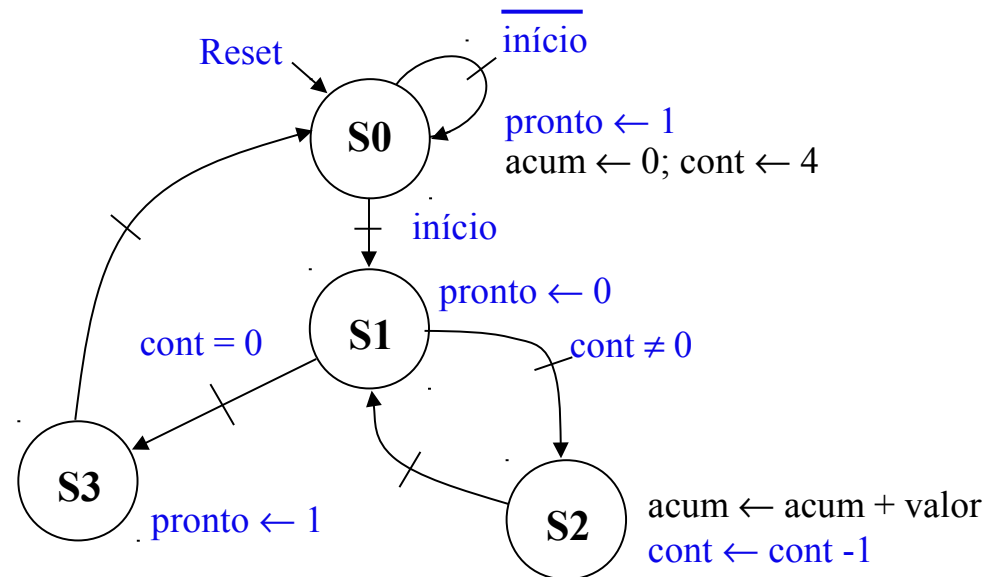
▶ Projetando um Sistema Digital

Exemplo 1a: Passo 2 (projeto do BO)



Questões:

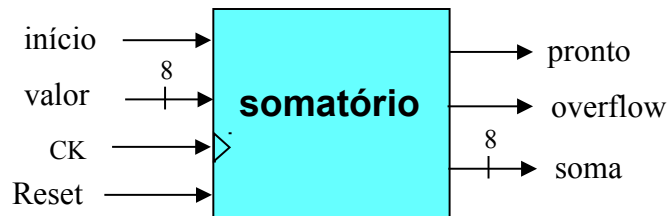
- Quais variáveis são usadas para armazenar dados?
- Apenas uma variável: “acum” (note que “valor” é uma entrada e “soma” é uma saída)
- Logo, teremos um registrador denominado “acum” para esta variável



4. Projeto de Sistemas Digitais no Nível RT

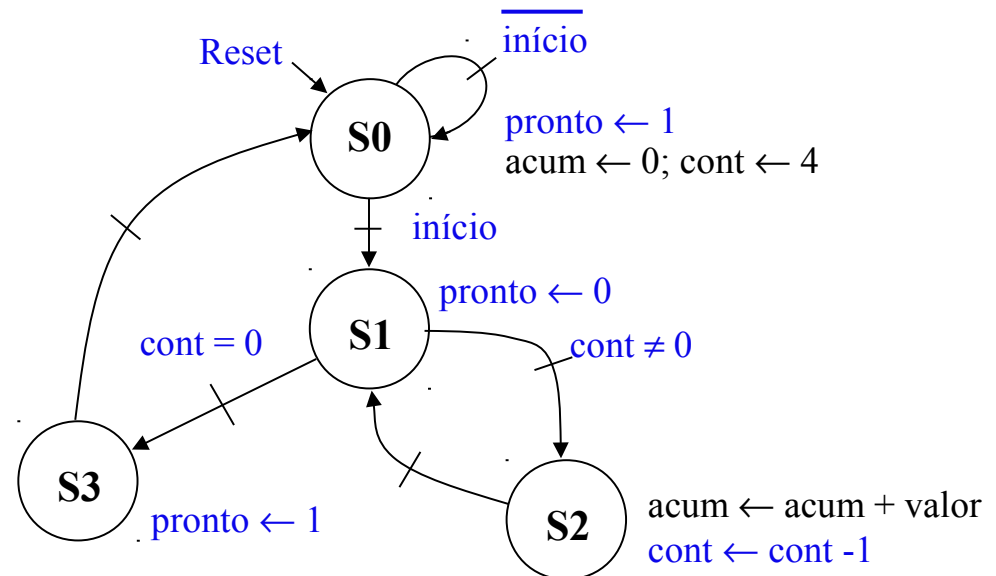
▶ Projetando um Sistema Digital

Exemplo 1a: Passo 2 (projeto do BO)



Porém:

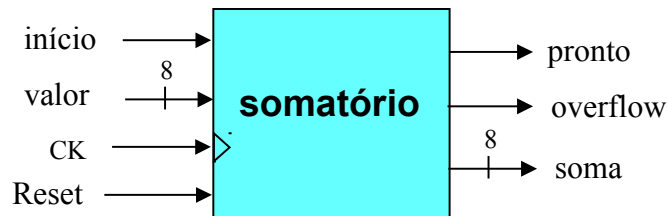
- Note que há um registrador que armazena informação de controle: "cont"



4. Projeto de Sistemas Digitais no Nível RT

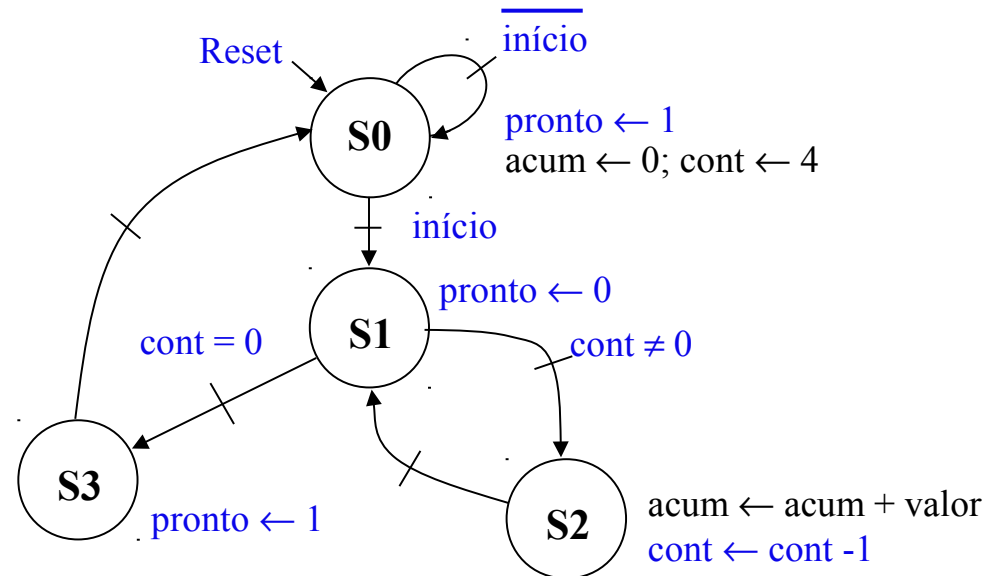
▶ Projetando um Sistema Digital

Exemplo 1a: Passo 2 (projeto do BO)



Questões:

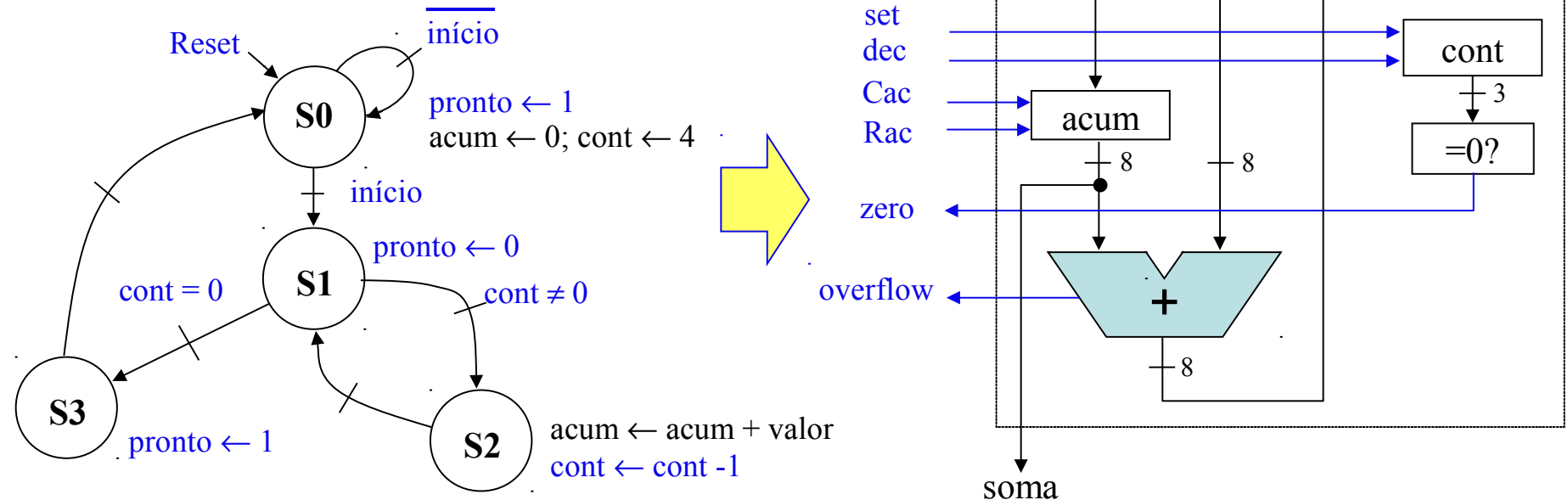
- Quais operações são realizadas sobre dados (incluindo-se as condições)?
- Uma adição para números de 8 bits ($\text{acum} \leftarrow \text{acum} + \text{valor}$)
- Um decremento ($\text{cont} \leftarrow \text{cont} - 1$). Iremos usar um contador-decrementador (mas poderíamos usar um subtrator ou um somador/subtrator...)
- Uma comparação com zero ($\text{cont}=0$).



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 1a: Passo 2 (projeto do BO)



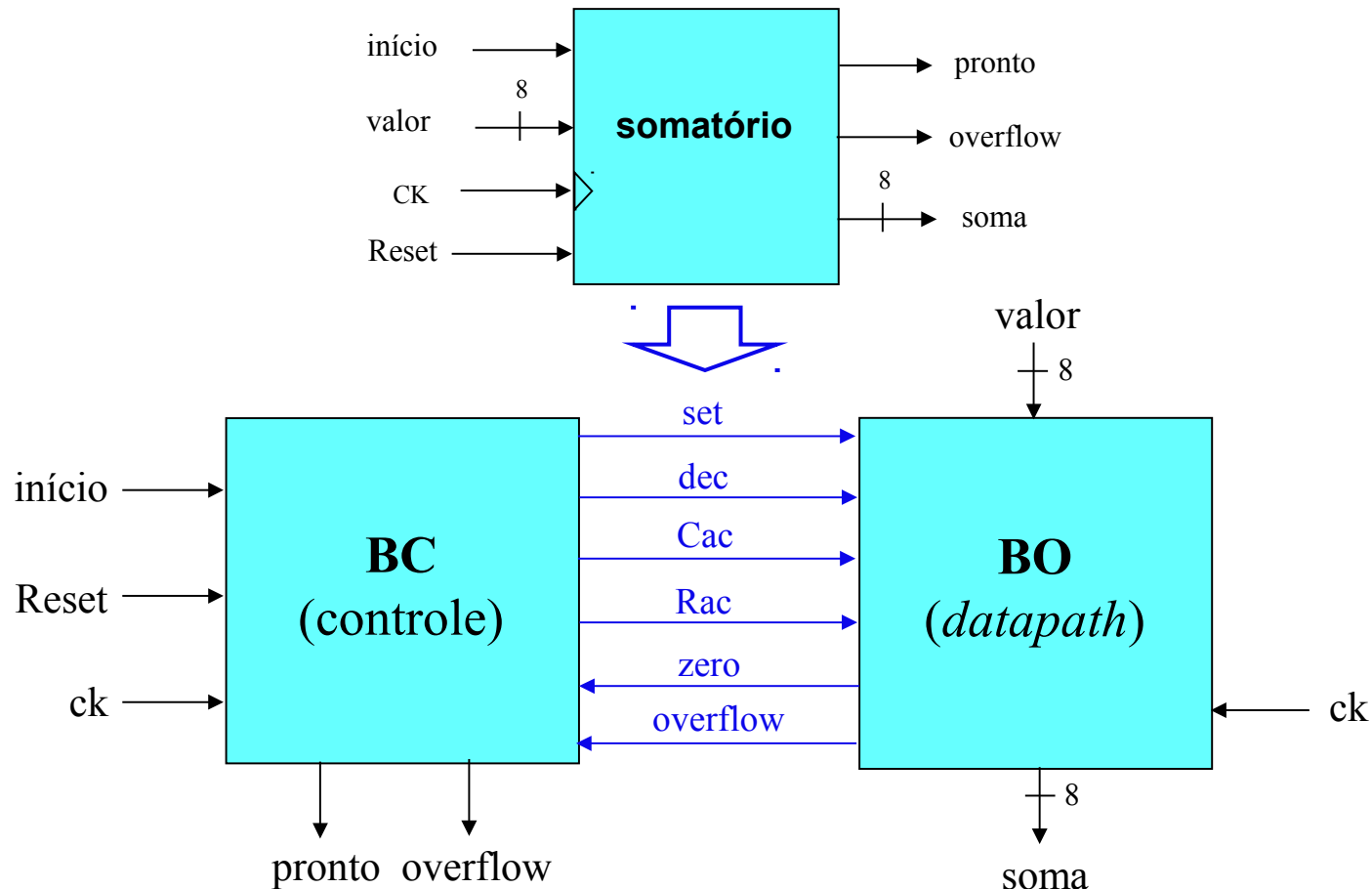
Convenção:

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

4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

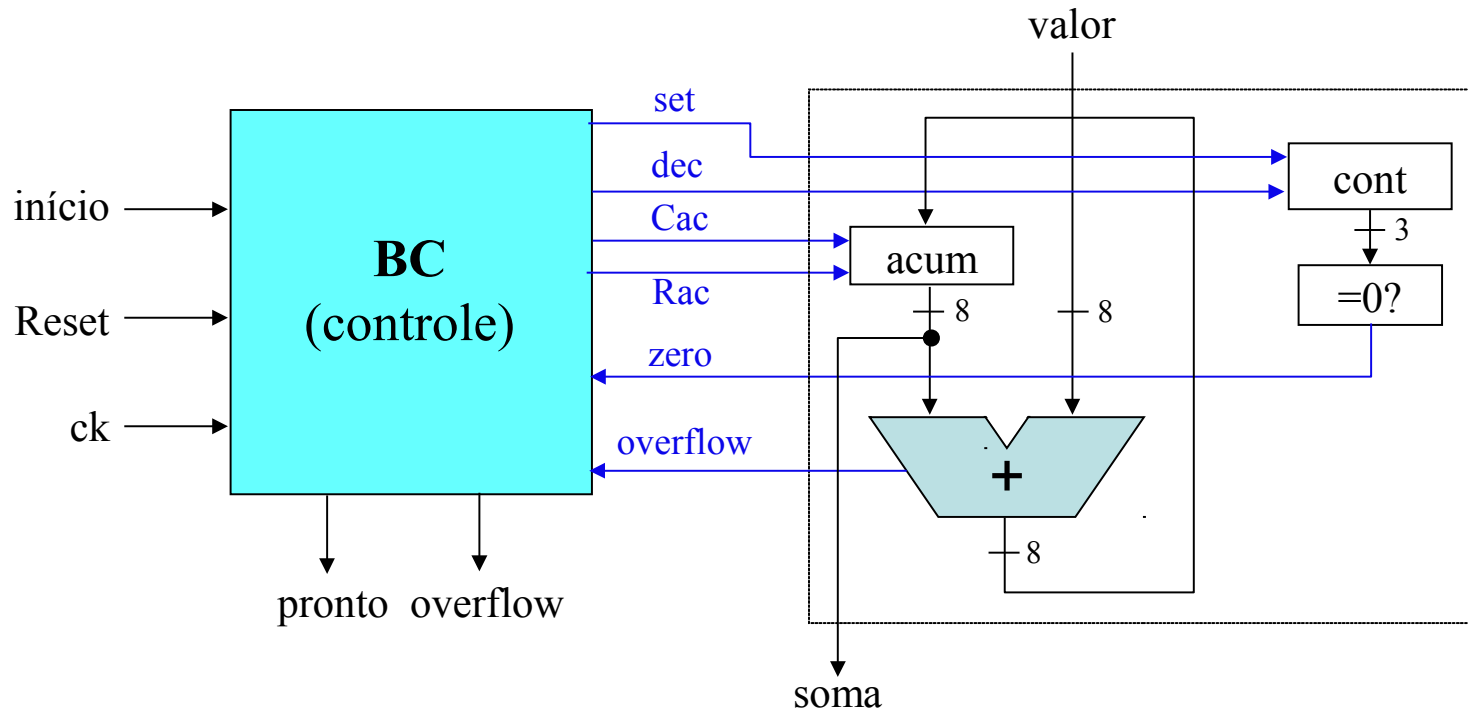
Exemplo 1a: Passo 3 (Esboçando o diagrama BO/BC)



4. Projeto de Sistemas Digitais no Nível RT

▶ **Projetando um Sistema Digital**

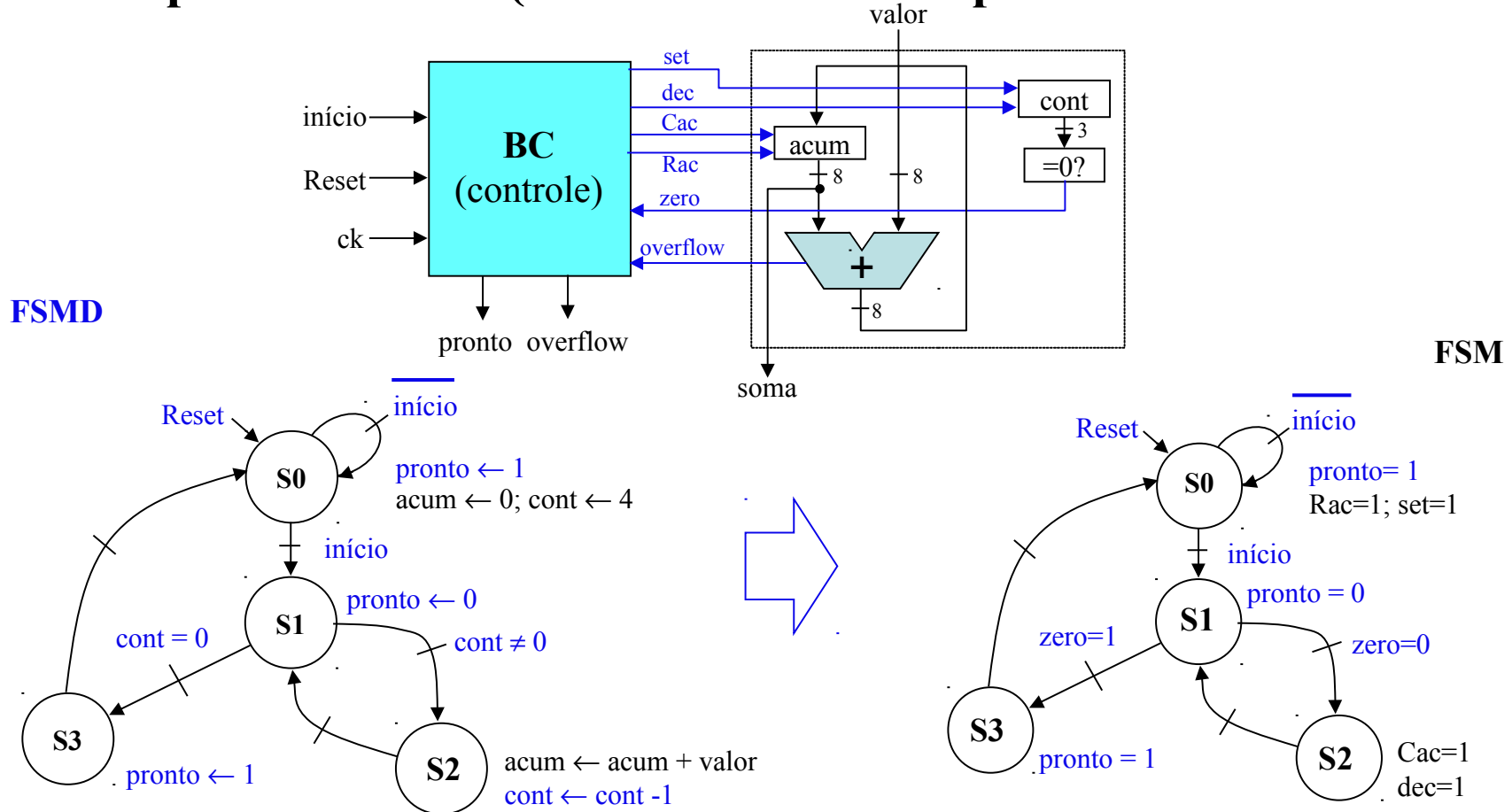
Exemplo 1a: Passo 3 (Um diagrama BO/BC mais detalhado...)



4. Projeto de Sistemas Digitais no Nível RT

Projetando um Sistema Digital

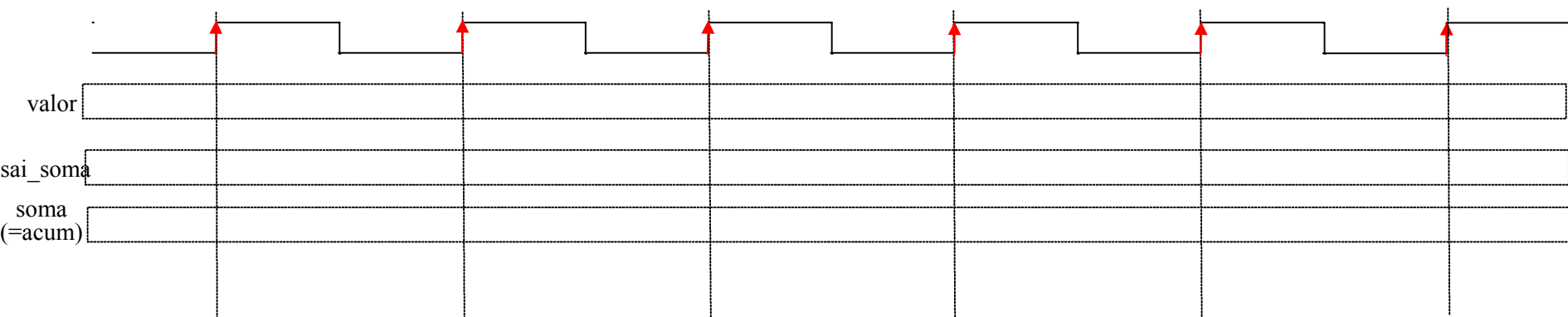
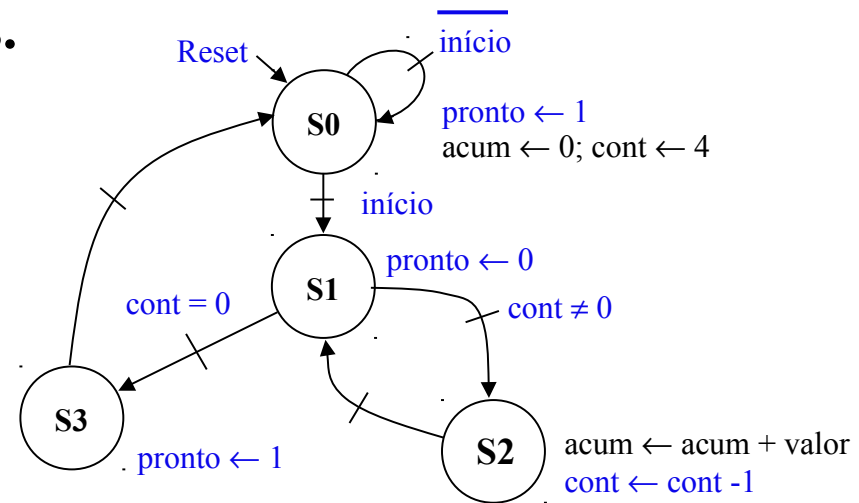
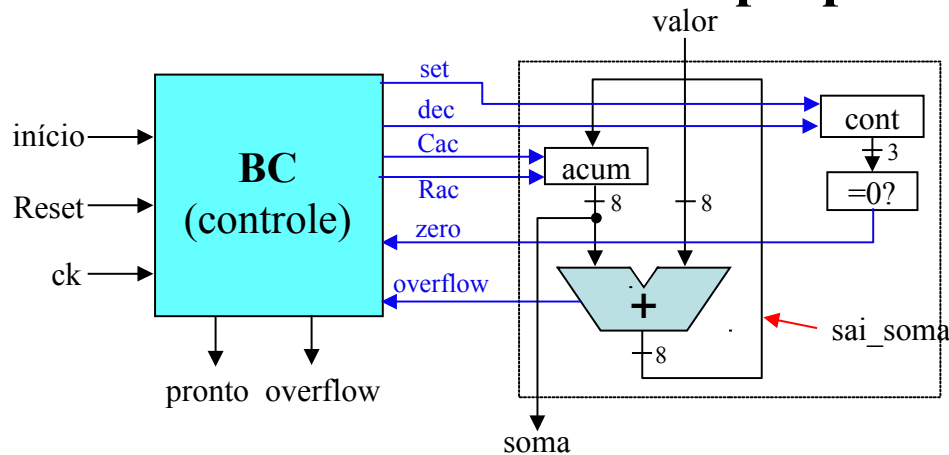
Exemplo 1a: Passo 4 (Derivando a FSM a partir do BO e da FSMD)



4. Projeto de Sistemas Digitais no Nível RT

▶ Proietando um Sistema Digital

Exercício: ilustrar o comportamento do S.D. Do exemplo 1a por meio de formas de onda apropriadas.

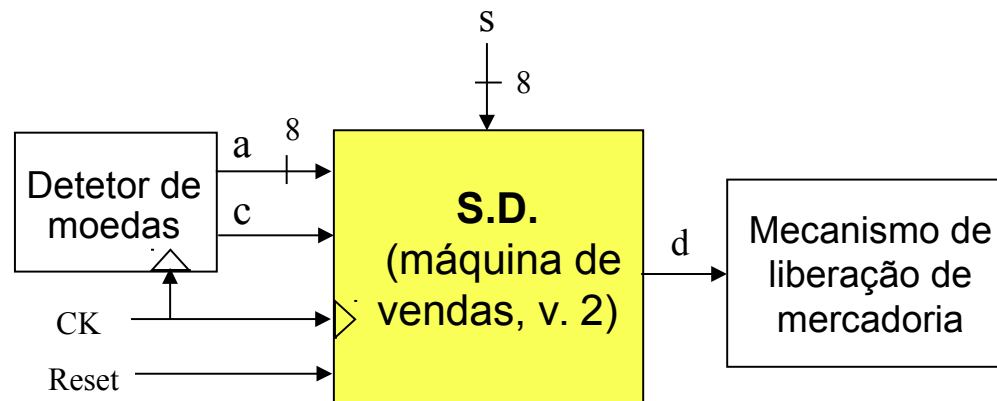


4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 2: máquina de vendas (versão 2)

Necessita-se de uma máquina de vendas capaz de gerenciar a venda de (apenas) um tipo de mercadoria, doravante denominada “item”. Esta máquina possui um detetor de moedas que provê um sinal de 1 bit chamado “**c**”, o qual vale “1” durante um ciclo de relógio quando uma moeda é detetada. (O controle da máquina de vendas e o seu detetor de moedas são sincronizados pelo mesmo sinal de relógio.)



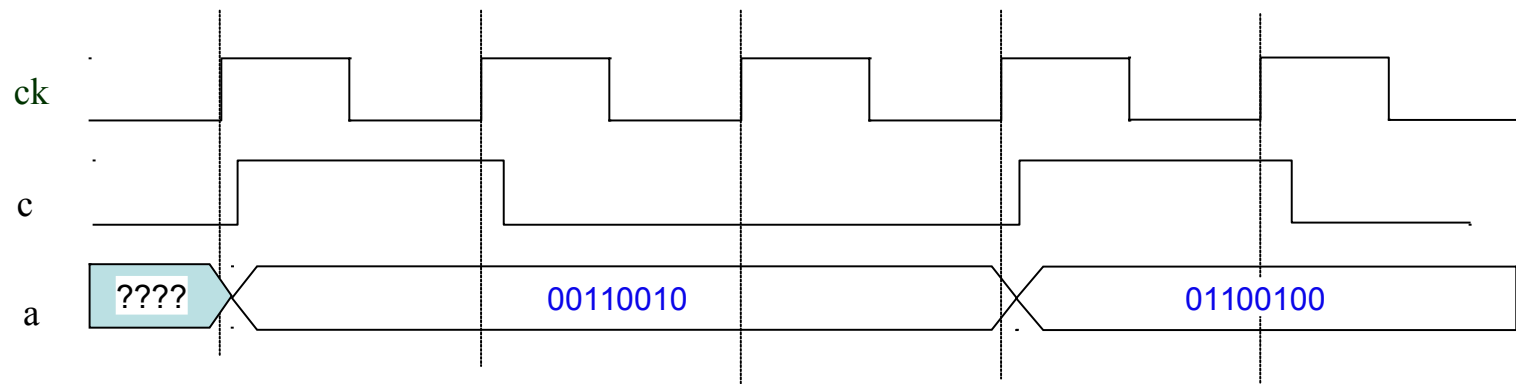
4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 2: máquina de vendas (versão 2)

O detetor também possui um registrador de 8 bits no qual ele armazena o valor da moeda inserida, em **centavos**. Este registrador é carregado somente quando o sinal “c” sobe. A saída deste registrador é entrada para a máquina de vendas, sendo chamada “a”.

Exemplo de sincronismo dos sinais “c” e “a” (detecção de uma moeda de 50 centavos e depois, de uma moeda de 1 real, supondo um relógio lento...)

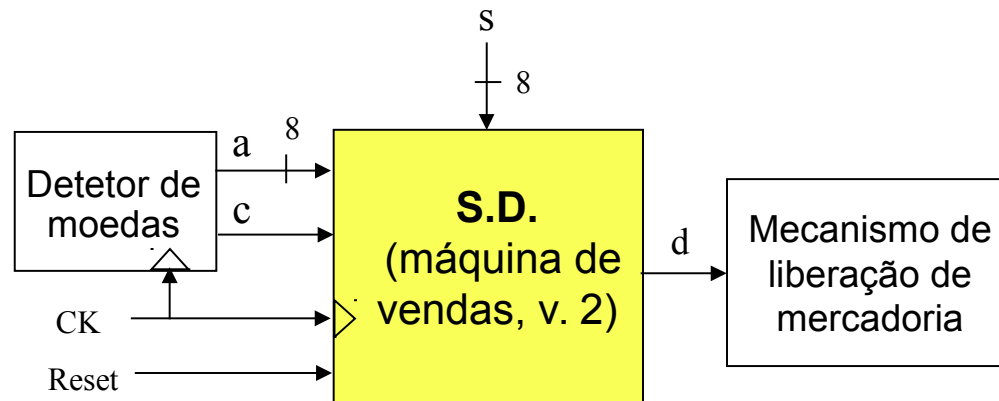


4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 2: máquina de vendas (versão 2)

A máquina possui ainda uma entrada de 8 bits denominada “s”, pela qual o proprietário pode definir o preço (unitário) da mercadoria. (Assuma que o valor correspondente ao preço permanece estável em “s” durante a operação normal.)



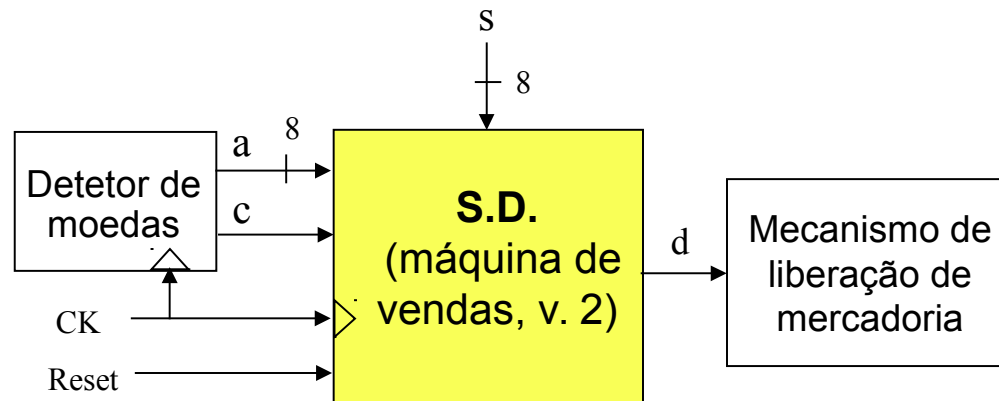
4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 2: máquina de vendas (versão 2)

Uma vez que o sistema digital identifica moedas cujo valor seja igual ou maior que o preço do item, ele “seta” o valor do sinal de saída “d” (que tem um bit) durante um ciclo de relógio, causando a liberação de um item.

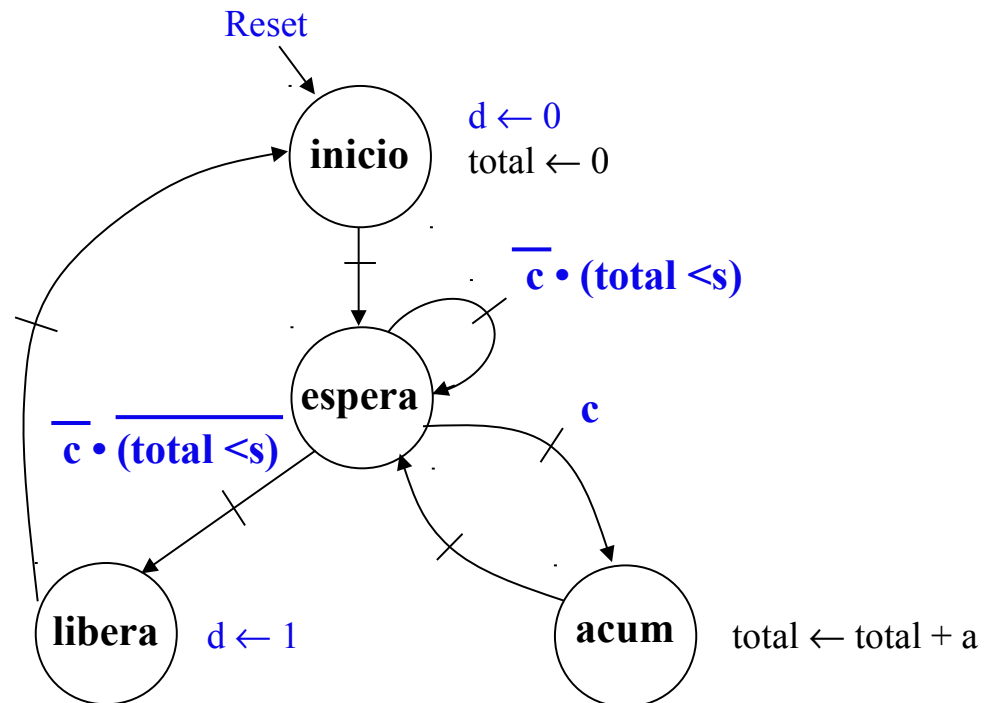
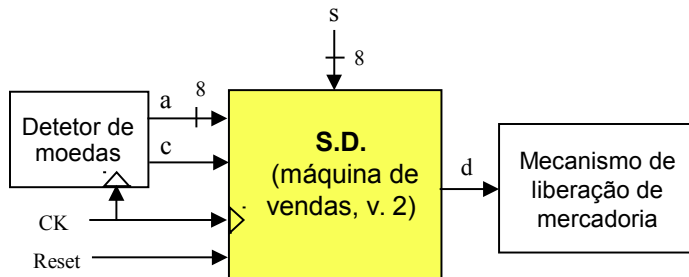
O Sistema não fornece troco.



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 2: Passo 1 (captura do comportamento com FSMD)



Pergunta:

- Em termos práticos, existe alguma limitação quanto à duração do período do relógio?

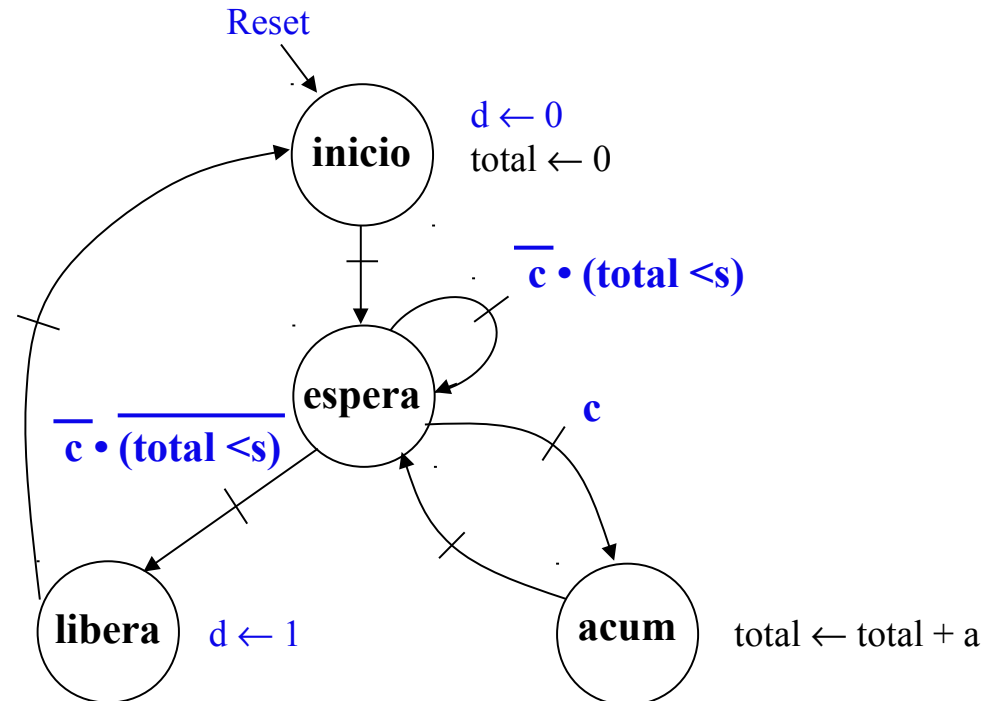
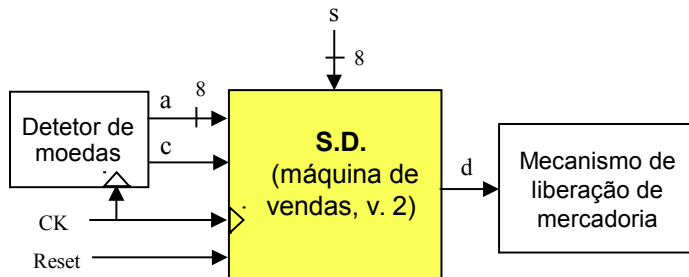
OBS:

- Condições para troca de estados em azul (junto às arestas), atribuições de controle também em azul (junto aos estados)
- Atribuições e operações com dados em preto, junto aos respectivos estados

4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 2: Passo 1 (captura do comportamento com FSMD)



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

Saídas: **d** (1 bit)

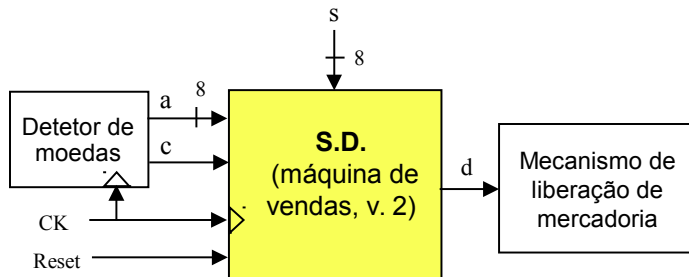
Variáveis internas ou locais: **total** (8 bits)

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

4. Projeto de Sistemas Digitais no Nível RT

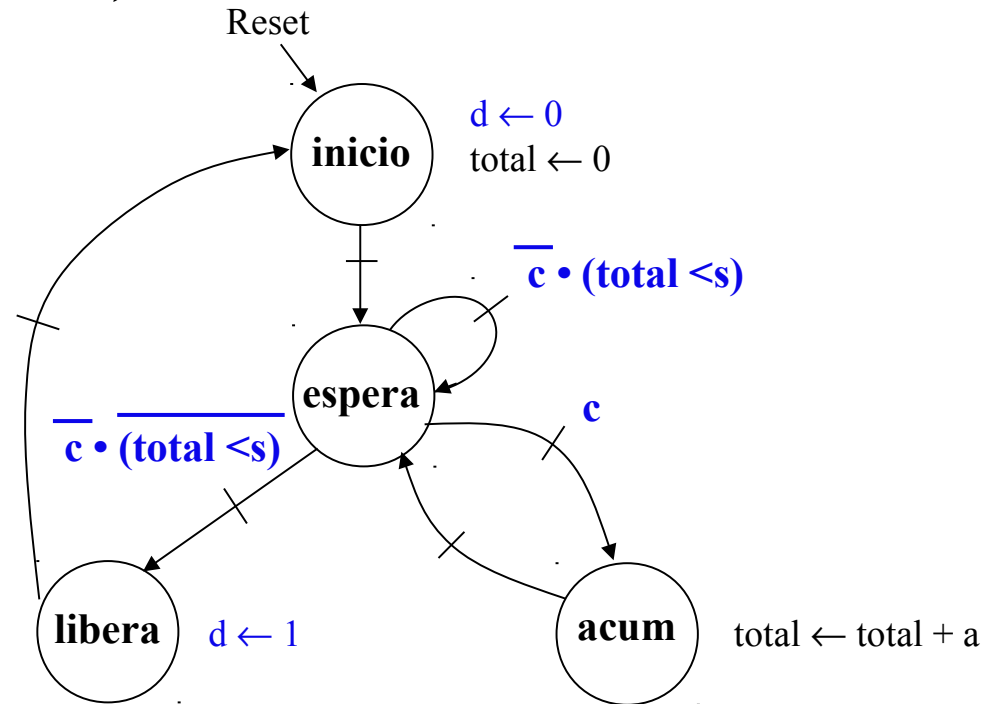
▶ Projetando um Sistema Digital

Exemplo 2: Passo 2 (projeto do BO)



Questões:

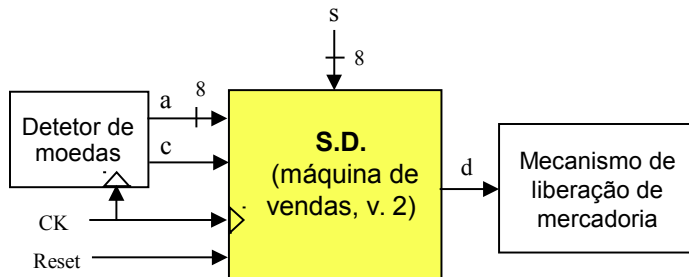
- Quais variáveis são usadas para armazenar dados?
- Quais operações são realizadas sobre dados (incluindo-se as condições)?



4. Projeto de Sistemas Digitais no Nível RT

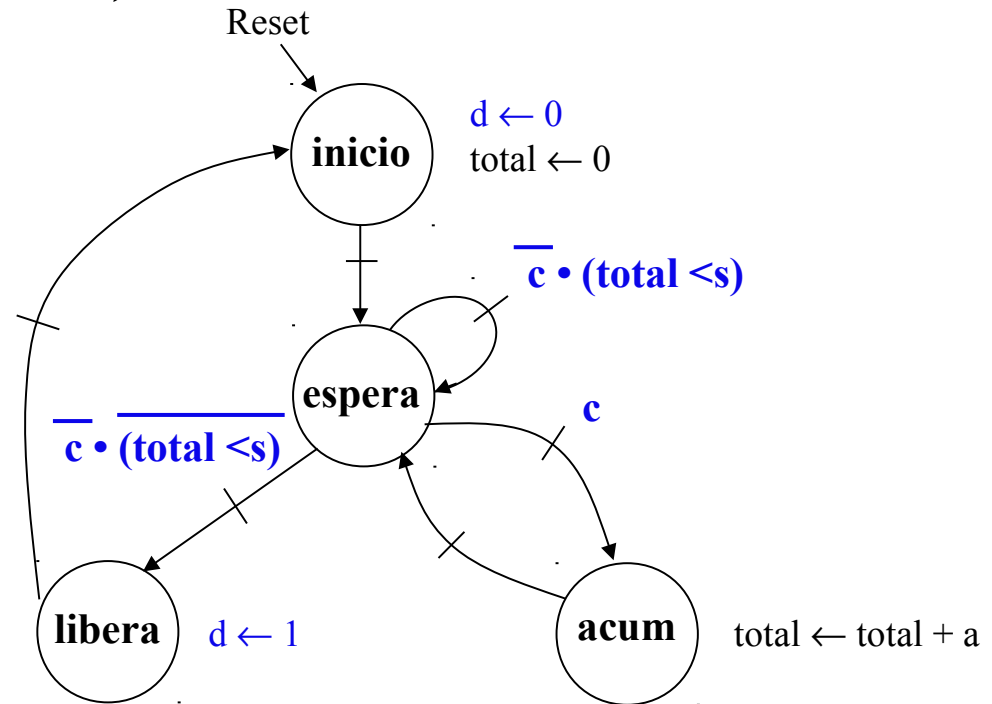
▶ Projetando um Sistema Digital

Exemplo 2: Passo 2 (projeto do BO)



Questões:

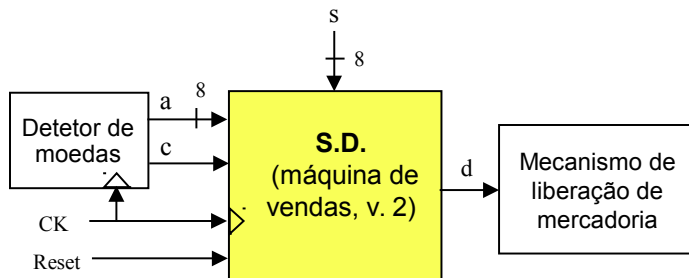
- Quais variáveis são usadas para armazenar dados?
- Apenas uma variável: “total” (note que “a” é uma entrada)
- Logo, teremos um registrador denominado “total” para esta variável



4. Projeto de Sistemas Digitais no Nível RT

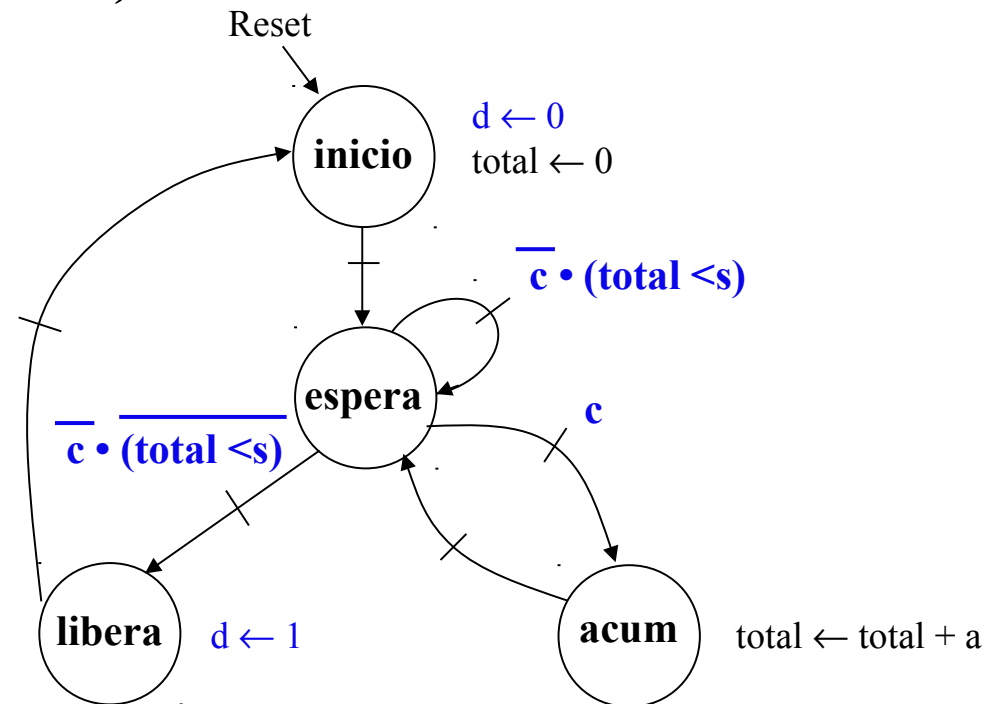
▶ Projetando um Sistema Digital

Exemplo 2: Passo 2 (projeto do BO)



Questões:

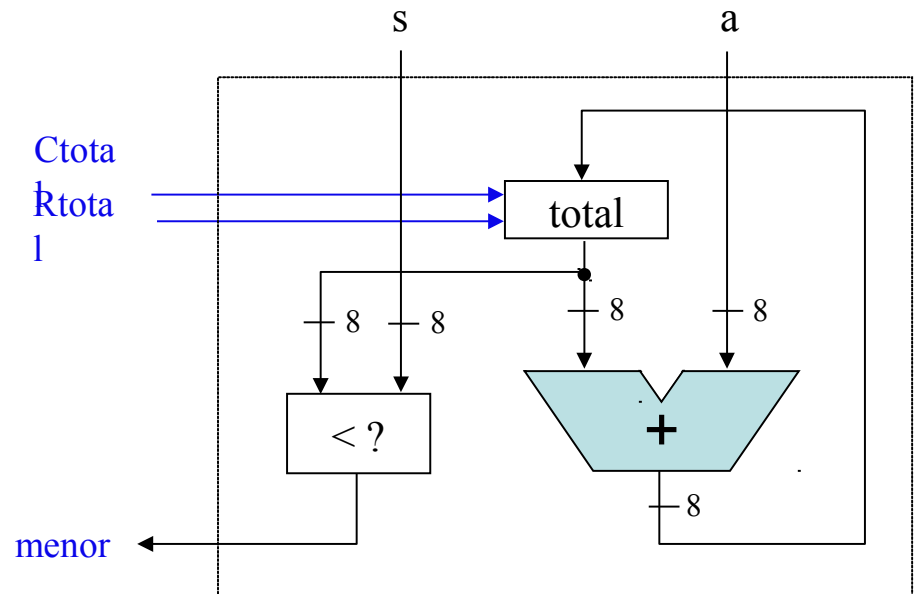
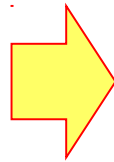
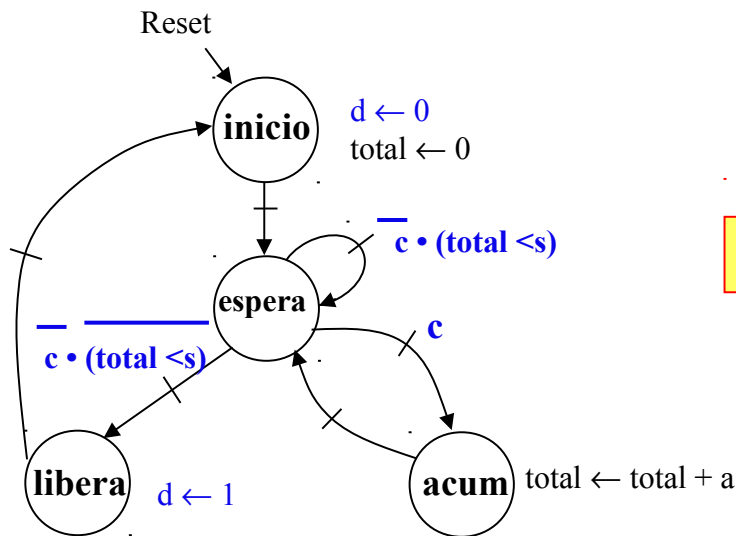
- Quais operações são realizadas sobre dados (incluindo-se as condições)?
- Uma adição para números de 8 bits ($\text{total} \leftarrow \text{total} + a$)
- Uma comparação entre “total” e “s”, ambos com 8 bits.



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 2: Passo 2 (projeto do BO)



Convenção:

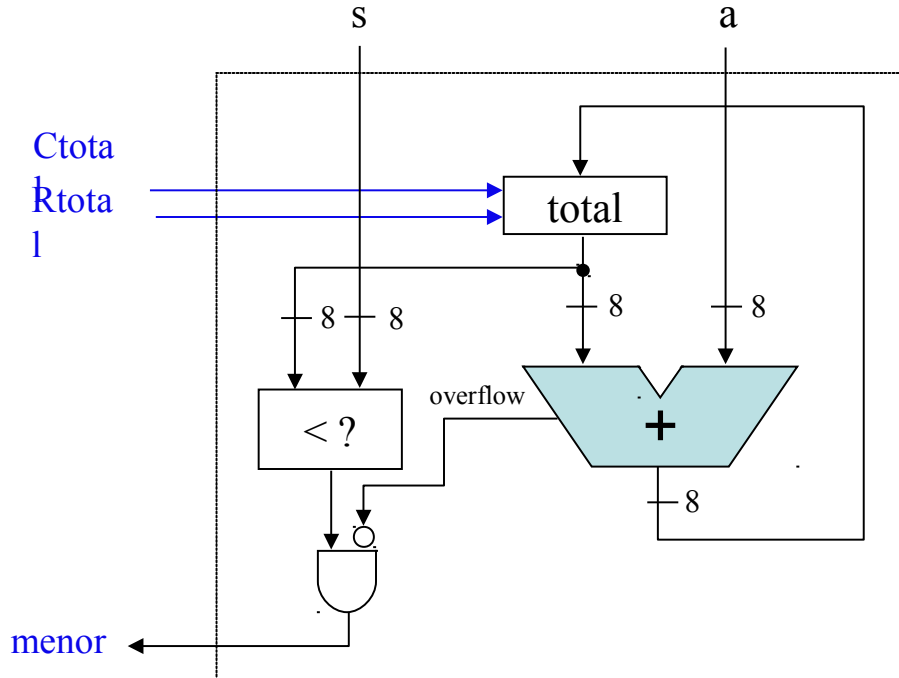
- Todos os registradores são cadenciados pelo sinal de relógio (ck). Entretanto, para simplificar o desenho, o sinal de relógio está omitido.

4. Projeto de Sistemas Digitais no Nível RT

▶ **Projetando um Sistema Digital**

Exemplo 2: Passo 2 (projeto do BO)

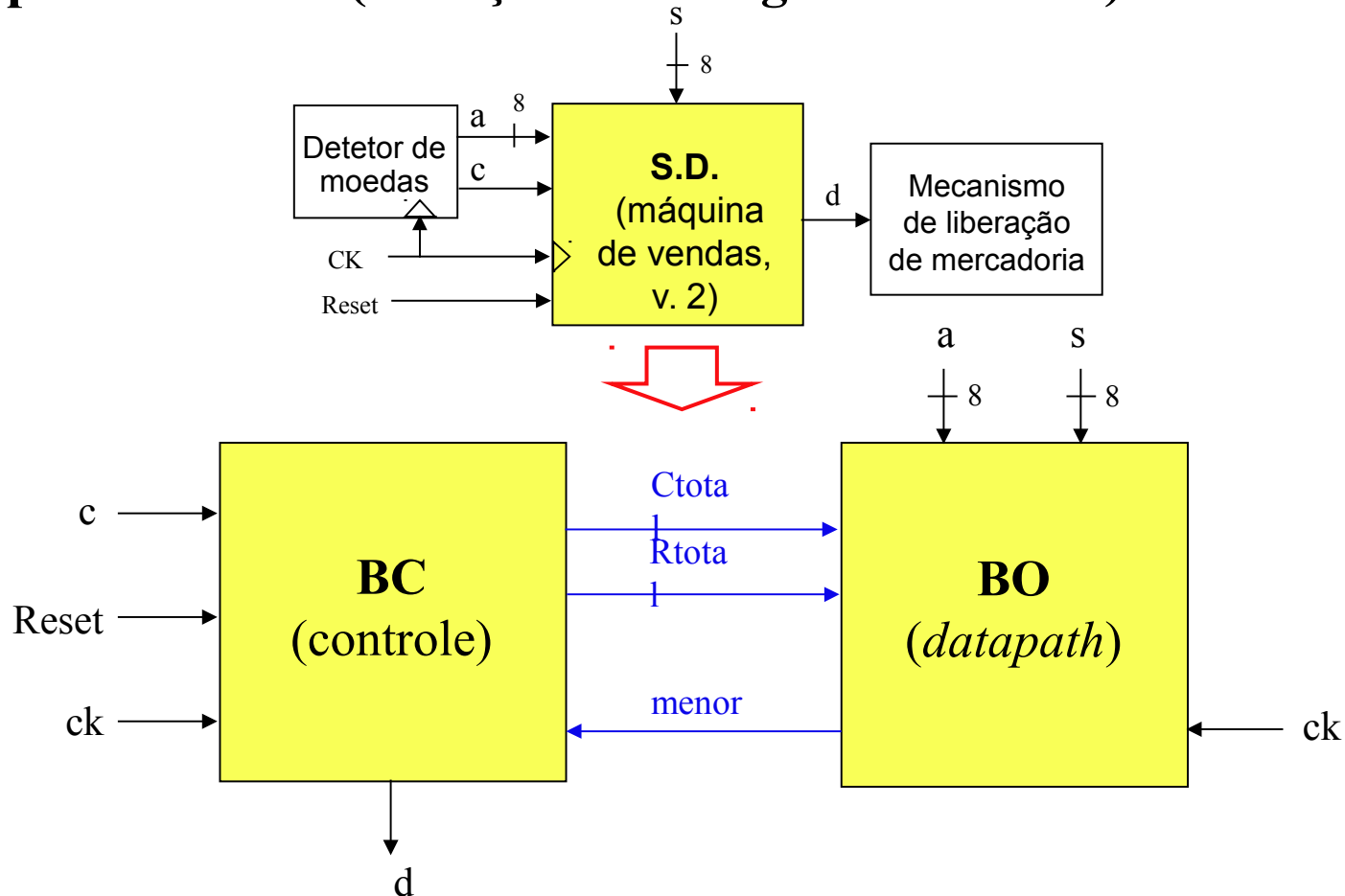
Prevendo a possibilidade de ocorrência de *overflow*.



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

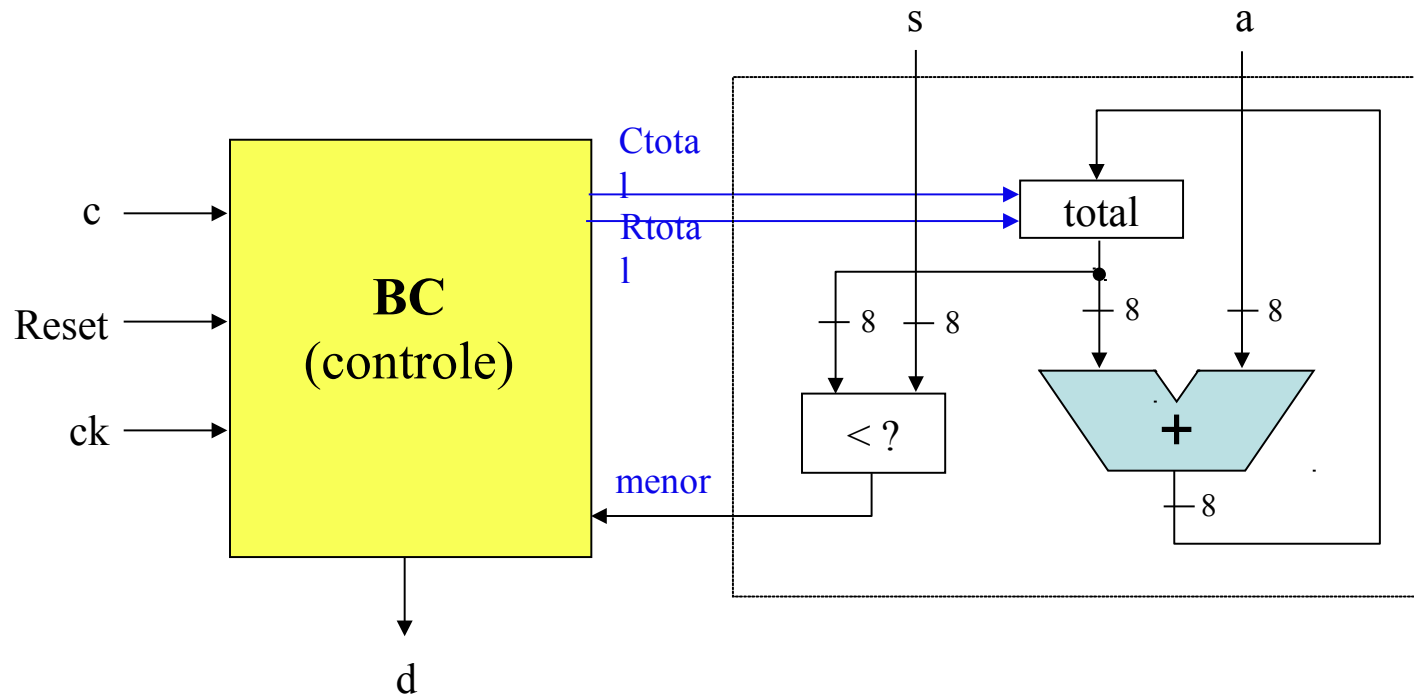
Exemplo 2: Passo 3 (Esboçando o diagrama BO/BC)



4. Projeto de Sistemas Digitais no Nível RT

▶ **Projetando um Sistema Digital**

Exemplo 2: Passo 3 (Um diagrama BO/BC mais detalhado...)



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 2: Passo 4 (Projeto do BC)

FSM

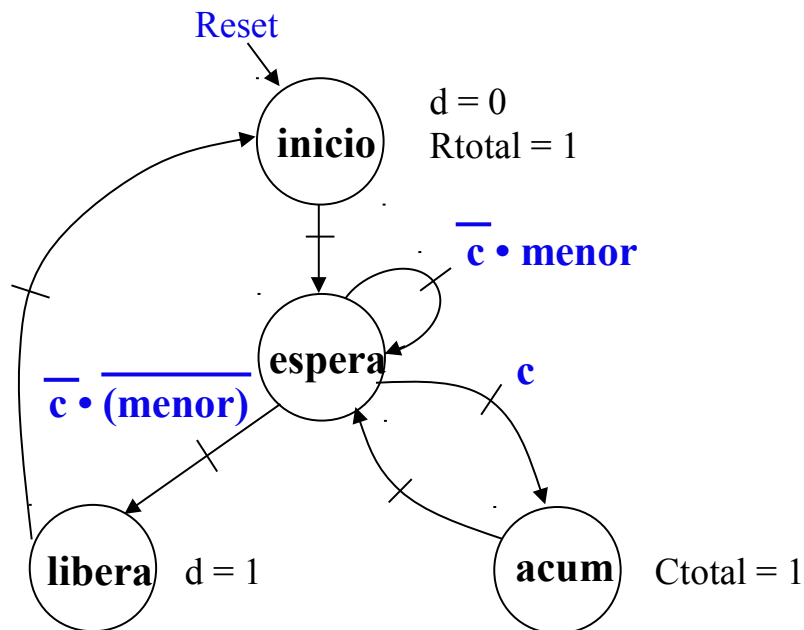


Tabela de Transição de Estados

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

4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

Exemplo 2: Passo 4 (Projeto do BC)

FSM

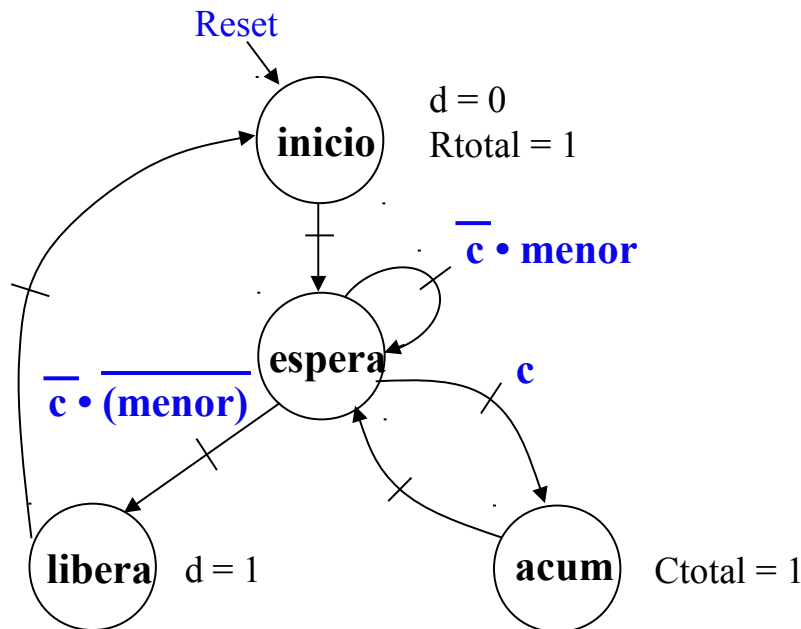


Tabela de Saídas

Estado	Rtotal	Ctotal	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 desativado (=0).