



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



Sistemas Digitais

INE 5406

Aula 13-T

4. Projeto de Sistemas Digitais no Nível RT. Estudo de Caso.

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

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

4. Projeto de Sistemas Digitais no Nível RT

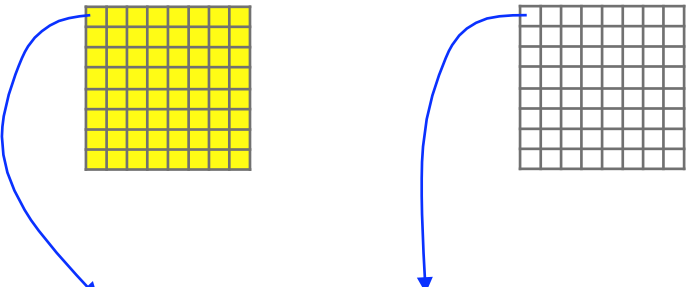
► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD (*Sum of Absolute Differences*)

- A SAD é uma operação realizada sobre duas matrizes de pixels (A e B), gerando um valor único:

Matriz A
($n \times n$)

Matriz B
($n \times n$)


$$SAD = \sum_{i=0, j=0}^{i \leq 7, j \leq 7} ABS (pixel_A(i,j) - pixel_B(i,j))$$

4. Projeto de Sistemas Digitais no Nível RT

► **Projetando um Sistema Digital**

Exemplo 3: cálculo da SAD (*Sum of Absolute Differences*)

- O valor calculado (SAD) é utilizado como uma medida do grau de semelhança entre as duas matrizes (e portanto, da semelhança entre as imagens por elas representadas): quanto menor for o valor “SAD” entre duas matrizes, mais semelhantes elas são.
- O objetivo deste exemplo é estudar sistemas digitais capazes de realizar o cálculo da SAD.
- A fim de contextualizar o exemplo, as próximas transparências abordam alguns dos princípios da compressão de vídeo digital

4. Projeto de Sistemas Digitais no Nível RT

► Princípios de Vídeo Digital

- Os equipamentos (digitais ou analógicos) armazenam filmes como sequências de imagens estáticas (ou seja, sequências de fotos)
- Em um filme, cada imagem estática é chamada de “quadro” (*frame*, em inglês)
- Para que as transições entre as imagens estáticas não sejam percebidas pelo olho humano (de modo que enxerguemos um filme), é necessário que a taxa de exibição seja igual ou superior a 30 quadros por segundo (*frames per second*, ou simplesmente, fps)

4. Projeto de Sistemas Digitais no Nível RT

► Princípios de Vídeo Digital

- A qualidade da imagem é diretamente proporcional ao número de pixels de cada quadro.
- Se não for usada alguma técnica de **compressão de vídeo**, todos os pixels de todos os quadros precisarão ser armazenados. Neste caso:
 - a quantidade de memória necessária para armazenar filmes (ou trechos de filmes) será enorme
 - a taxa necessária para transmitir um vídeo será ser enorme

4. Projeto de Sistemas Digitais no Nível RT

► Princípios de Vídeo Digital

- Seja um vídeo sem compressão, com 10 minutos de duração, **30** quadros por segundo, sendo cada pixel armazenado com **24** bits:

Formato	Resolução	Memória para armazenar 10 minutos de vídeo
SDTV (e DVD)	720x480 pixels	19 GB
HDTV	1920x1080 pixels	112 GB

Formato	Resolução	Taxa requerida p/ transmitir 10 minutos de vídeo
SDTV (e DVD)	720x480 pixels	249 Mbps
HDTV	1920x1080 pixels	1,5 Gbps

Fonte: L. Agostini. Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas à Compressão de Vídeo Segundo o Padrão H.264/AVC. Porto Alegre: PPGC da UFRGS, 2007. Tese de doutorado.

4. Projeto de Sistemas Digitais no Nível RT

► Princípio da Compressão de Vídeo

- **Boa notícia!** Nos vídeos, normalmente quadros consecutivos são bastante semelhantes (Por que será?...)
- Logo, não é necessário armazenar todos os pixels de todos os quadros!
- Basta que se busquem semelhanças
 - entre as regiões dentro de um quadro: esta é a chamada **previsão intraquadro** (*intraframe prediction*)
 - e/ou semelhanças entre quadros próximos: esta é a chamada **previsão interquadros** (*interframe prediction*)
- Em vídeo, a **previsão interquadros** é a responsável por altas taxas de compressão

4. Projeto de Sistemas Digitais no Nível RT

► Previsão Interquadros

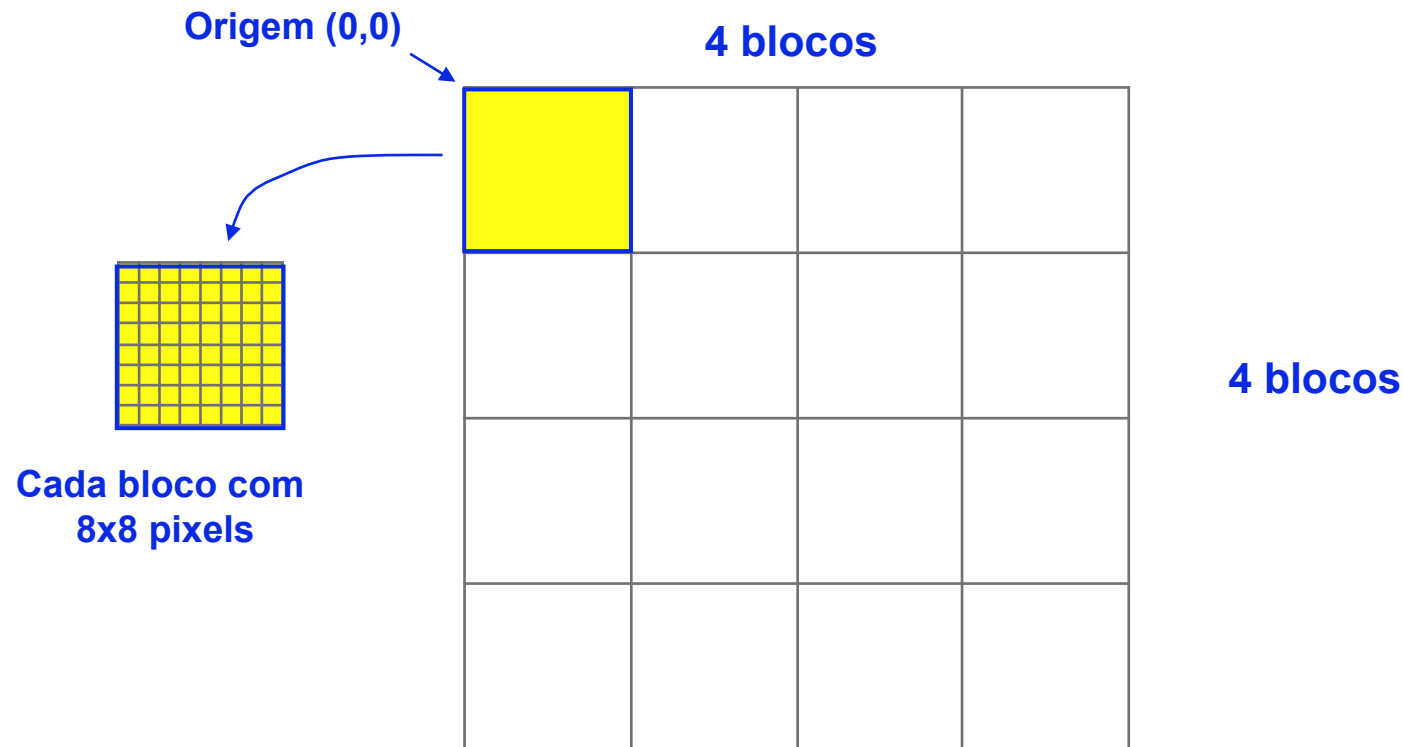
Princípio:

- De tantos em tantos quadros, somente um quadro é armazenado completamente. Ele recebe o nome de **quadro de referência (Qref)**.
- Os demais quadros que sucedem (e às vezes, alguns que antecedem também) são armazenados de maneira simplificada, por meio de “**vetores de movimento**”

4. Projeto de Sistemas Digitais no Nível RT

► Previsão Interquadros

- Cada quadro é dividido em sub-matrizes de pixels, denominadas “blocos”. Exemplo hipotético:



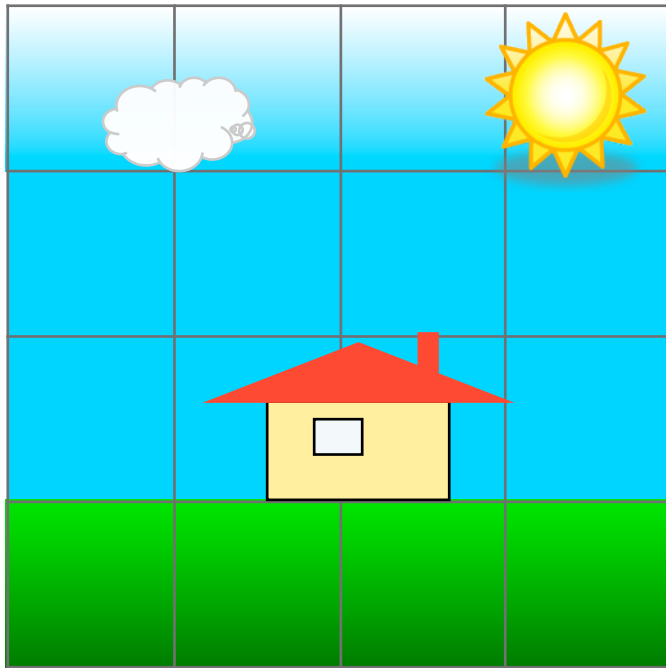
4. Projeto de Sistemas Digitais no Nível RT

► Previsão Interquadros

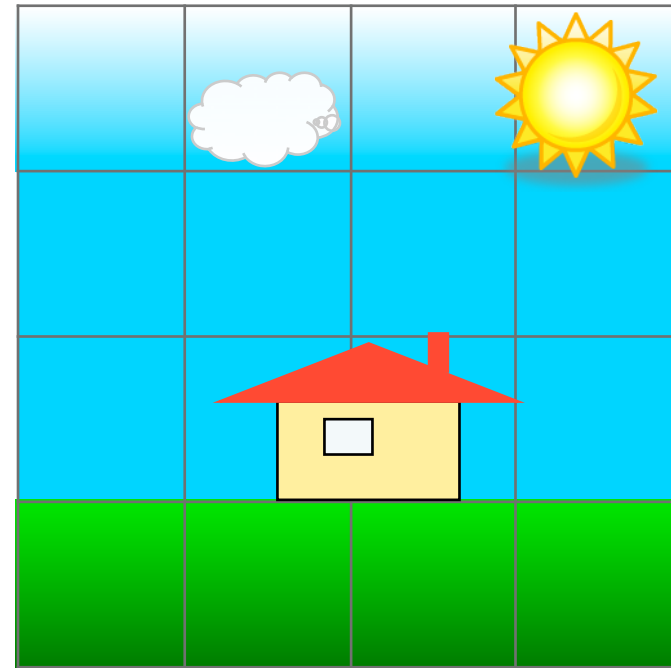
- Um determinado quadro é escolhido para ser “referência”(Q_{ref}). Todos os pixels deste quadro serão armazenados.
- Para os quadros Q_i que sucedem Q_{ref} , serão calculados os “**vetores de movimento**” (pois somente estes serão armazenados).
- Assim, para cada bloco $b_j \in Q_i$ será encontrado um vetor de movimento da seguinte maneira:
 - É buscado o melhor “casamento” (*matching*) entre cada bloco $b_j \in Q_i$ e alguma porção (sub-matriz) de Q_{ref}
 - Encontrado o melhor casamento, o bloco de Q_i será representado como um par $\{x_i, y_i\}$ denominado de “**vetor de movimento**”, que aponta para a aresta superior esquerda da porção (sub-matriz) em Q_{ref} que contém os pixels que serão usados para representar o bloco $b_j (\in Q_i)$

4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”



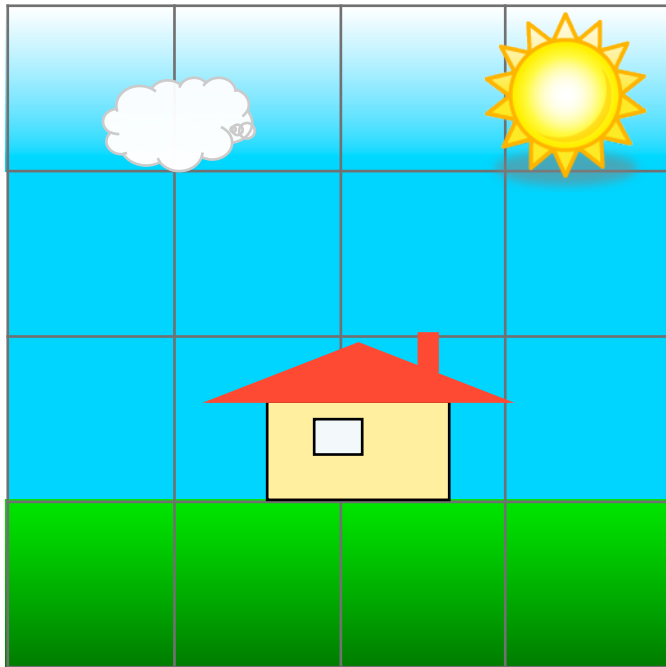
Q_{ref}



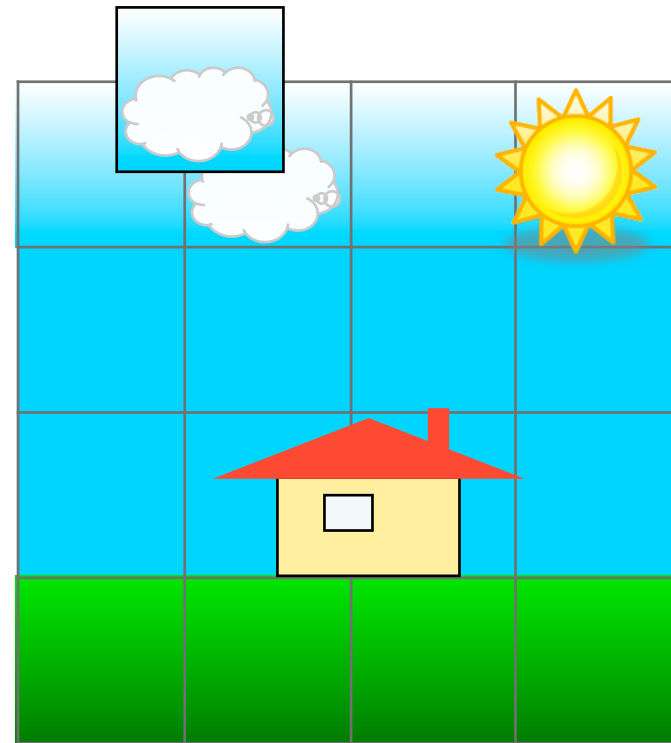
Q_i

4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”



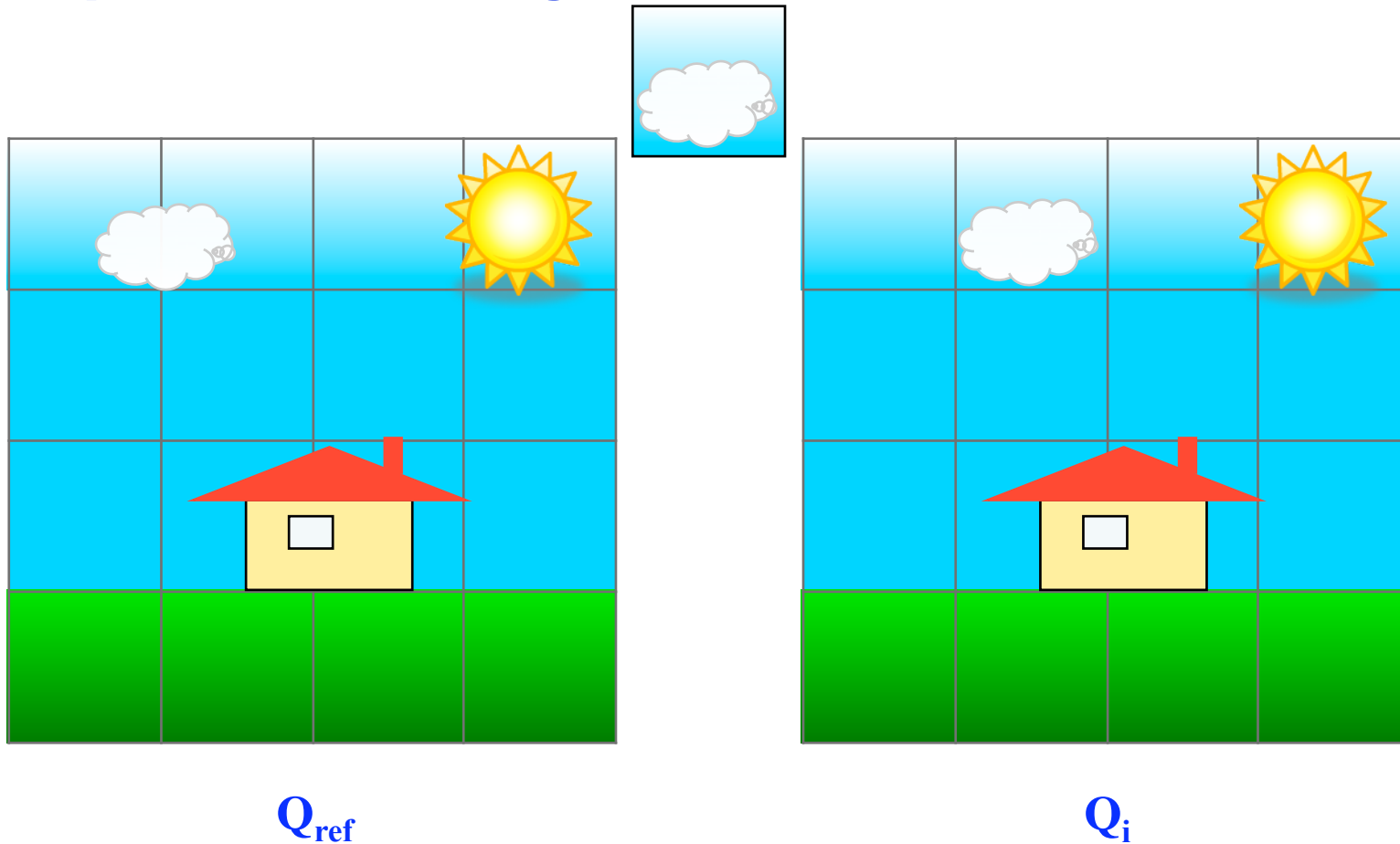
Q_{ref}



Q_i

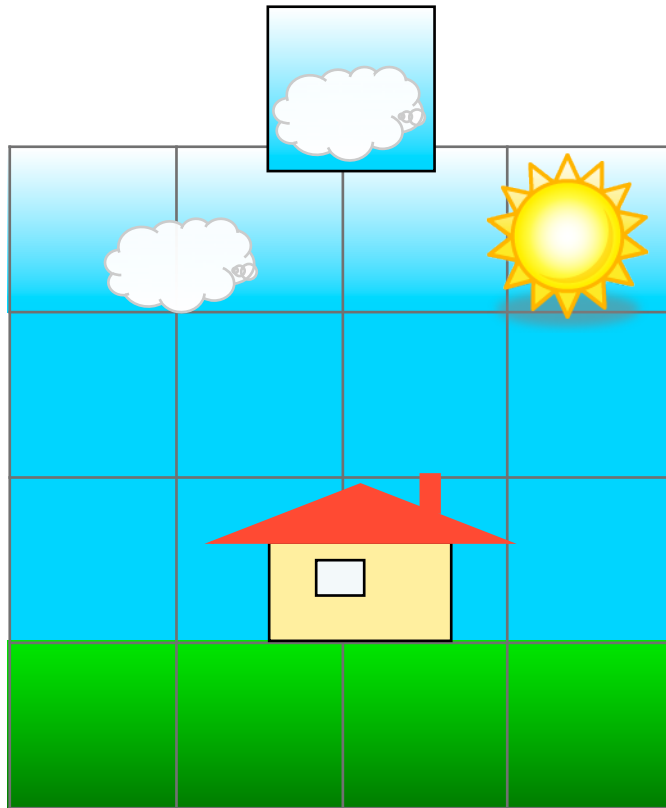
4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”

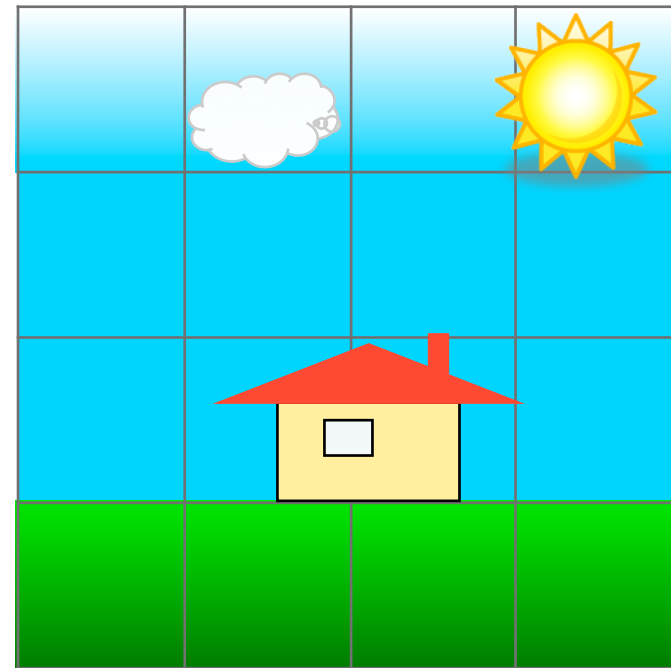


4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”



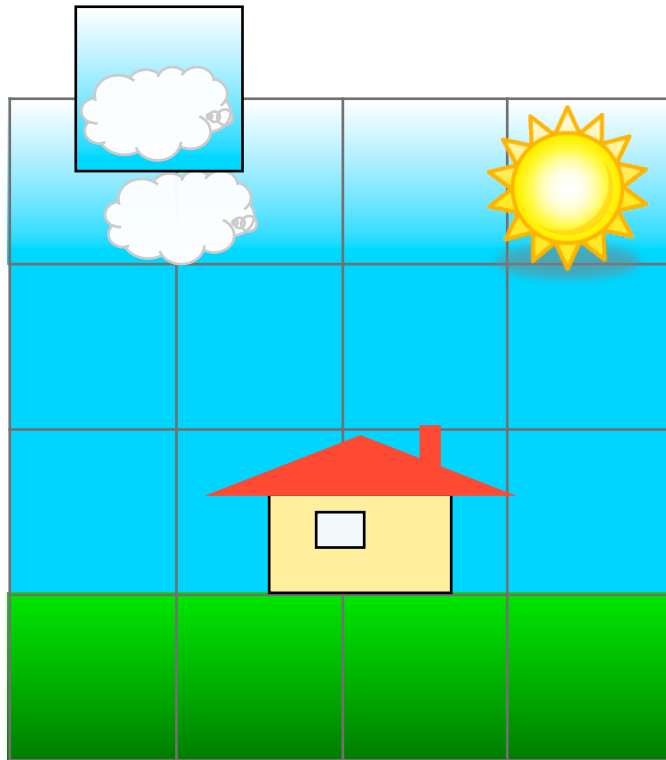
Q_{ref}



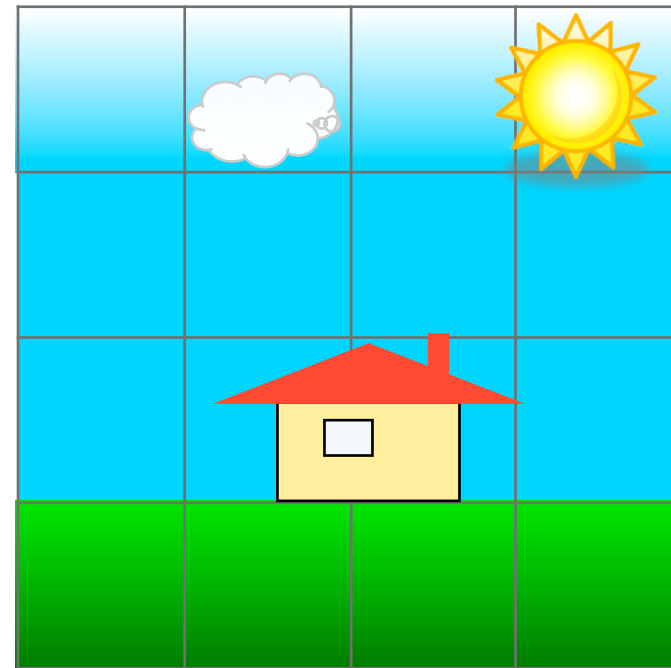
Q_i

4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”



Q_{ref}

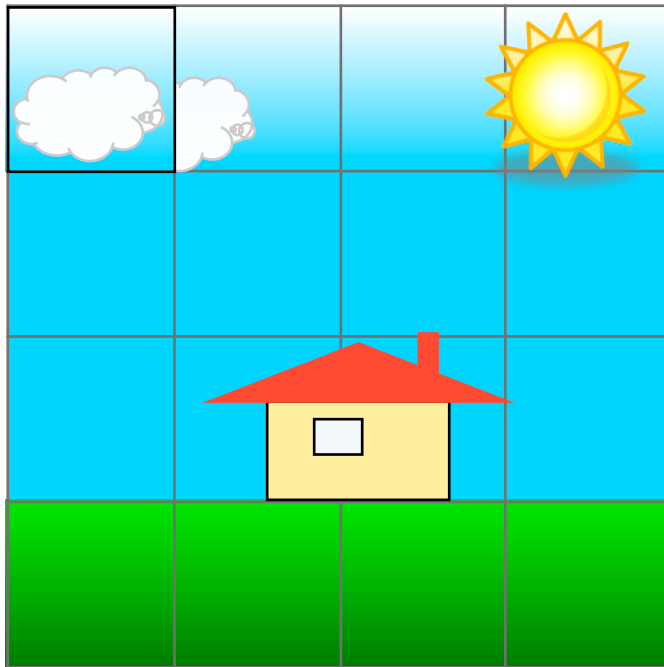


Q_i

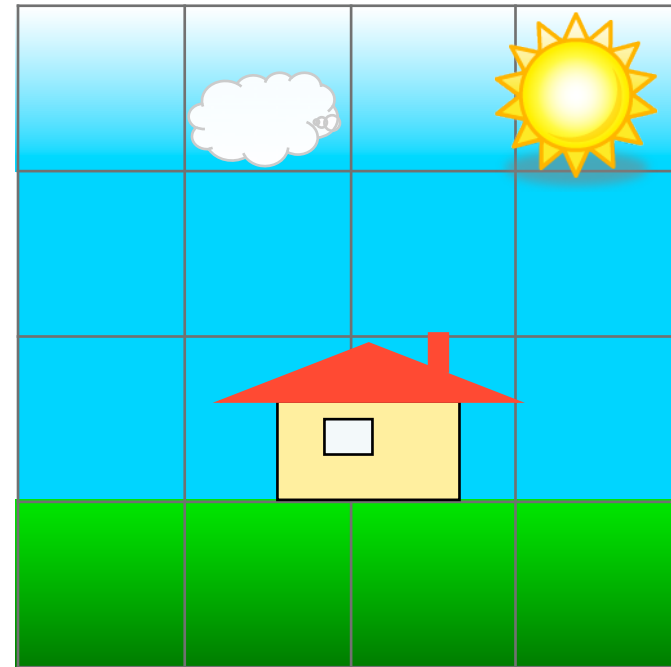
4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”

Comparando $b_j \in Q_i$ com bloco
 $\in Q_{ref}$ cuja origem é $x=0, y=0$



Q_{ref}

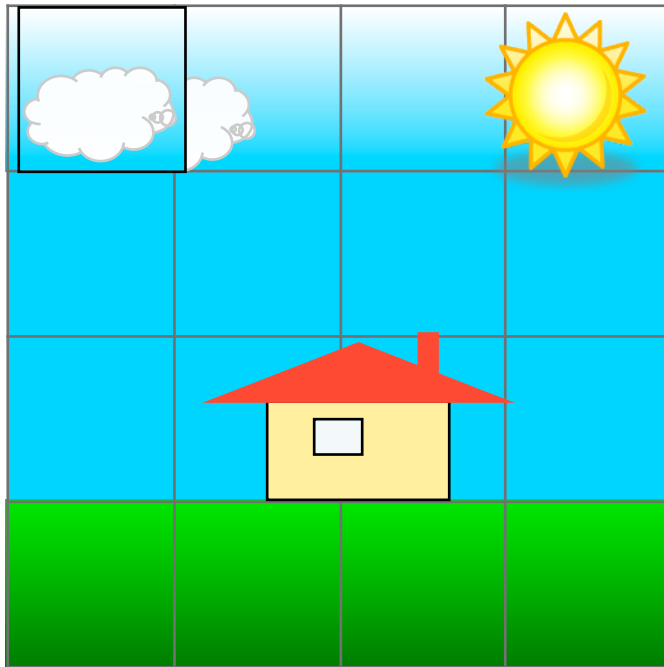


Q_i

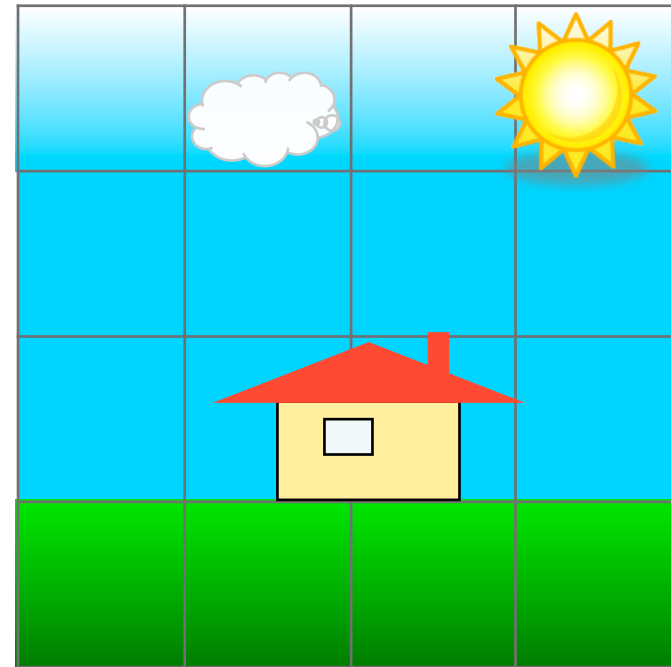
4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”

Comparando $b_j \in Q_i$ com bloco
 $\in Q_{ref}$ cuja origem é $x=1, y=0$



Q_{ref}

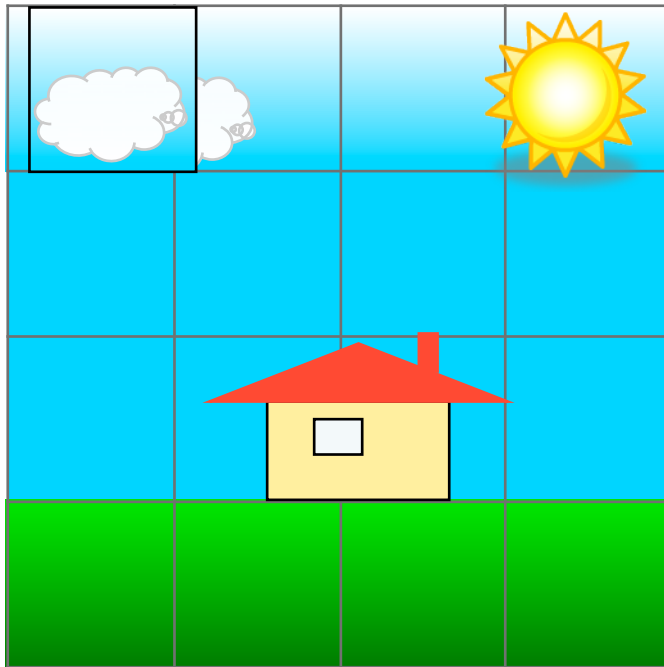


Q_i

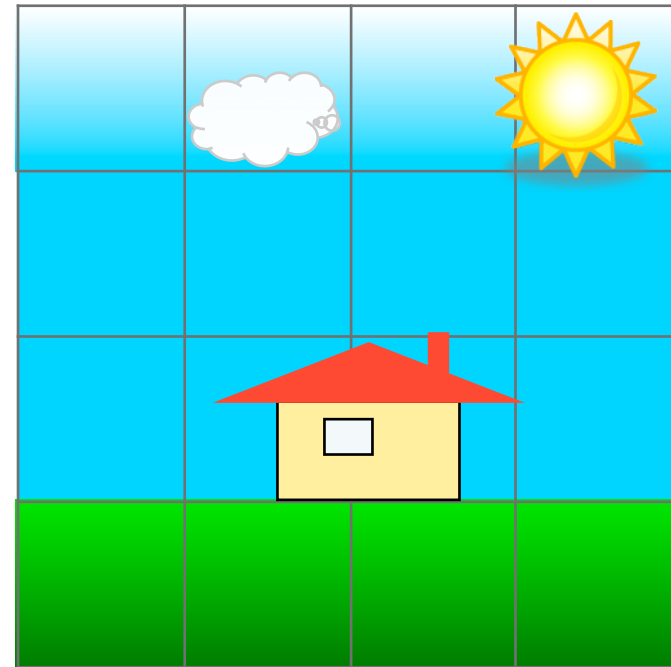
4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”

Comparando $b_j \in Q_i$ com bloco
 $\in Q_{ref}$ cuja origem é $x=2, y=0$



Q_{ref}

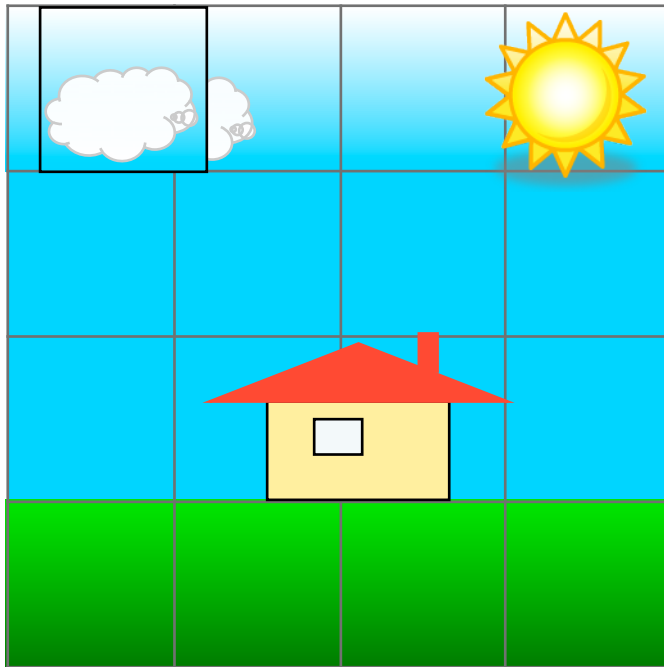


Q_i

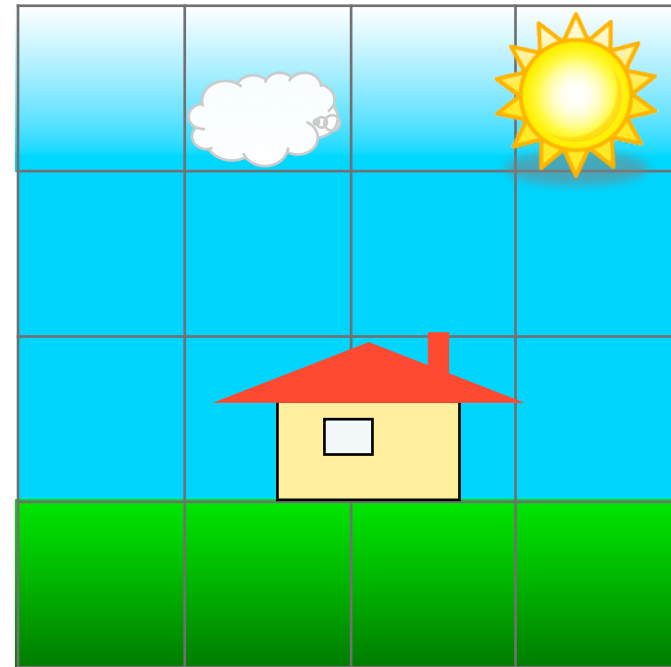
4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”

Comparando $b_j \in Q_i$ com bloco
 $\in Q_{ref}$ cuja origem é $x=3, y=0$



Q_{ref}

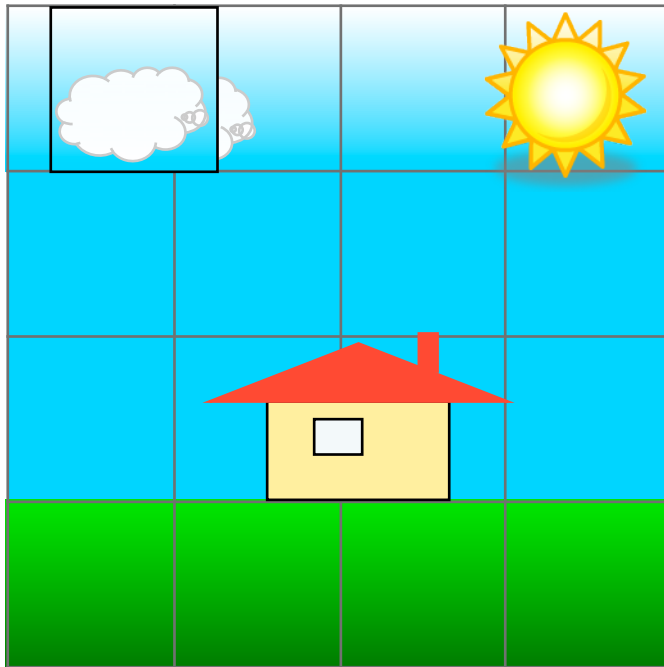


Q_i

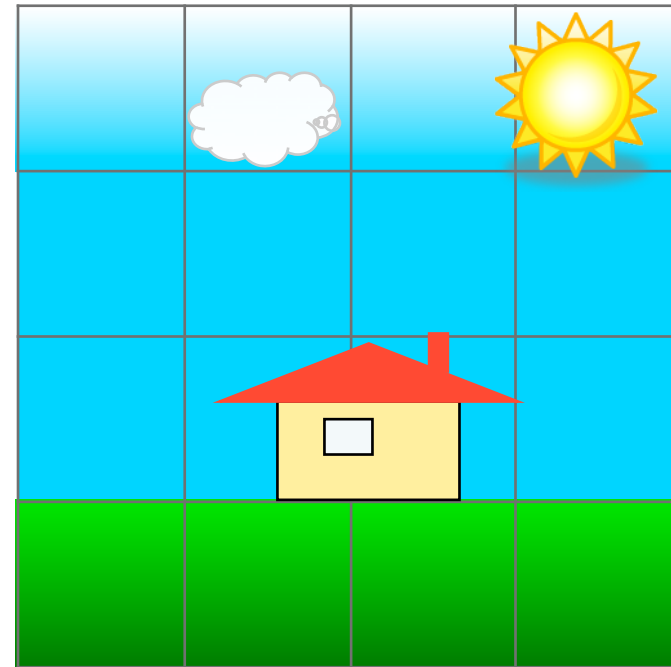
4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”

Comparando $b_j \in Q_i$ com bloco
 $\in Q_{ref}$ cuja origem é $x=4, y=0$



Q_{ref}

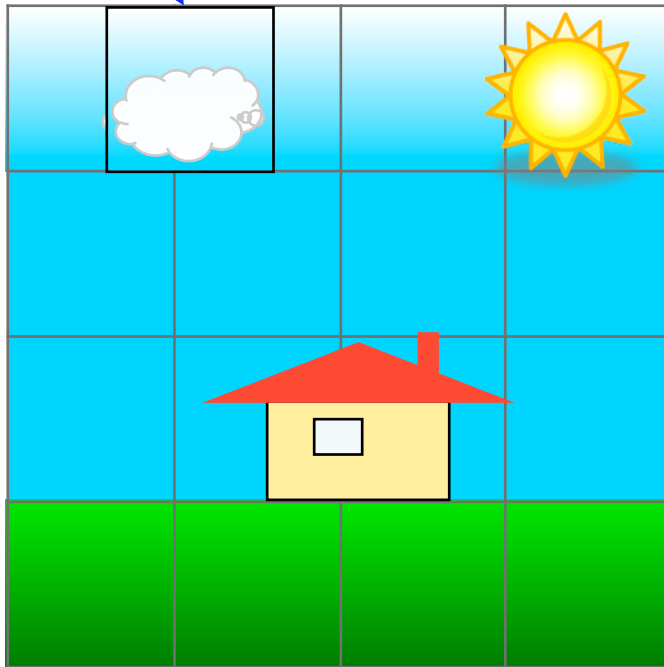


Q_i

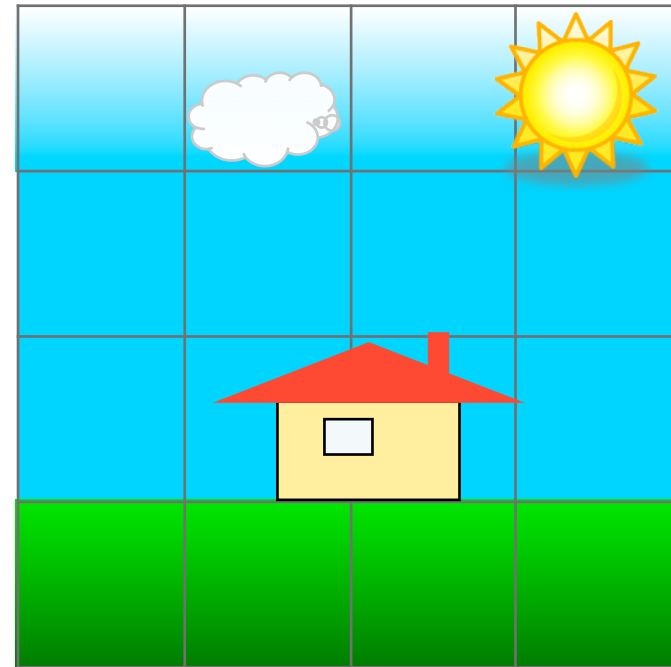
4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”

Melhor casamento! Logo,
vetor de movimento: $x=5$,
 $y=0$



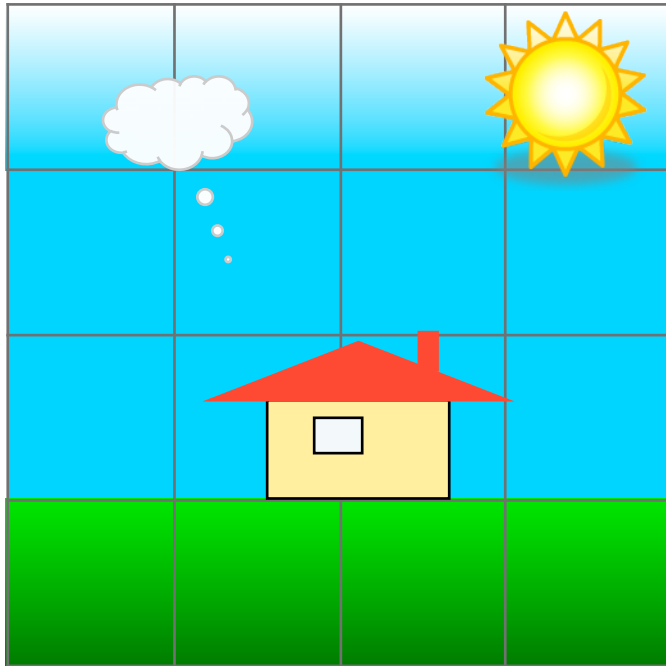
Q_{ref}



Q_i

4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”



Q_{ref}

(x_0, y_0)	$(x_1=5, y_1=0)$	(x_2, y_2)	(x_3, y_3)
(x_4, y_4)	(x_5, y_5)	(x_6, y_6)	(x_7, y_7)
(x_8, y_8)	(x_9, y_9)	(x_{10}, y_{10})	(x_{11}, y_{11})
(x_{12}, y_{12})	(x_{13}, y_{13})	(x_{14}, y_{14})	(x_{15}, y_{15})

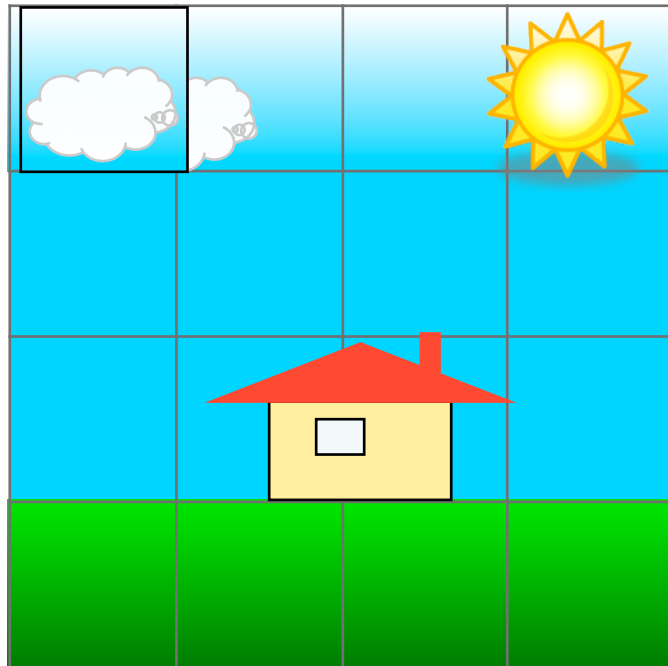
Q_i

Cada bloco de $b_j \in Q_i$ será representado por um vetor de movimento (ao invés 8x8 pixels)

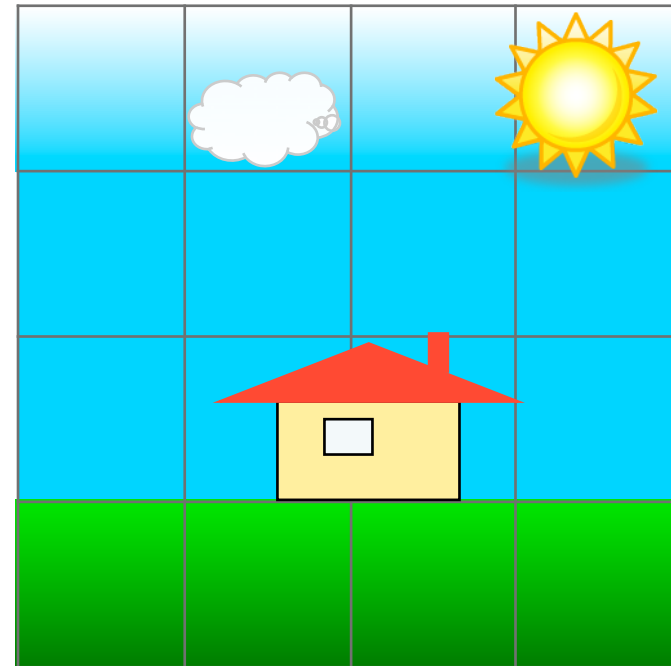
4. Projeto de Sistemas Digitais no Nível RT

► Etapa de “Estimação de Movimento”

Cada vez que se compara um bloco $b_j \in Q_i$ com um bloco de Q_{ref} , é preciso calcular a SAD entre duas matrizes de pixels



Q_{ref}



Q_i

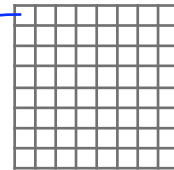
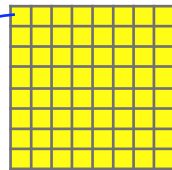
4. Projeto de Sistemas Digitais no Nível RT

► Cálculo do SAD (*Sum of Absolute Differences*)

Cada vez que se compara um bloco $b_j \in Q_i$ com uma porção de Q_{ref} , é preciso calcular a SAD entre duas matrizes

Matriz A:
Porção de 8x8 pixels
 $\in Q_{ref}$

Matriz B:
 $b_j \in Q_i$



$$SAD = \sum_{i=0, j=0}^{i \leq 7, j \leq 7} ABS (pixel_A(i,j) - pixel_B(i,j))$$

Calcular para cada par de pixels
 $\{ pixel_A(i,j), pixel_B(i,j) \}$
com $i \in [0,7]$ e $j \in [0,7]$

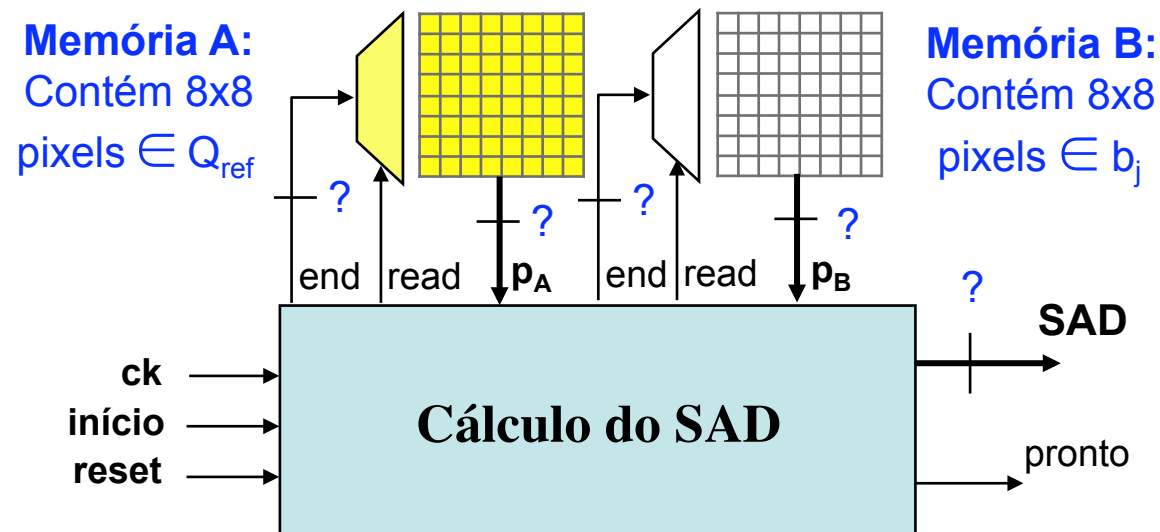
4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD (*Sum of Absolute Differences*)

Projetar um sistema digital capaz de realizar o cálculo da SAD entre um bloco $b_j \in Q_i$ e uma porção de pixels (de mesmo tamanho) $\in Q_{ref}$.

- Cada pixel é representado por 1 byte (8 bits)
- O bloco b_j e a porção de Q_i tem 8x8 pixels cada e (já) estão armazenados nas memórias B e A, respectivamente.

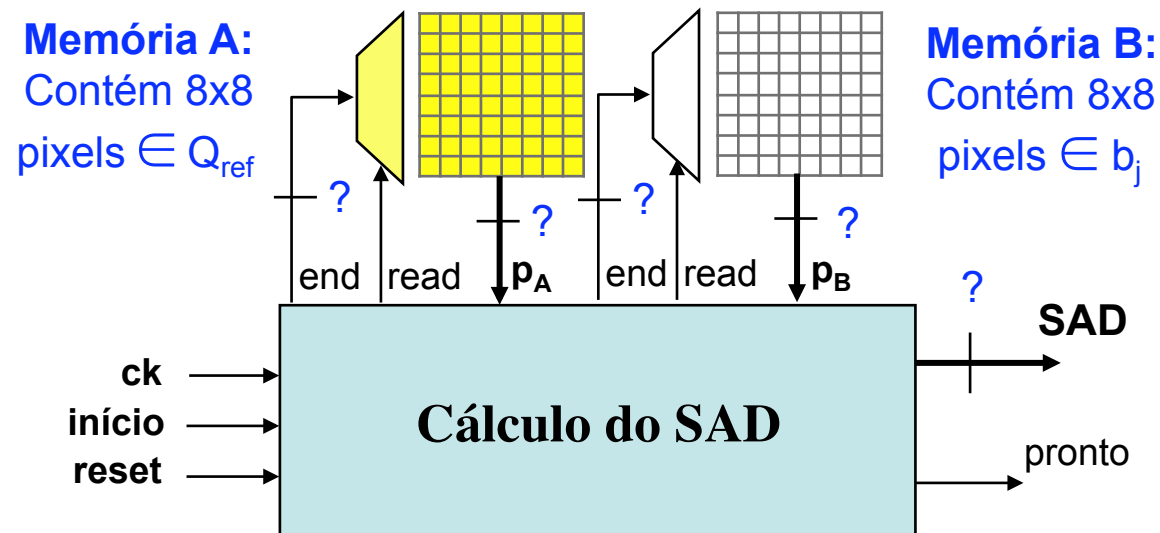


4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- O sinal início=1 indica que um cálculo de SAD deve iniciar.
- Funcionamento de cada memória: um pixel pode ser lido a cada ciclo de relógio; basta atualizar o “end” e manter “read=1”.
- O projeto deste sistema digital deve favorecer a otimização do custo.

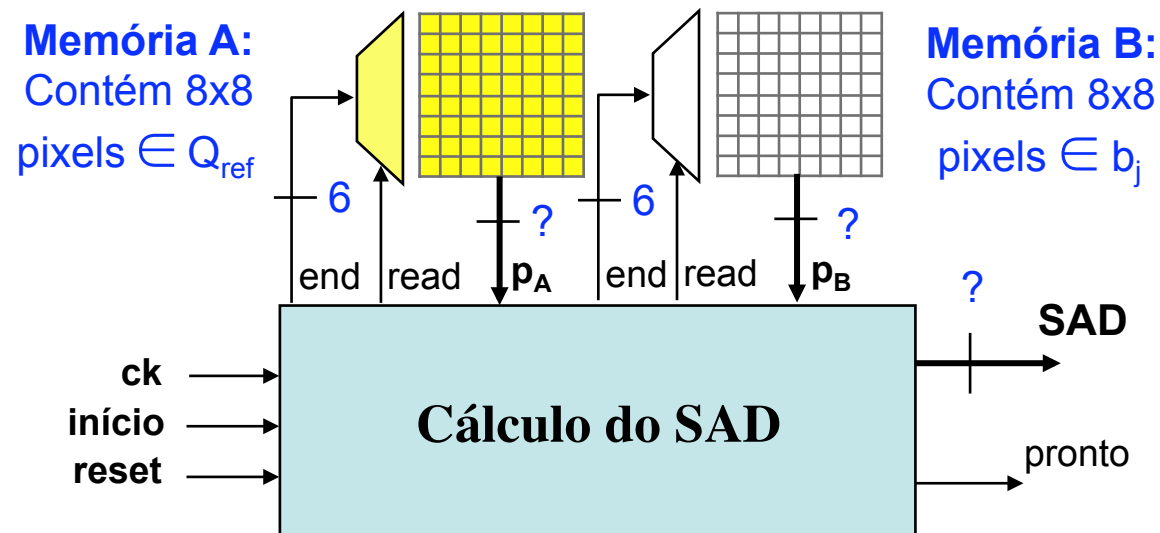


4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- Quantos bits devem ter as entradas p_A e p_B ?
- Quantos bits deve ter a saída SAD?

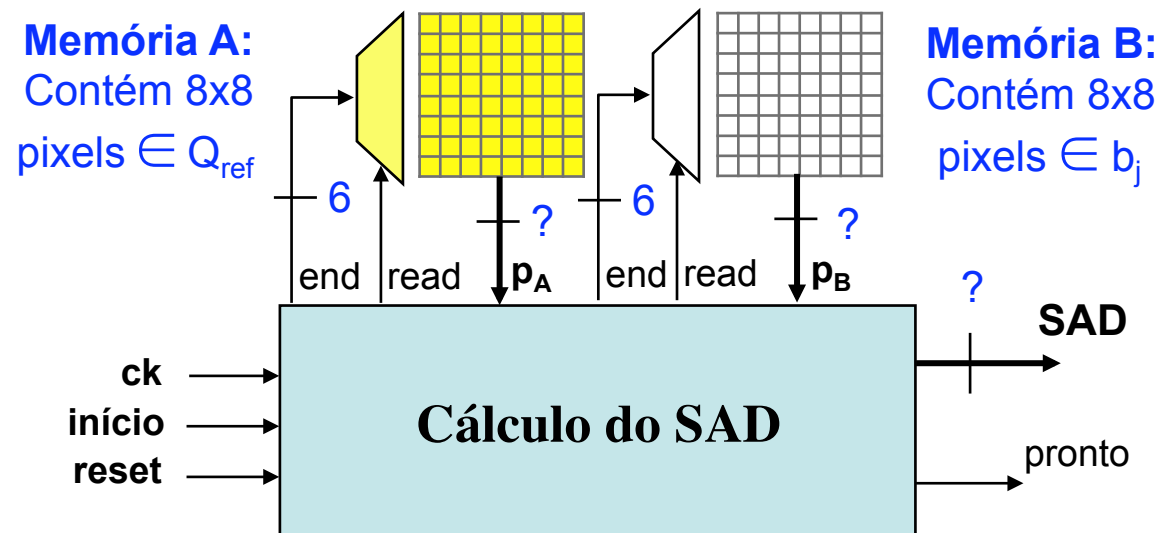


4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- Quantos bits devem ter as entradas p_A e p_B ? **Resp.: 8 bits (=1 pixel)**
- Quantos bits deve ter a saída SAD?



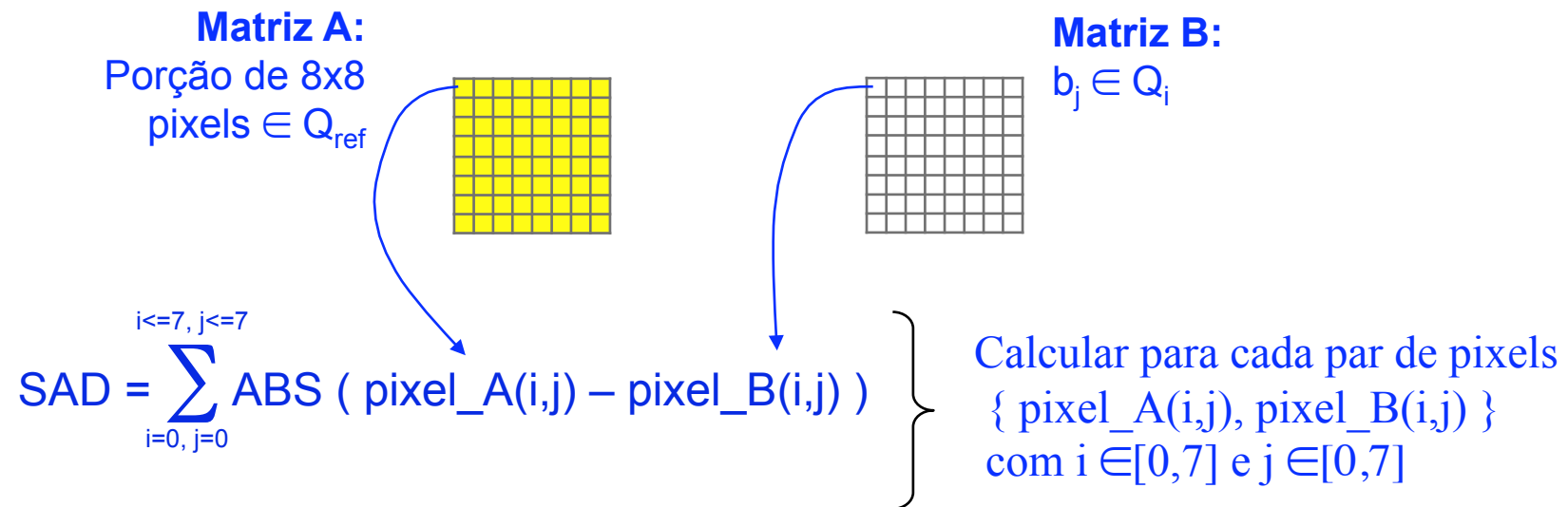
4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- Quantos bits devem ter as entradas p_A e p_B ? **Resp.: 8 bits (=1 pixel)**
- Quantos bits deve ter a saída SAD?

Ou seja, qual é o **valor máximo** que **SAD** pode atingir? Para responder, imaginar que todos os pixels armazenados na matriz A valham zero e todos os pixels armazenados na matriz B valham 255 (ou vice-versa)



4. Projeto de Sistemas Digitais no Nível RT

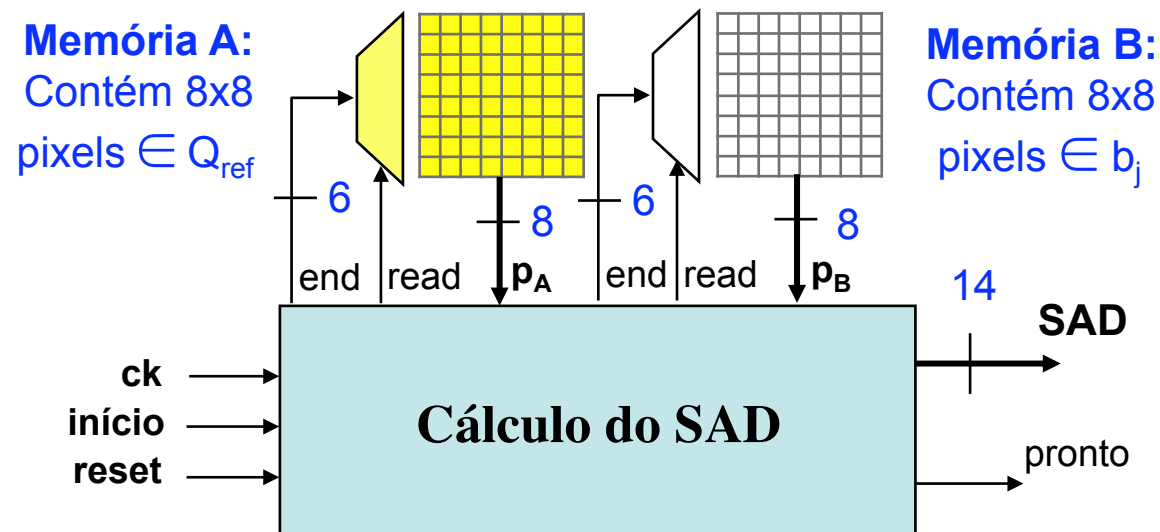
► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- Quantos bits devem ter as entradas p_A e p_B ? **Resp.: 8 bits (=1 pixel)**
- Quantos bits deve ter a saída SAD?

Resp.: $64 \times 255 \sim 2^6 \times 2^8 = 2^{14} \Rightarrow 14$ bits

(O cálculo exato seria $64 \times 255 = 16.320 \Rightarrow 14$ bits...)

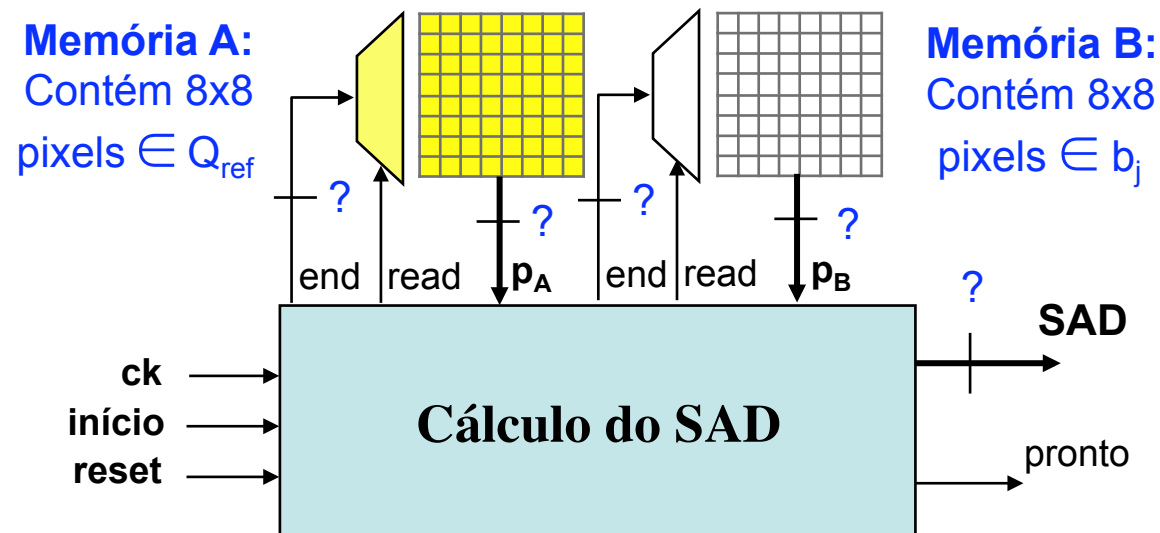


4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- Qual é a faixa de endereços das memórias A e B?
- Quantos bits são necessários para se endereçar uma linha destas memórias?

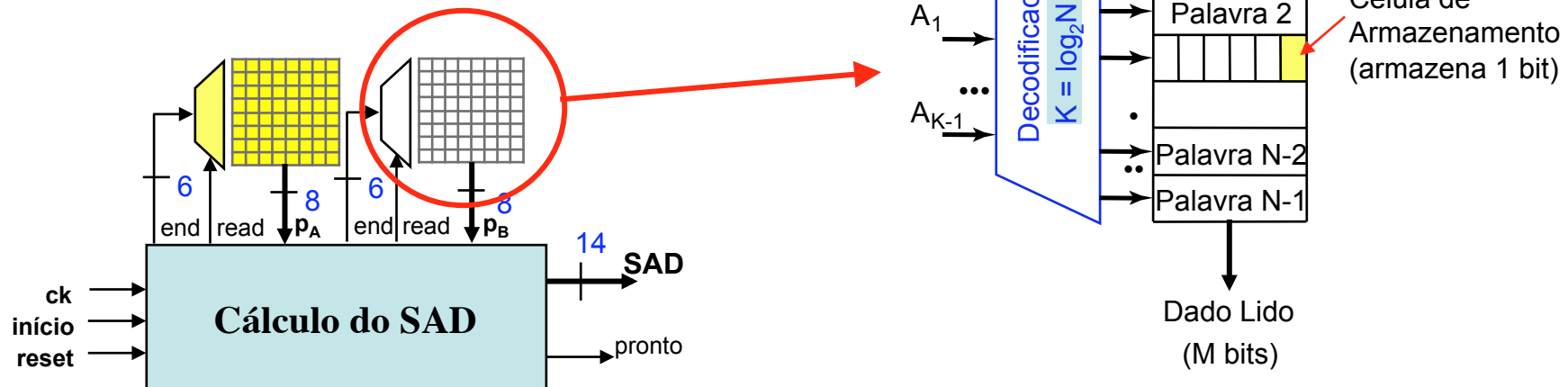


4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

Cada matriz de pixels deverá ser armazenada em uma memória como a mostrada ao lado.



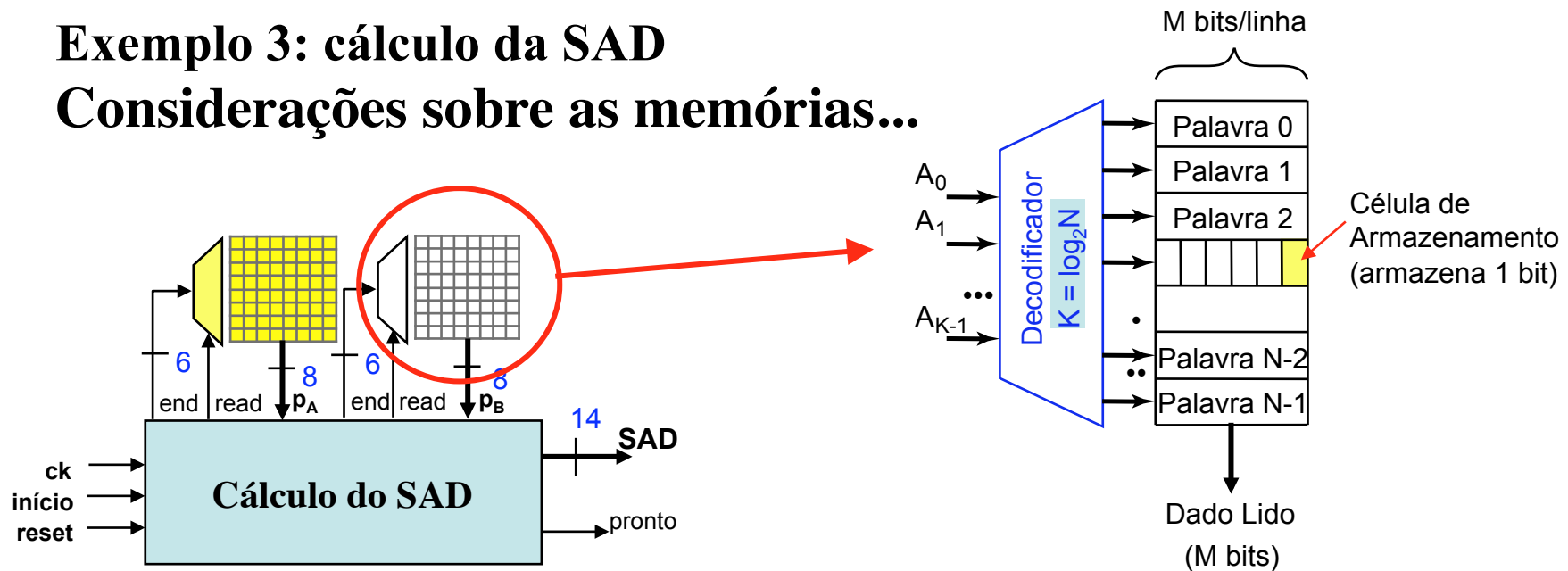
Fonte: RABAEY, Jan M. et al. Digital Integrated Circuits - A Design Perspective. 2nd edition (adaptado)

4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

Considerações sobre as memórias...



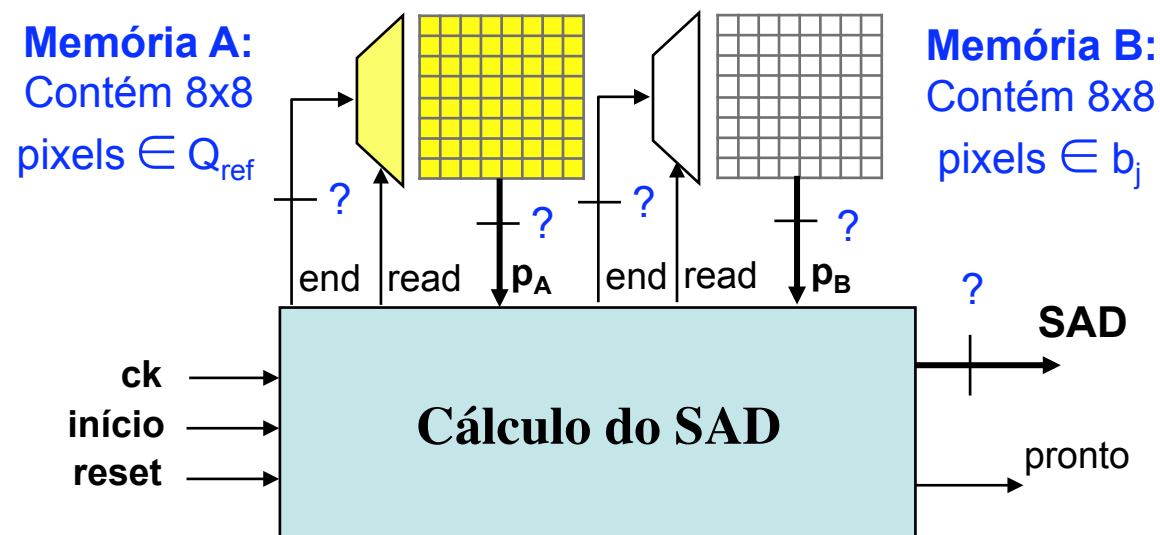
- Cada palavra de memória armazenará um pixel (logo, cada palavra deverá ter 8 bits)
- Total de pixels por matriz = $8 \times 8 = 64$ (logo, serão necessários 64 palavras, uma palavra por linha)
- Cada linha tem um endereço: endereço inicial = 0; endereço final = 63

4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

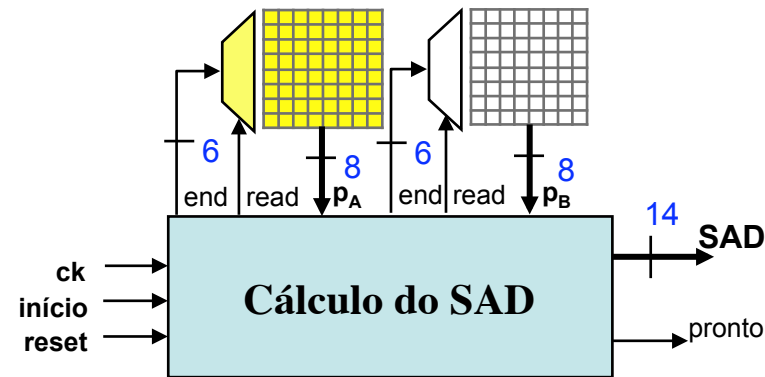
- Qual é a faixa de endereços das memórias A e B? **Resp.: 0 até 63**
- Quantos bits são necessários para se endereçar uma linha destas memórias? **Resp.: 6 bits**



4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: Passo 1 (captura do comportamento com FSMD)



Cálculo a ser feito

$$SAD = \sum_{i=0, j=0}^{i \leq 7, j \leq 7} ABS (pixel_A(i,j) - pixel_B(i,j))$$



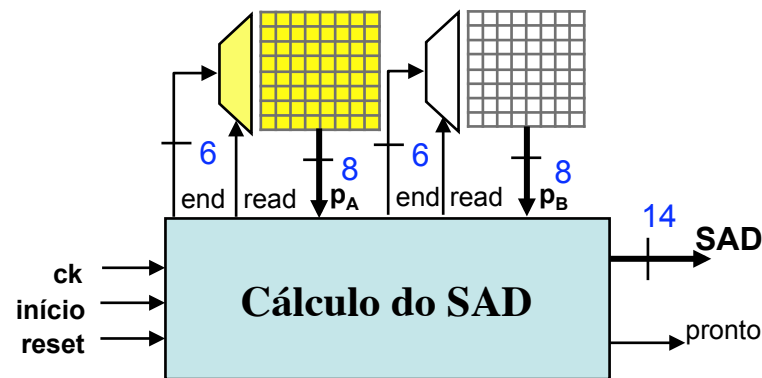
Cálculo expresso como um algoritmo

```
Início
pronto ← 0; soma ← 0; i ← 0;
Enquanto i < 64 faça
{
    soma ← soma + ABS(pA – pB);
    i ← i + 1;
}
SAD_reg ← soma; pronto ← 1;
Fim
```

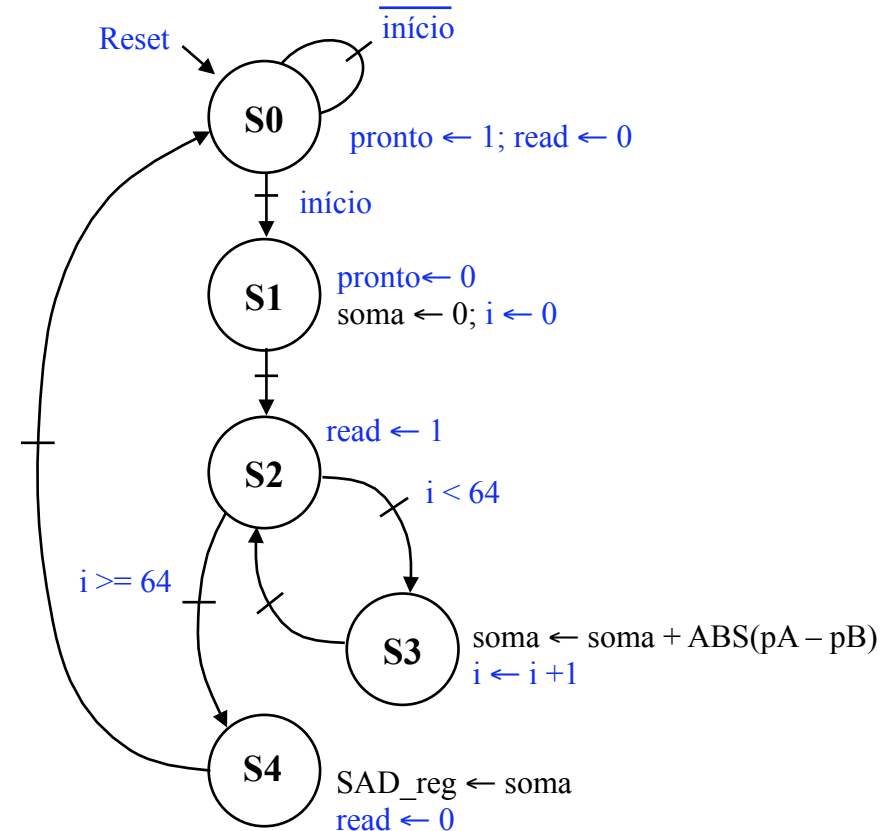
4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

Exemplo 3: Passo 1 (captura do comportamento com FSMD)



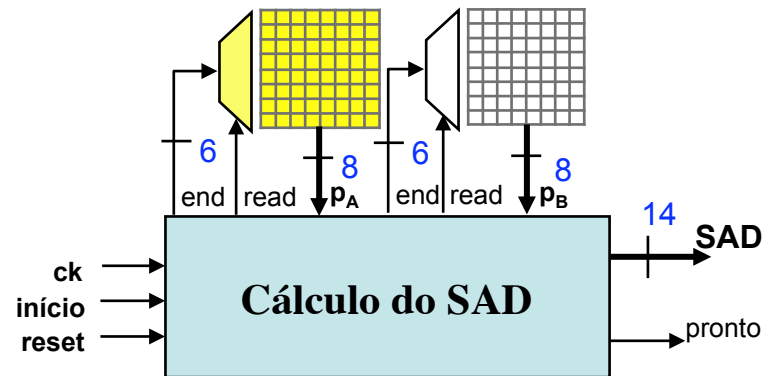
```
Início
pronto ← 0; soma ← 0; i ← 0;
Enquanto i < 64 faça
{
    soma ← soma + ABS(pA - pB);
    i ← i + 1;
}
SAD_reg ← soma; pronto ← 1;
Fim
```



4. Projeto de Sistemas Digitais no Nível RT

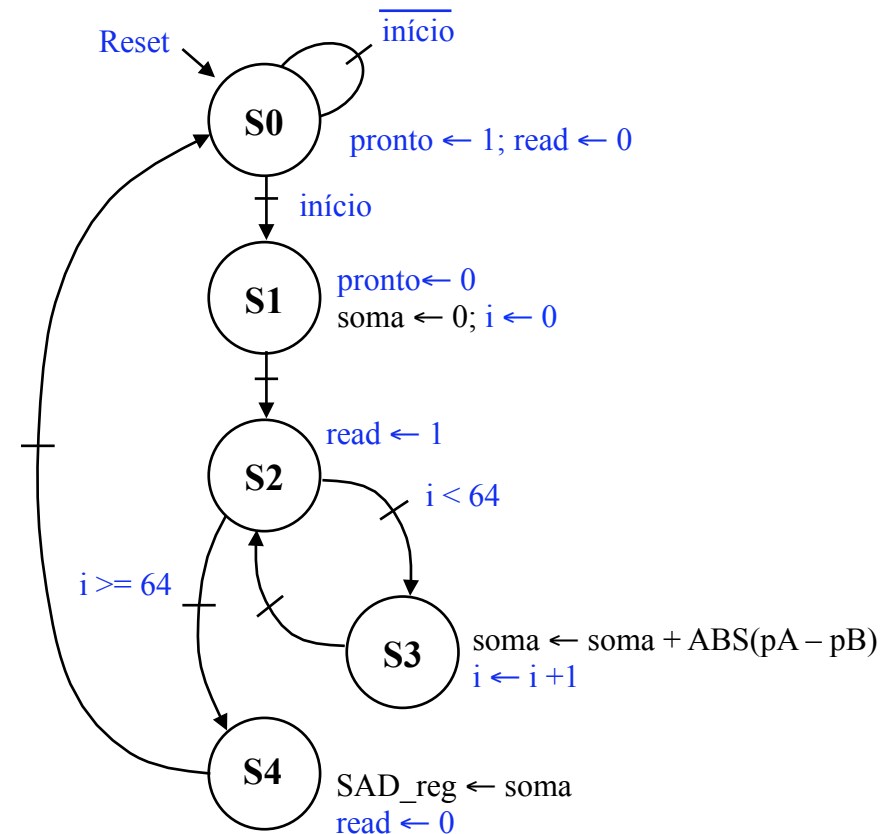
► Projetando um Sistema Digital

Exemplo 3: Passo 2 (projeto do BO)



Quais variáveis são usadas para armazenar dados?

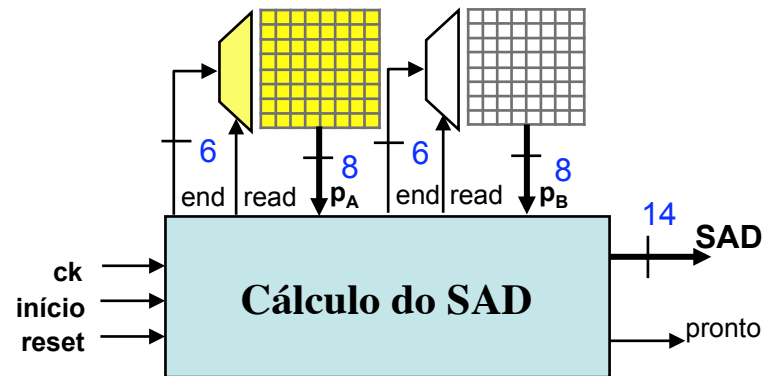
- Duas: “soma” e “SAD_reg” (“pA” e “pB” são entradas que ficam estáveis, já que provem das memória A e B)
- Logo, teremos dois registradores: “soma” e “SAD_reg”



4. Projeto de Sistemas Digitais no Nível RT

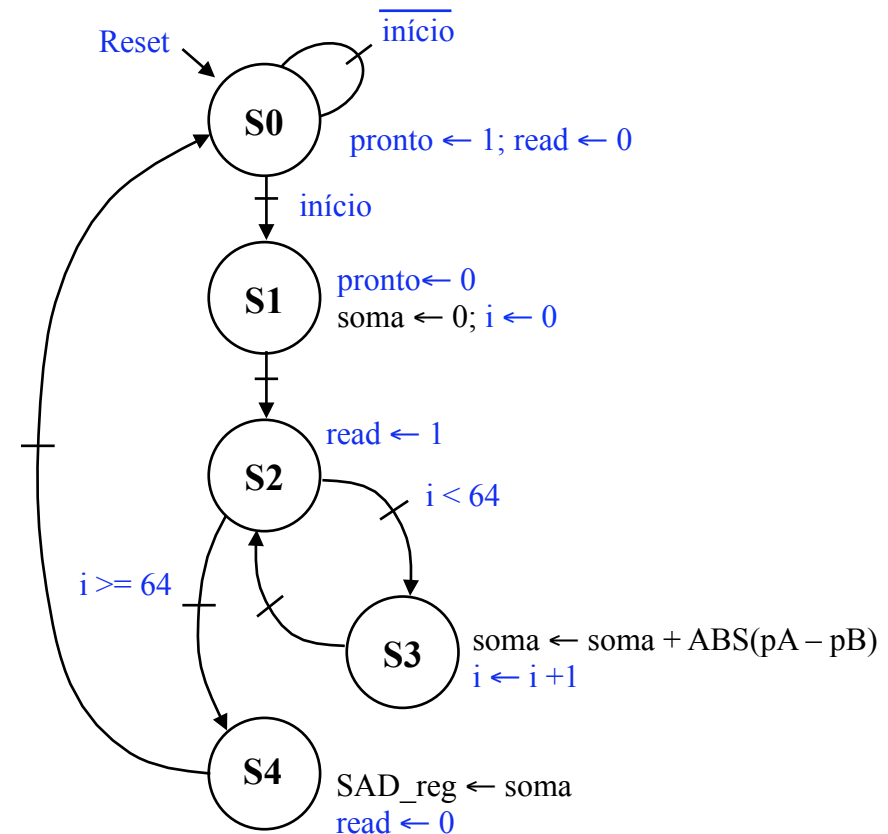
► Projetando um Sistema Digital

Exemplo 3: Passo 2 (projeto do BO)



Porém:

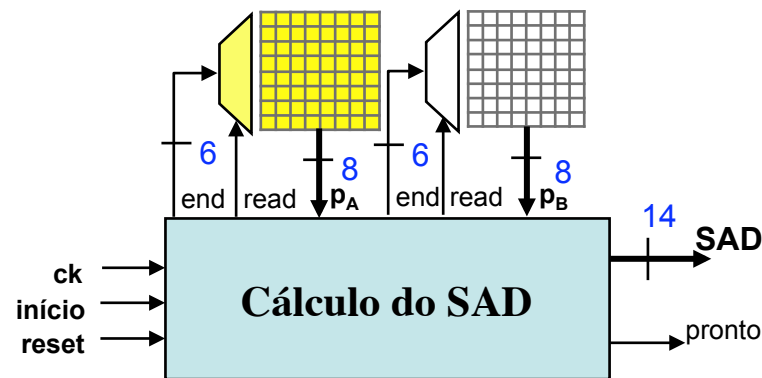
- Note que há uma variável ("i") que armazena o endereço a ser acessado nas memórias e também serve para controlar o laço. Logo, será preciso um registrador (decrementador) para esta variável. Chamemo-lo de "i".



4. Projeto de Sistemas Digitais no Nível RT

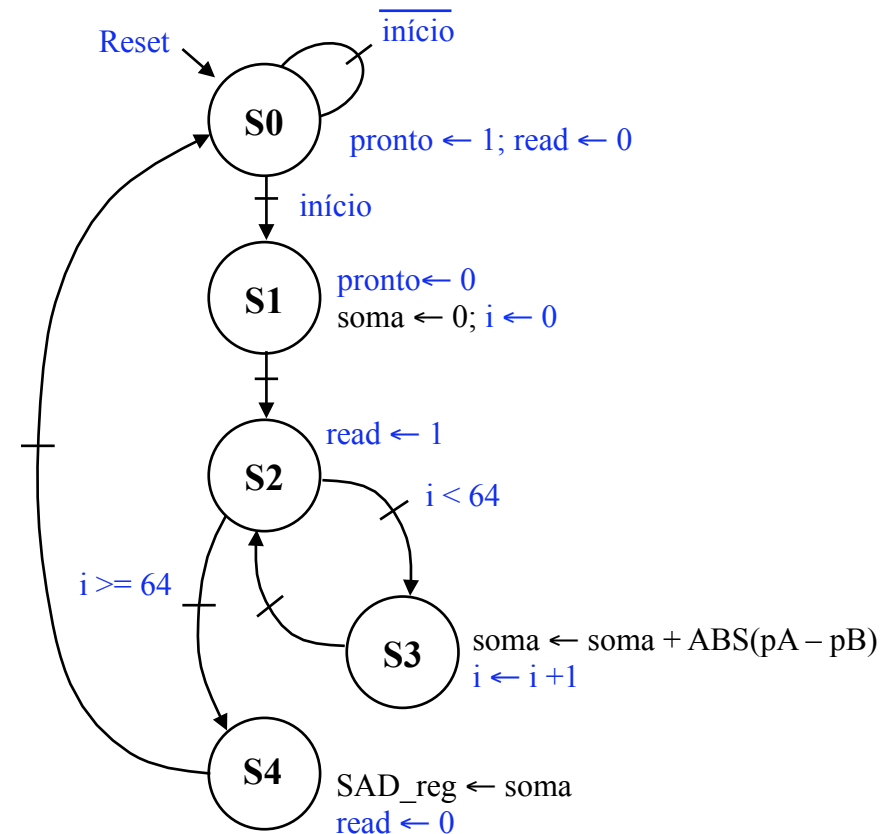
► Projetando um Sistema Digital

Exemplo 3: Passo 2 (projeto do BO)



Quais operações são realizadas sobre dados ?

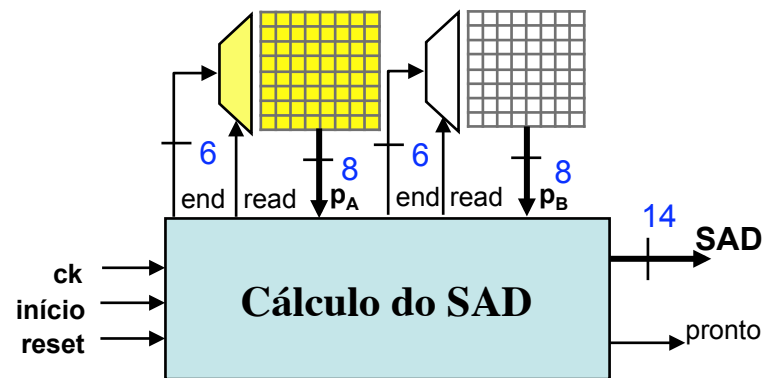
- Dentro do laço há: uma subtração, uma extração de módulo e uma adição.



4. Projeto de Sistemas Digitais no Nível RT

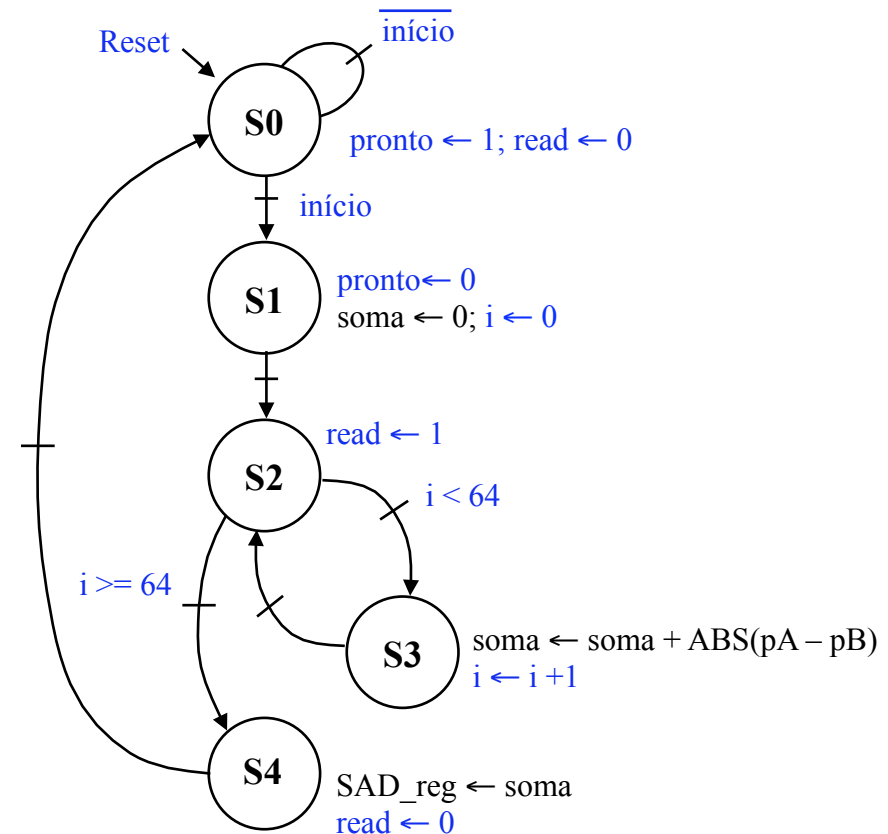
► Projetando um Sistema Digital

Exemplo 3: Passo 2 (projeto do BO)



Quais operações são realizadas sobre variáveis de controle ?

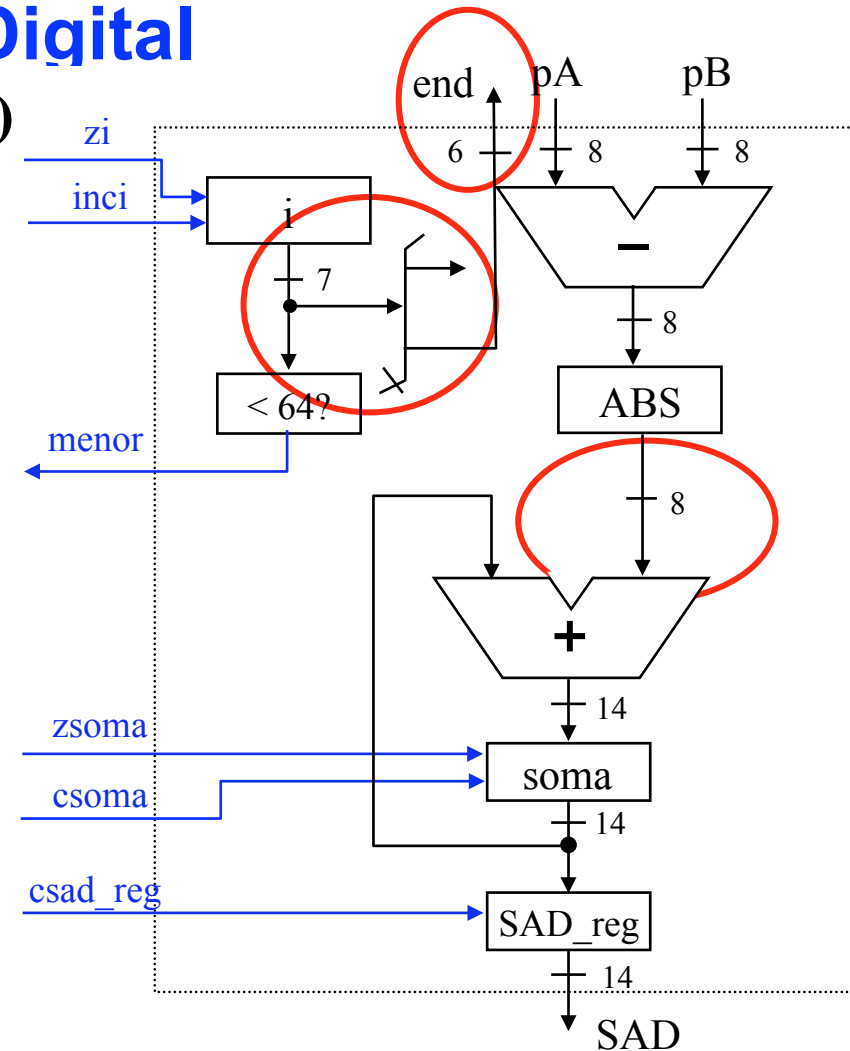
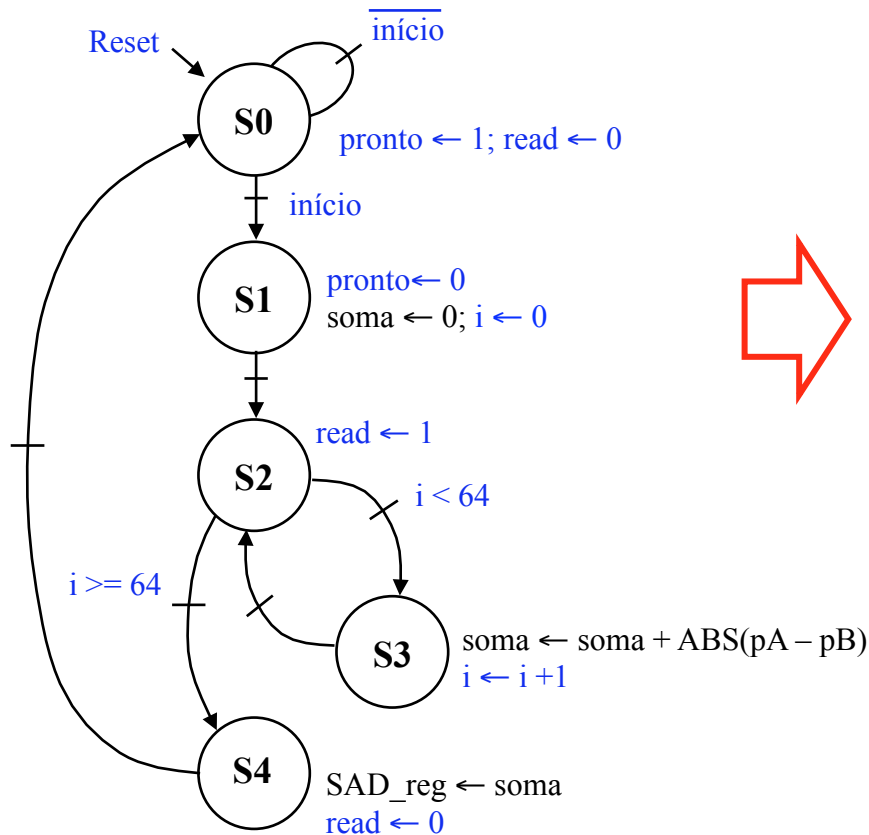
- Dentro do laço há: um incremento sobre "i".
- Também há comparação com 64



4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

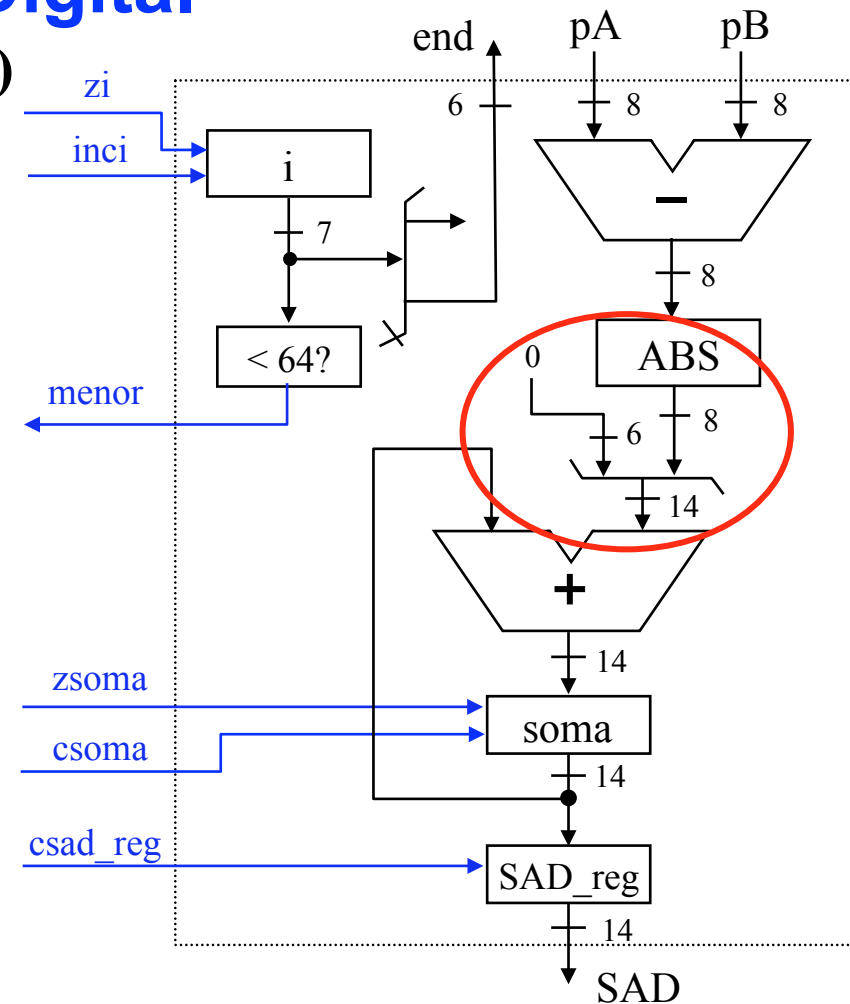
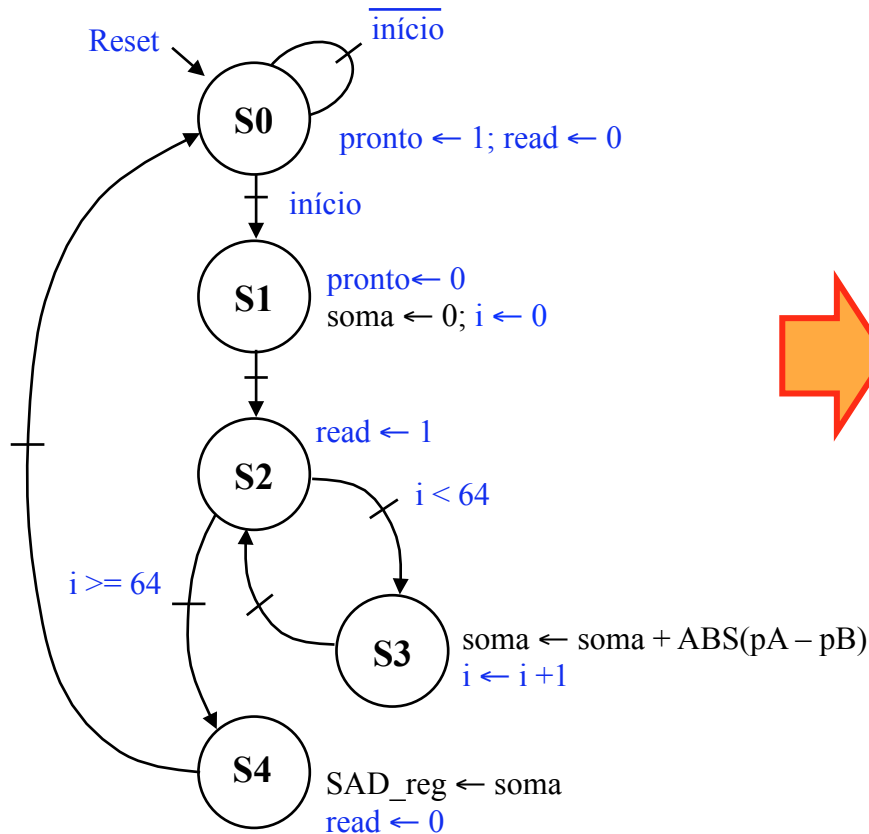
Exemplo 3: Passo 2 (projeto do BO)



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

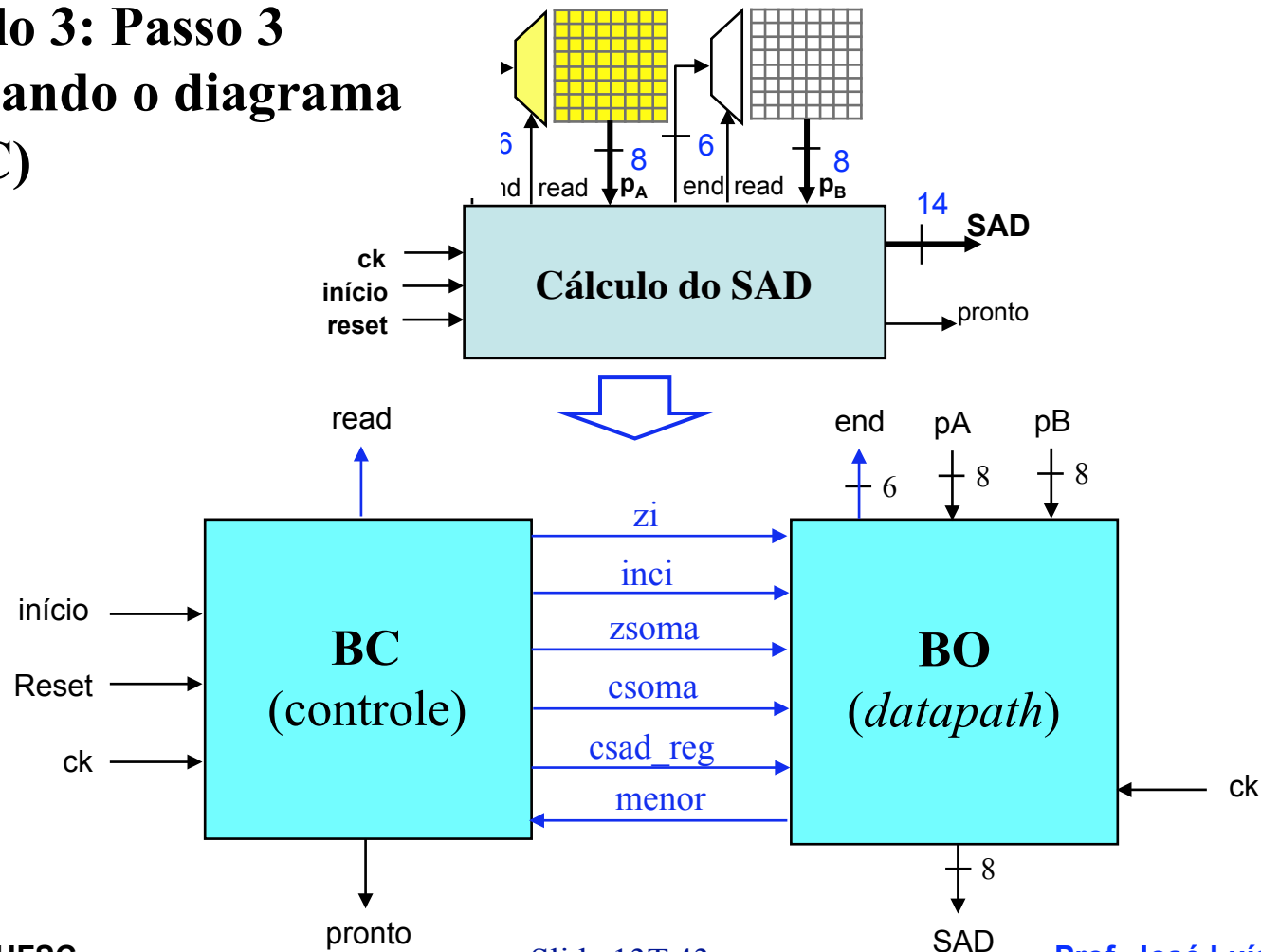
Exemplo 3: Passo 2 (projeto do BO)



4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

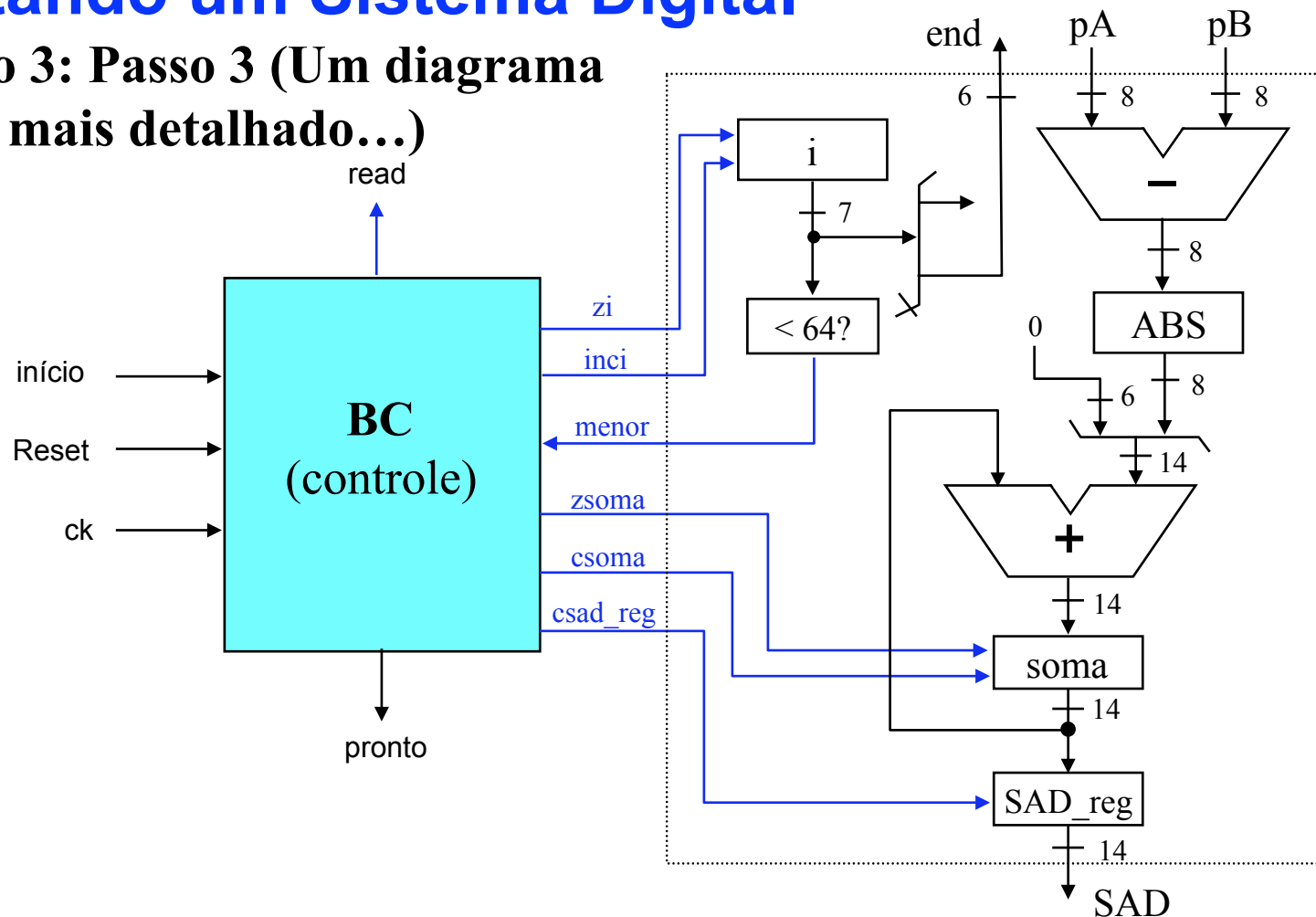
Exemplo 3: Passo 3
(Esboçando o diagrama BO/BC)



4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

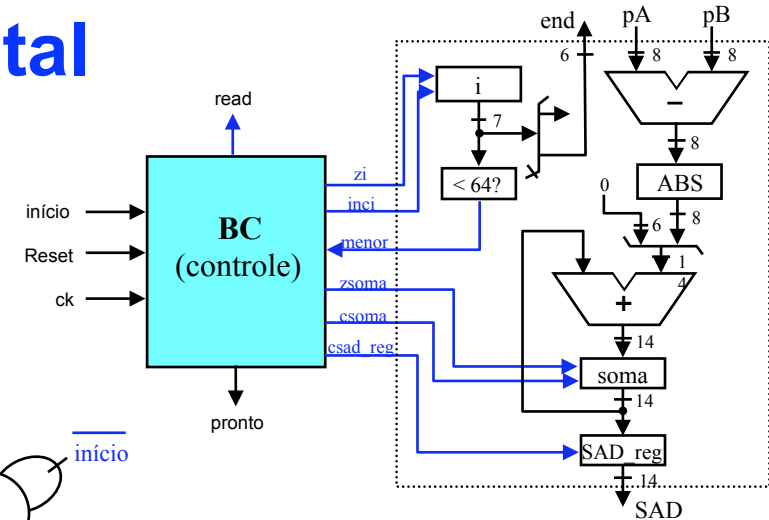
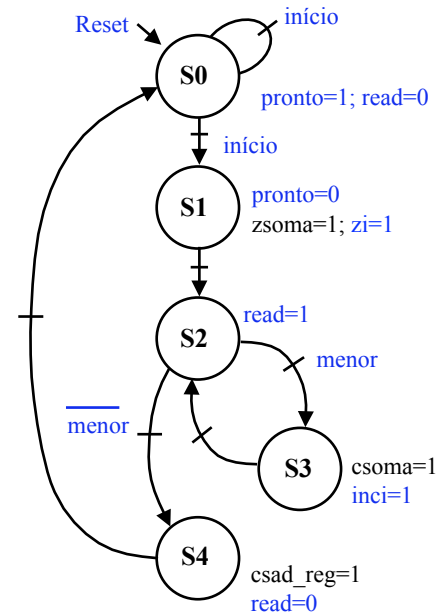
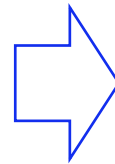
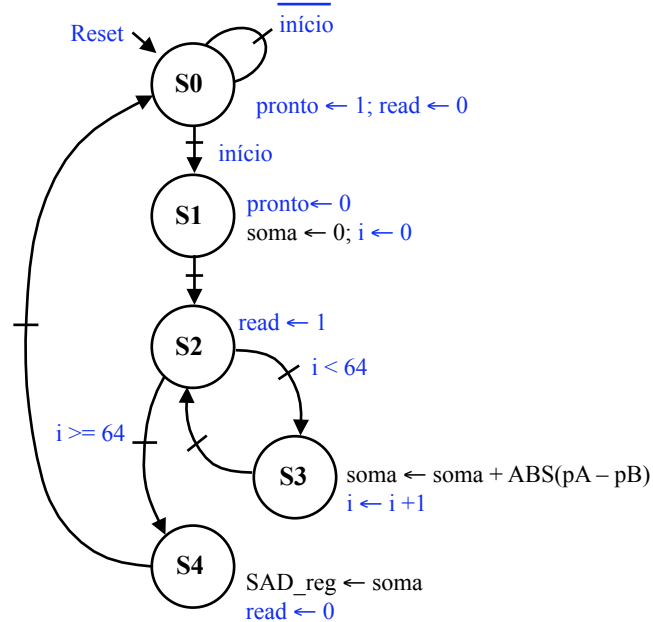
Exemplo 3: Passo 3 (Um diagrama BO/BC mais detalhado...)



4. Projeto de Sistemas Digitais no Nível RT

▶ Projetando um Sistema Digital

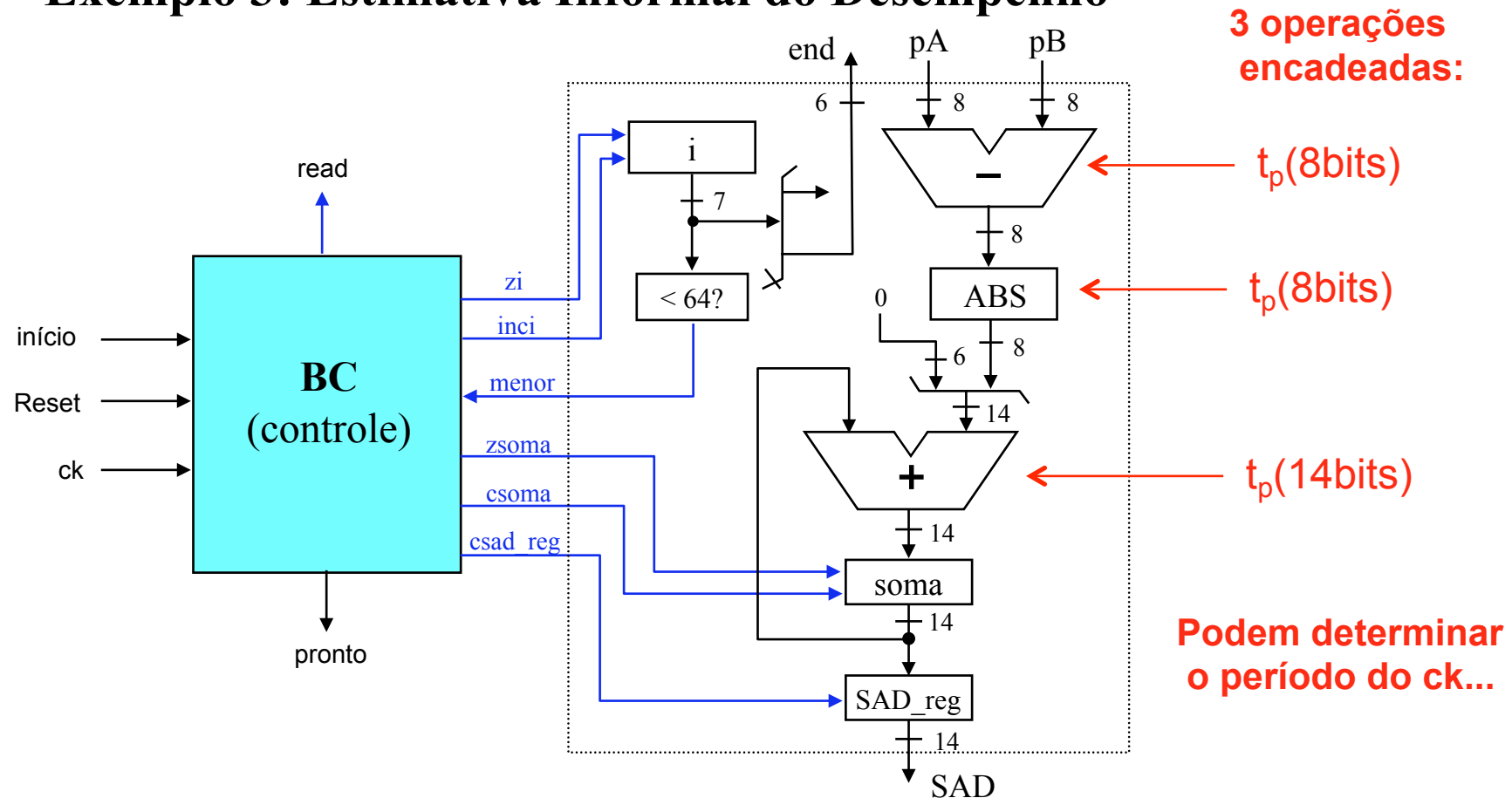
Exemplo 3: Passo 4 (Derivando a FSM a partir do BO e da FSMD)



4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

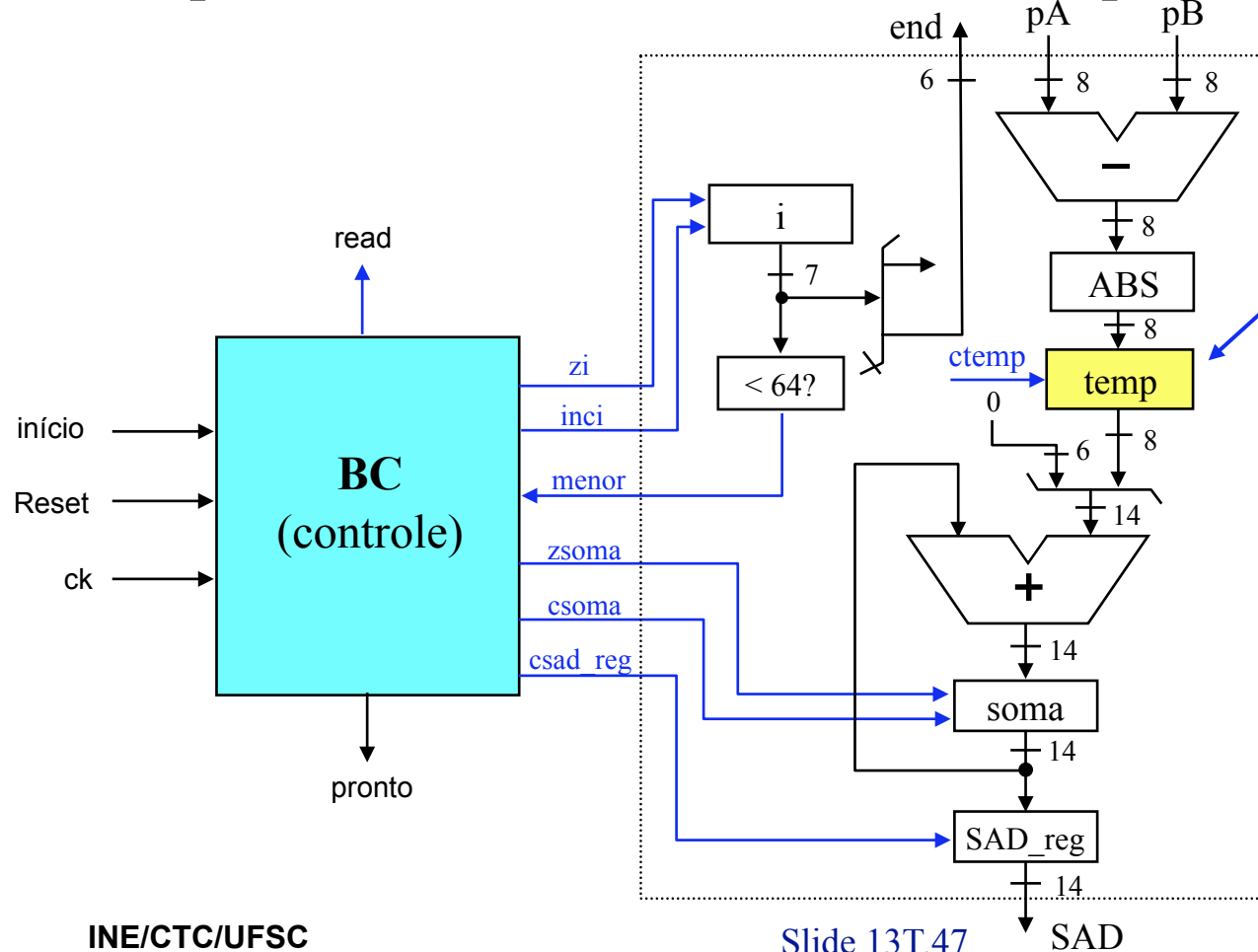
Exemplo 3: Estimativa Informal do Desempenho



4. Projeto de Sistemas Digitais no Nível RT

▶ **Projetando um Sistema Digital**

Exemplo 3: Estimativa Informal do Desempenho



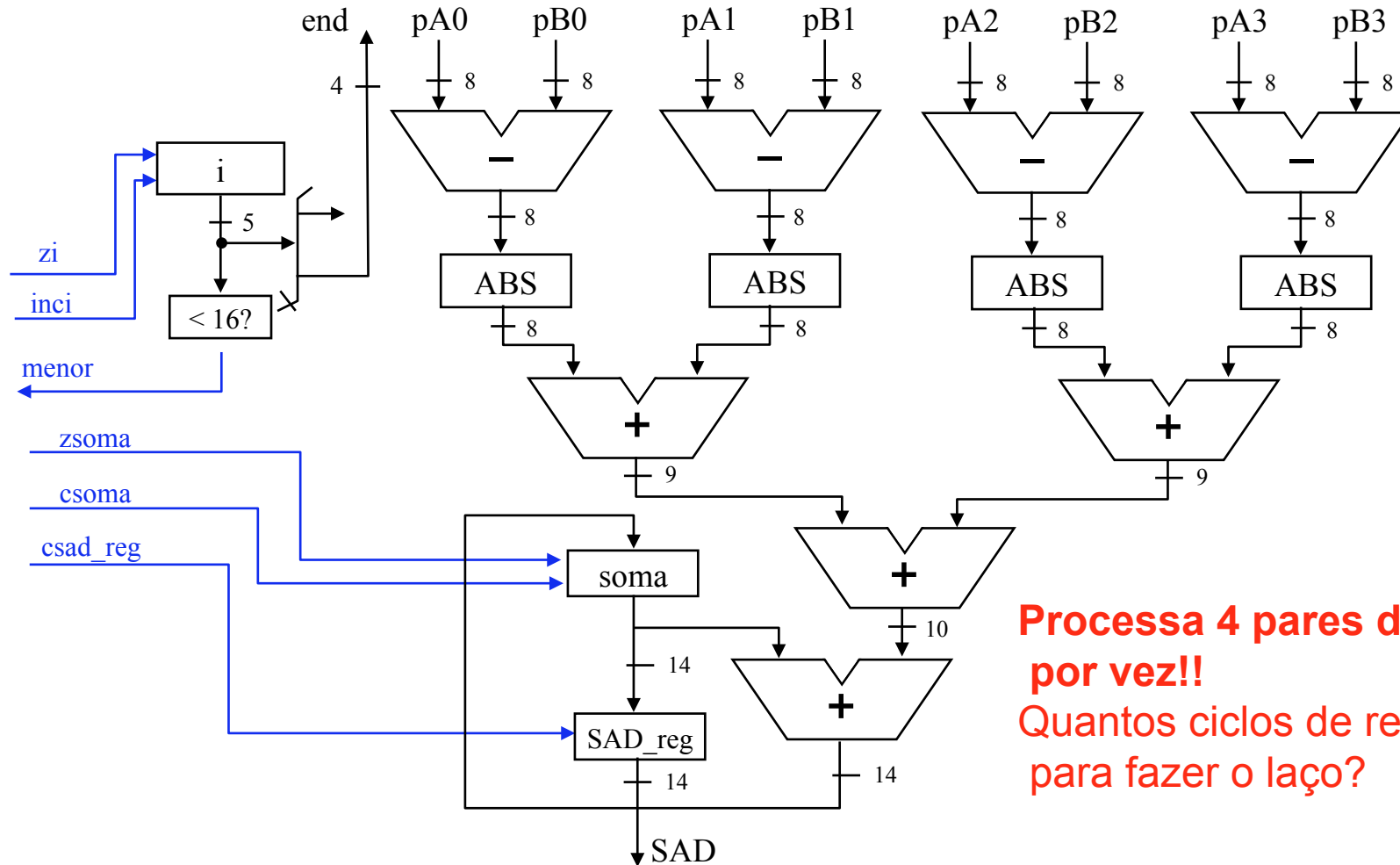
Solução: inserir um registrador (quebrando assim, esta operação em dois ciclos de relógio)

Consequências:

- 1 ciclo a mais dentro do laço (logo, x64)
 - 1 registrador a mais
 - 1 sinal de controle a mais
- (analisar se vale a pena...)

4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

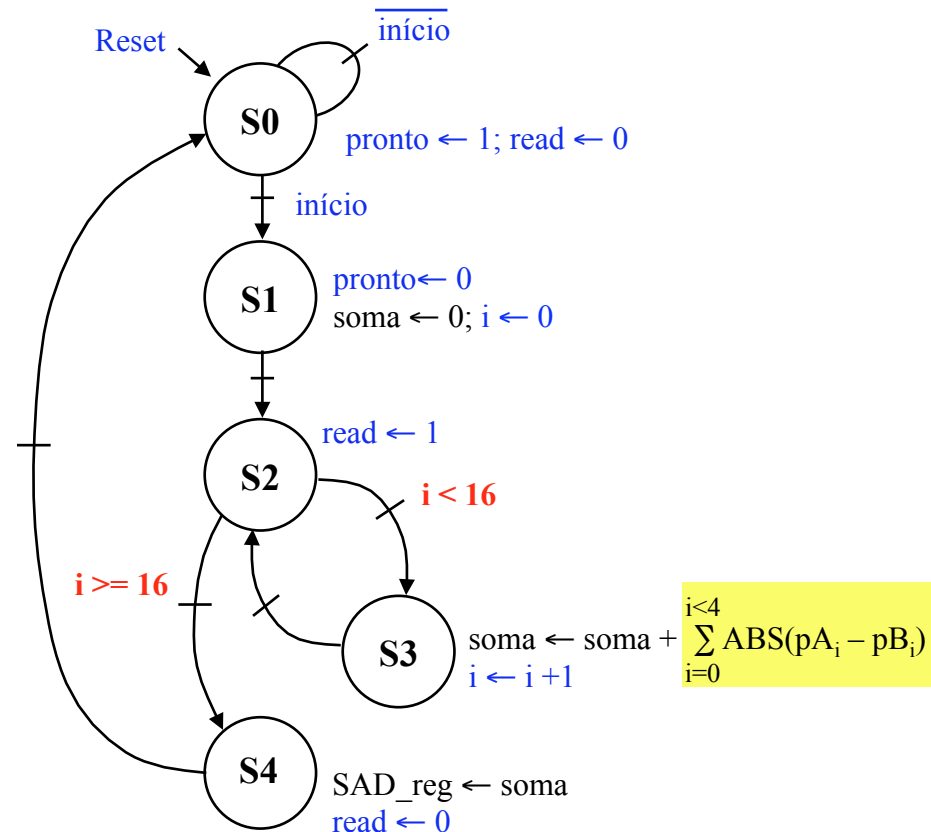


Processa 4 pares de pixels por vez!!
Quantos ciclos de relógio para fazer o laço?

4. Projeto de Sistemas Digitais no Nível RT

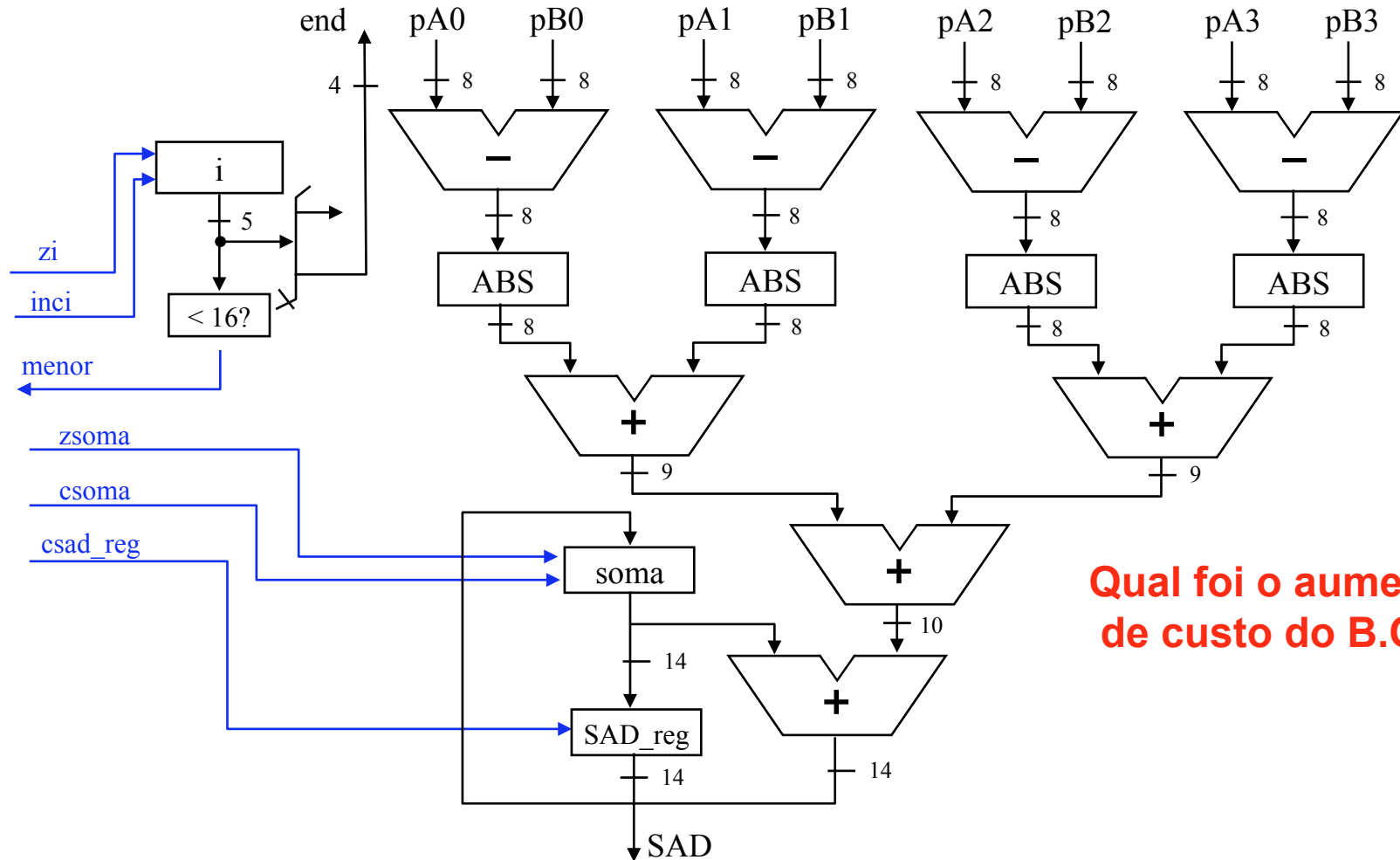
► Projetando um Sistema Digital

Quantos ciclos de relógio para fazer o laço?



4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital



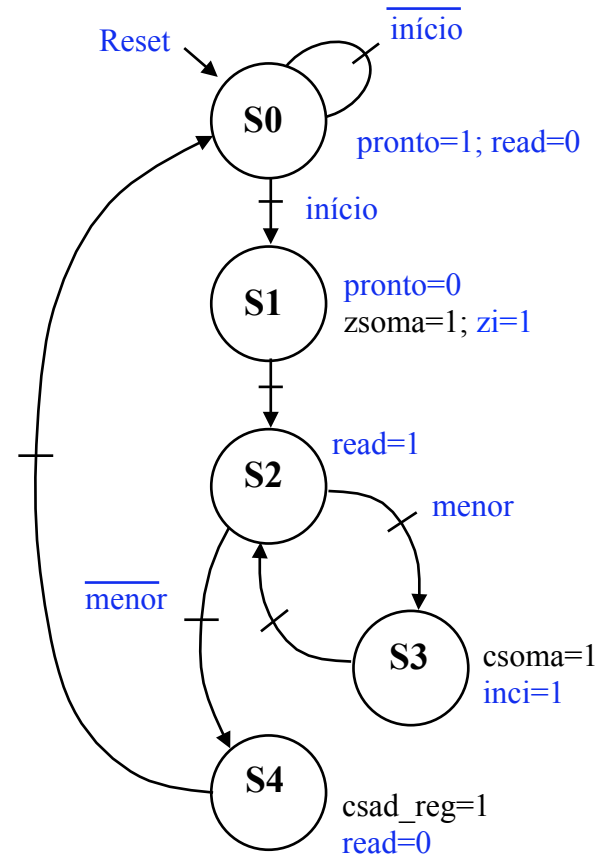
Qual foi o aumento de custo do B.O.?

4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital

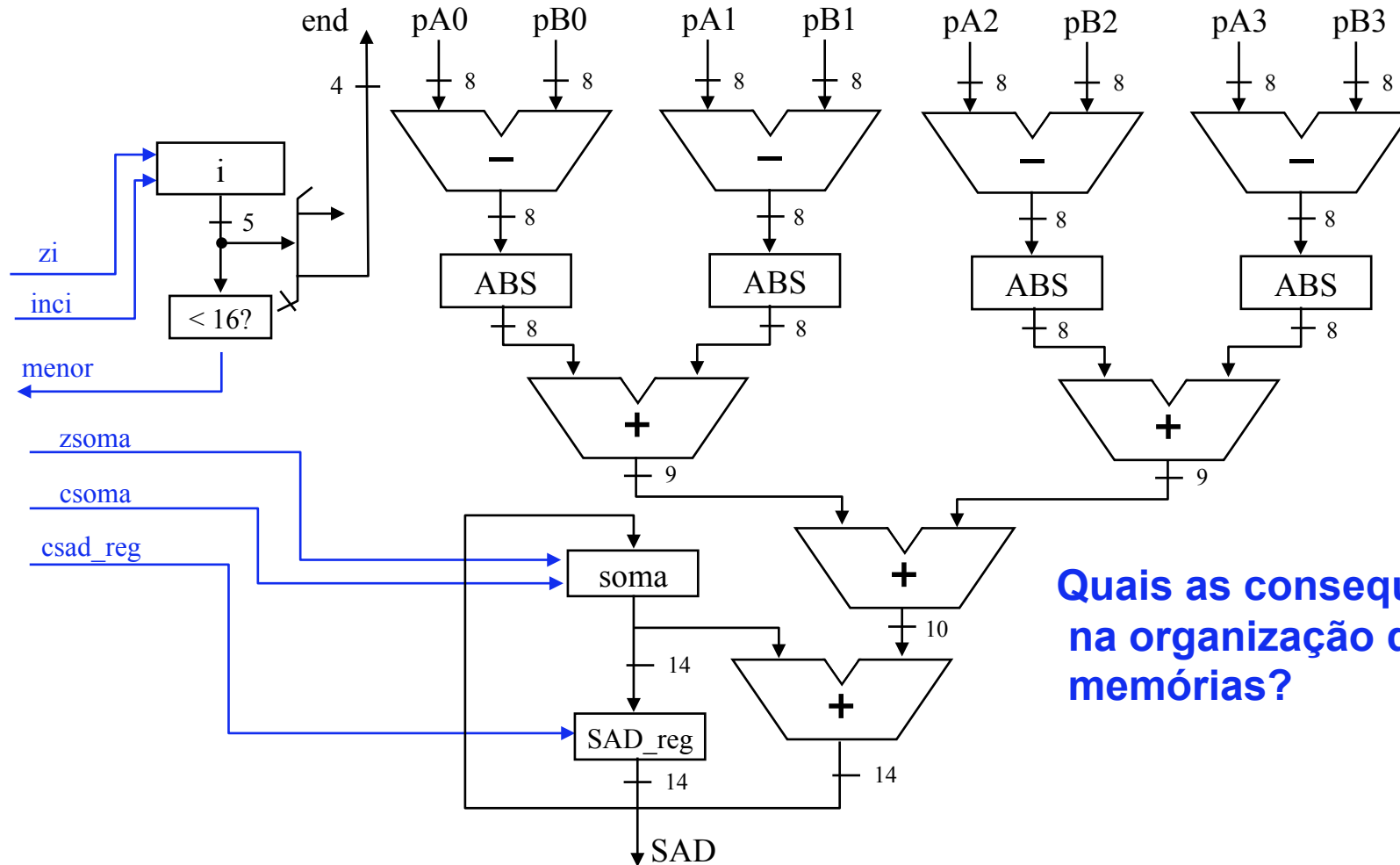
Qual é o reflexo no bloco de controle?

- Número de estados?
- Número de sinais?



4. Projeto de Sistemas Digitais no Nível RT

► Projetando um Sistema Digital



Quais as consequências na organização das memórias?

4. Projeto de Sistemas Digitais no Nível RT

► Indo Mais Além: *Pipeline Aritmético!*

