



Universidade Federal de Santa Catarina

Centro Tecnológico

Departamento de Informática e Estatística
Ciências da Computação & Engenharia Eletrônica



Sistemas Digitais

INE 5406

Aula 6-T

2. Processadores Dedicados (Blocos Aceleradores). Estudo de caso: compressão de vídeo.

**Profs. José Luís Güntzel e Cristina Meinhardt
& Est. Docência Marcio Monteiro**

{j.guntzel, cristina.meinhardt}}@ufsc.br, marcio@inf.ufsc.br

Princípios de Compressão de Vídeo

What happens in an internet minute?

2018 *This Is What Happens In An Internet Minute*



2019 *This Is What Happens In An Internet Minute*



Fonte: <https://www.visualcapitalist.com/what-happens-in-an-internet-minute-in-2019/>

Princípios de Compressão de Vídeo



- ✓ Um vídeo é uma sequência de imagens (chamadas *frames* ou quadros) amostradas rapidamente;
- ✓ A velocidade da amostragem engana o cérebro, criando a ilusão de movimento.

Foreman 30fps →
(30 imagens exibidas a cada segundo)

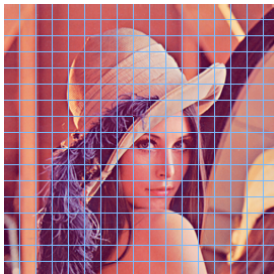


Princípios de Compressão de Vídeo



Bloco de 4x4 pixel

Macrobloco de 16x16 pixel



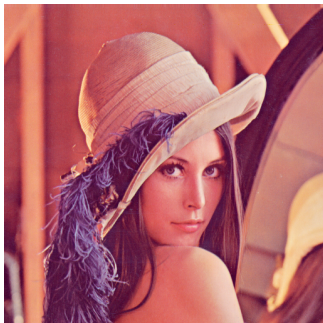
8 bits 8 bits 8 bits

Representação binária do pixel
no espaço de cores Y Cb Cr

- Dividindo em macroblocos 16x16;
- Dividindo em blocos 4x4;
- Para facilitar o processamento de imagens em diferentes tamanhos, elas são divididas em macroblocos, e estes em blocos;

Princípios de Compressão de Vídeo

Armazenamento



512 pixels

Quantidade de bytes para armazenar esta imagem:

$512 * 512 \text{ pixels} = 262144 \text{ pixels}$

$262144 \text{ pixels} * 3 \text{ bytes} = 786432 \text{ bytes} = 768 \text{ KiB}$

Princípios de Compressão de Vídeo

Formato DVD (Armazenamento)



- Resolução padrão de um quadro de DVD: **704 x 480** pixels/quadro

$704 \times 480 \text{ pixels/quadro} \times 3 \text{ bytes/pixel} = \mathbf{990 \text{ KiB por quadro}}$

- Um filme em formato DVD (Filmes tem pelo menos **24 quadros/s**)

Supondo um filme com 94 minutos:

$990 \text{ KiB por quadro} \times 24 \text{ quadros/s} \times 5640 \text{ s} = 134217728 \text{ KiB} =$
 $131072 \text{ MiB} = \mathbf{128 \text{ GiB!!!}}$

Mas em um DVD-DL só cabem 7,96 GiB !! E agora???

Solução (paliativo): comprimir 16x (taxa de compressão = 16:1)

Princípios de Compressão de Vídeo

Formato Blu-ray



Capacidade média *Blu-ray* de dupla camada: 46.6GiB

Problema 1:
armazenamento

Problema 2:
transmissão

"RAW"	Resolução	Taxa de quadros	90 min. Vídeo	Taxa de bits
HD 1080p@ 24fps	1920x1080	24/s progressivo	750,85 GiB!!!	1,19 Gbps

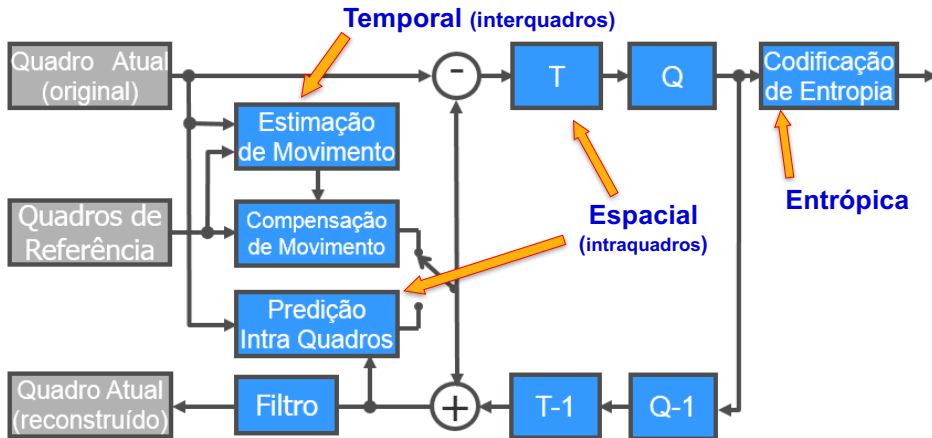
Solução (paliativo):

Compressão necessária mantendo a qualidade: 16:1

Princípios de Compressão de Vídeo



Princípios de Compressão de Vídeo



Princípios de Compressão de Vídeo

Previsão Interquadros



Q_{ref} é um “quadro de referência”:

Todos os seus pixels serão armazenados.

Princípios de Compressão de Vídeo

Previsão Interquadros



Para um dos demais quadros Q_i :

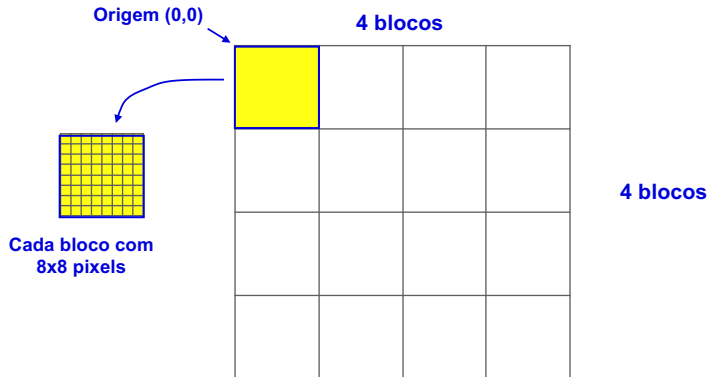
Para cada bloco de pixels de Q_i :

Será buscada dentro de Q_{ref} a porção de pixels com melhor casamento (i.e., a mais parecida)

Princípios de Compressão de Vídeo

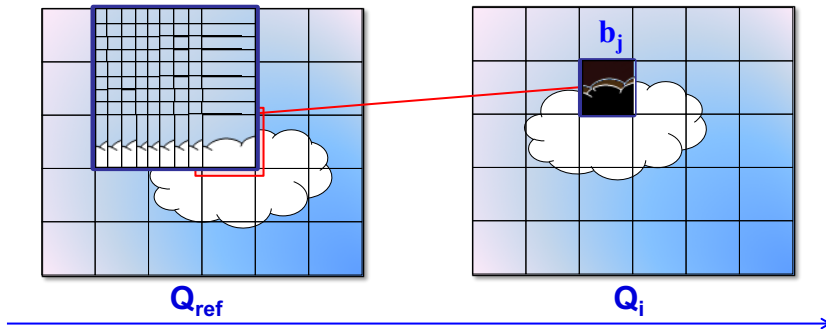
Representação Simplificada de um Quadro (*frame*)

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



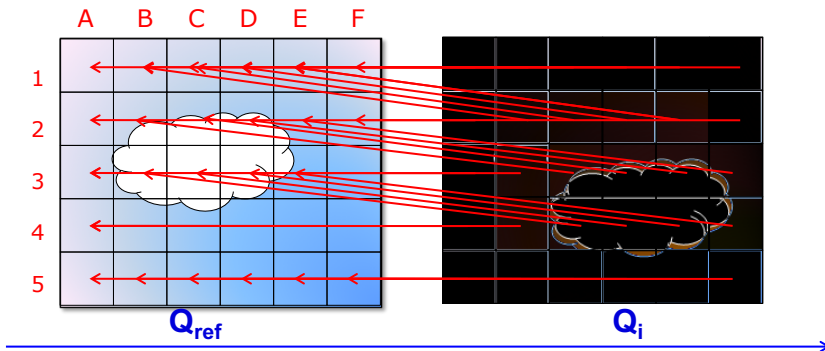
Princípios de Compressão de Vídeo

Estimação de Movimento:



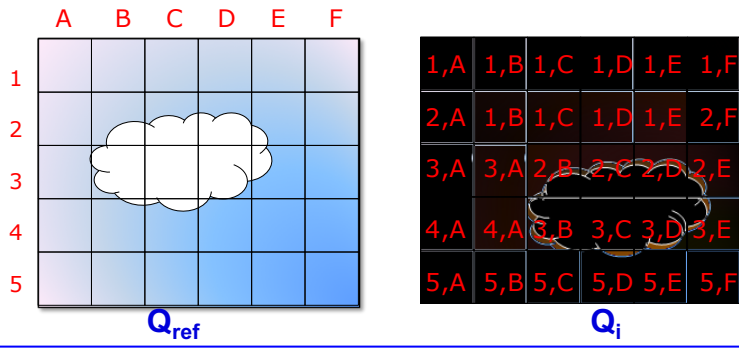
- Cada vez que se compara um bloco $b_j \in Q_i$ com um bloco de Q_{ref} é preciso calcular a similaridade entre duas matrizes de pixels

Princípios de Compressão de Vídeo



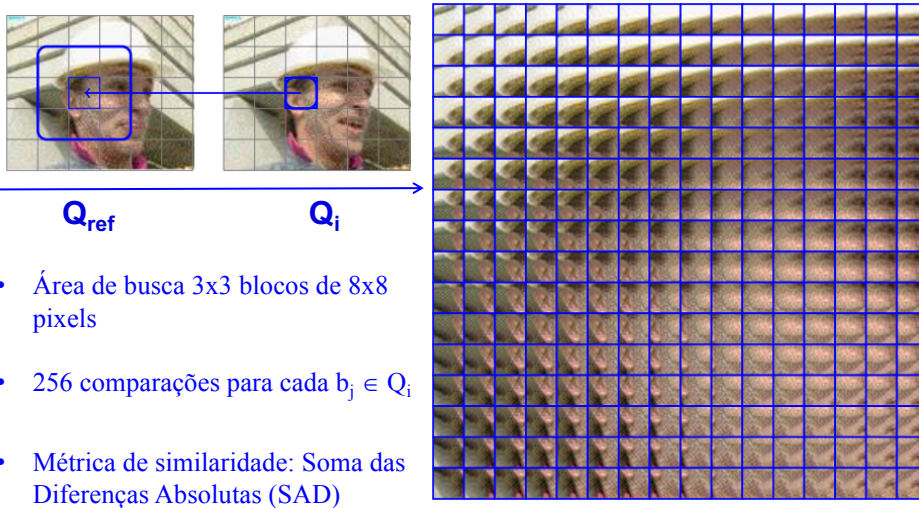
- Cada bloco de $b_j \in Q_i$ será representado por um vetor de movimento (ao invés de 8x8 pixels) + resíduo (diferença entre o bloco $b_j \in Q_i$ e a sub-matriz $b_{ref} \in Q_{ref}$)

Princípios de Compressão de Vídeo



- Cada bloco de $b_j \in Q_i$ será representado por um vetor de movimento (ao invés de 8x8 pixels) + resíduo (diferença entre o bloco $b_j \in Q_i$ e a sub-matriz $b_{ref} \in Q_{ref}$)

Princípios de Compressão de Vídeo



Processadores Dedicados

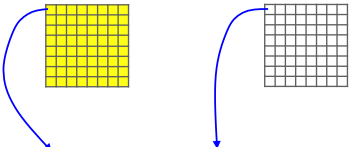
Projetando um Sistema Digital

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
($n \times n$)


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

Processadores Dedicados

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.

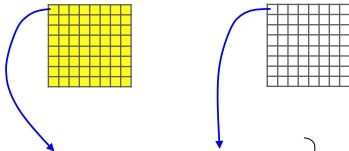
Processadores Dedicados

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 $j \in [0,7]$

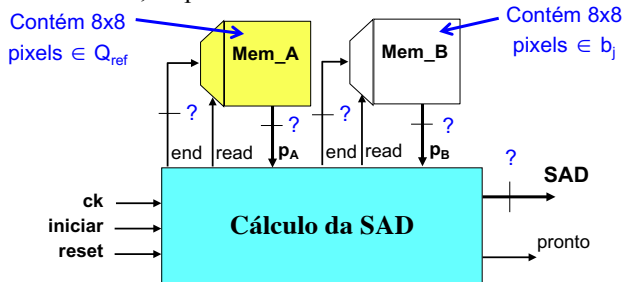
Processadores Dedicados

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.

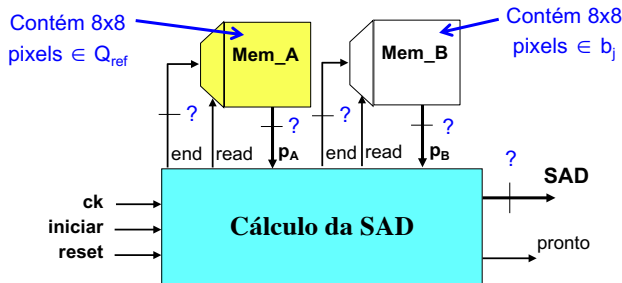


Processadores Dedicados

Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- O sinal iniciar=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.

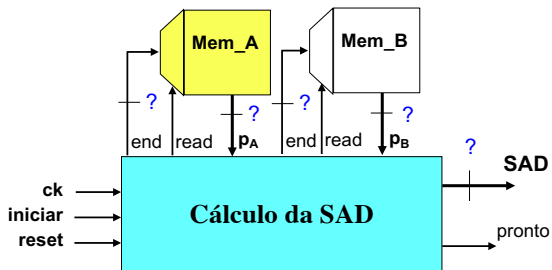


Processadores Dedicados

Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- Quantos bits deve ter a saída SAD?



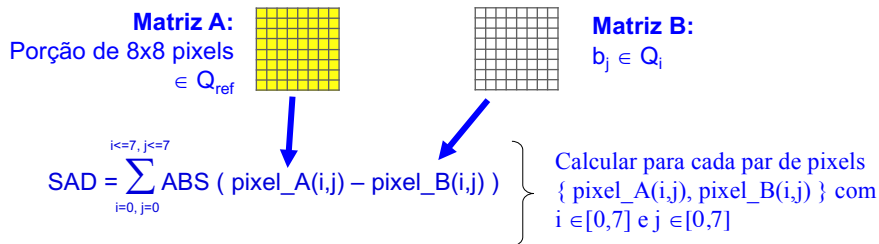
Processadores Dedicados

Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- 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)



Processadores Dedicados

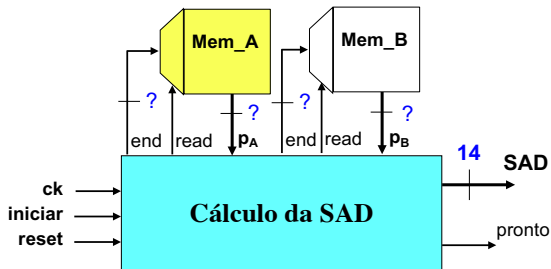
Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

- Quantos bits deve ter a saída SAD?

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

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

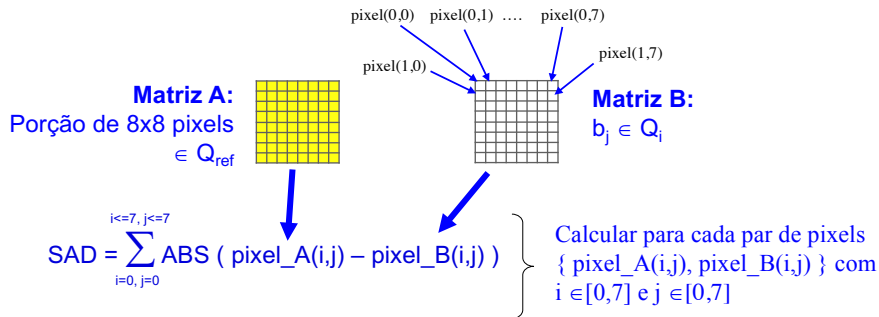


Processadores Dedicados

Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

Possível Organização da “Memória A” e da “Memória B”

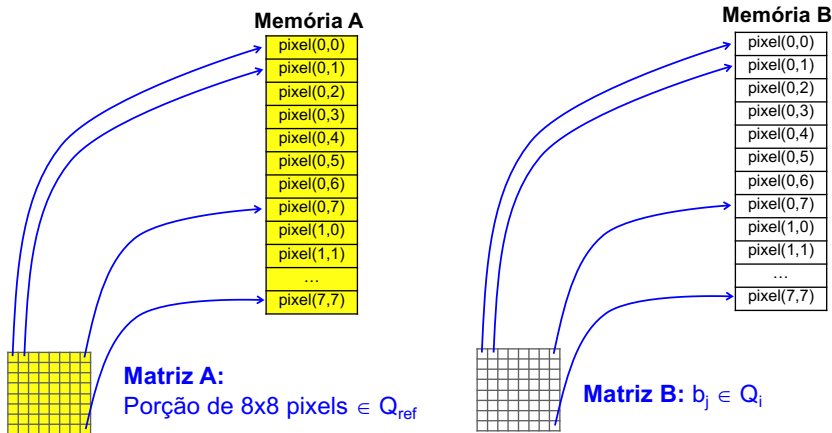


Processadores Dedicados

Projetando um Sistema Digital

Exemplo 3: cálculo da SAD

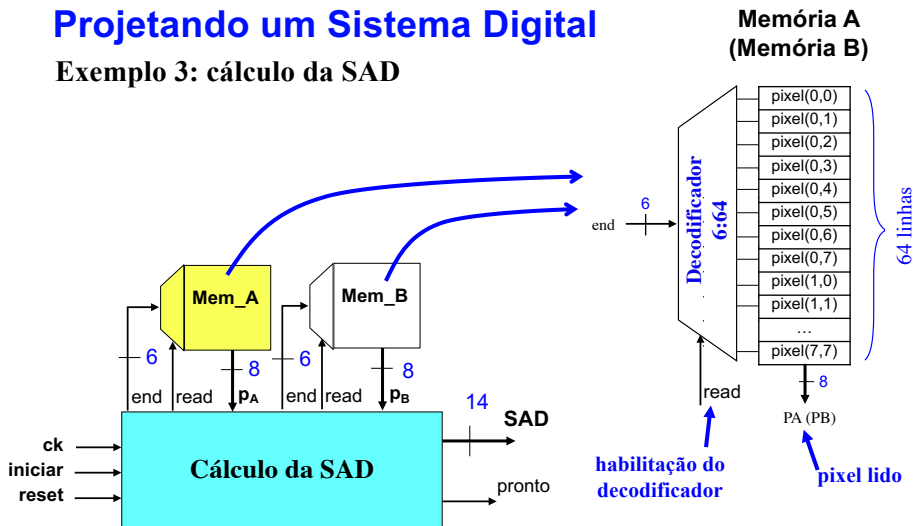
Possível Organização da “Memória A” e da “Memória B”



Processadores Dedicados

Projetando um Sistema Digital

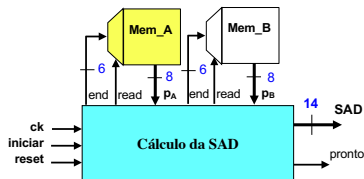
Exemplo 3: cálculo da SAD



Processadores Dedicados

Projetando um Sistema Digital

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



Cálculo a ser feito

$$SAD = \sum_{i=0}^{i \leq 64} ABS (pixel_A(end) - pixel_B(end))$$



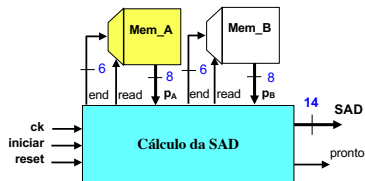
Cálculo expresso como um algoritmo

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

Processadores Dedicados

Projetando um Sistema Digital

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



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

Observar que:

Para cada $end \in \{0, 1, \dots, 63\}$:

$pA = Mem_A[end]$

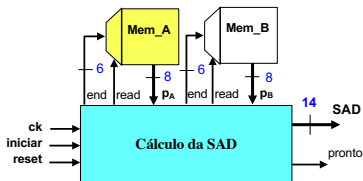
$pB = Mem_B[end]$

Onde $Mem_A[end]$ é um acesso de leitura à posição “end” de Memória A

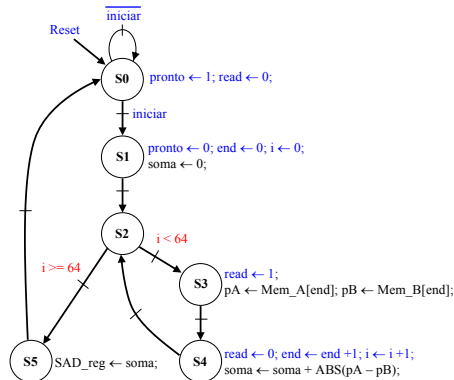
Processadores Dedicados

Projetando um Sistema Digital

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



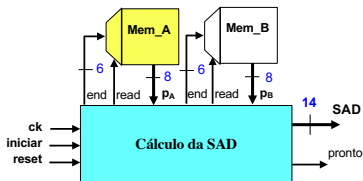
```
Inicio
pronto ← 0; soma ← 0; i ← 0; end ← 0;
Enquanto i < 64 faça
{
    pA ← Mem_A[end]; pB ← Mem_B[end];
    soma ← soma + ABS(pA - pB); end ← end + 1; i ← i + 1;
}
SAD_reg ← soma;
pronto ← 1;
Fim
```



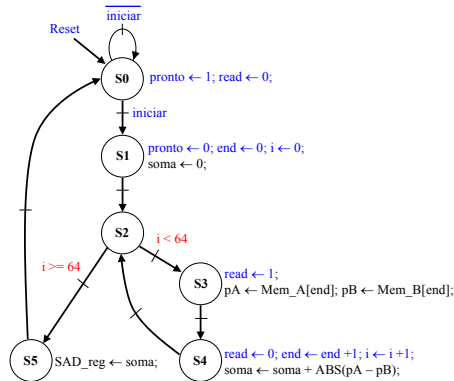
Processadores Dedicados

Projetando um Sistema Digital

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



```
Inicio
pronto ← 0; soma ← 0; i ← 0; end ← 0;
Enquanto i < 64 faça
{
  pA ← Mem_A[end]; pB ← Mem_B[end];
  soma ← soma + ABS(pA - pB); end ← end + 1; i ← i + 1;
}
SAD_reg ← soma;
pronto ← 1;
Fim
```

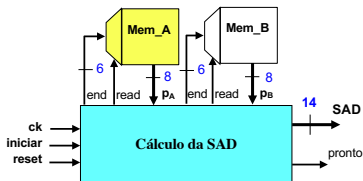


Por que “pronto ← 1” está em S0 (e não em S5)?

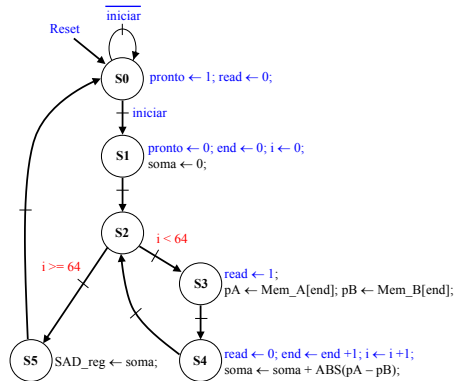
Processadores Dedicados

Projetando um Sistema Digital

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



```
Inicio
pronto ← 0; soma ← 0; i ← 0; end ← 0;
Enquanto i < 64 faça
{
  pA ← Mem_A[end]; pB ← Mem_B[end];
  soma ← soma + ABS(pA - pB); end ← end + 1; i ← i + 1;
}
SAD_reg ← soma;
pronto ← 1;
Fim
```

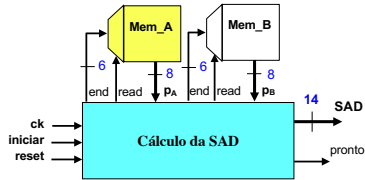


Por que “pronto ← 1” está em S0 (e não em S5)?
Resp.: porque a carga de SAD_reg só termina em S0.

Processadores Dedicados

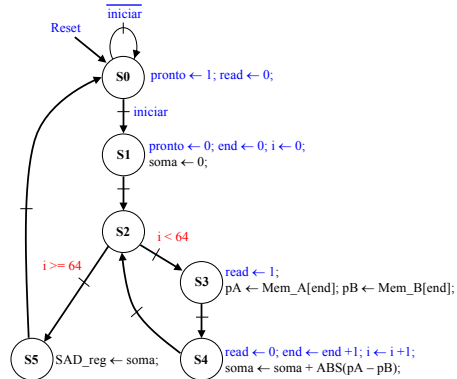
Projetando um Sistema Digital

Exemplo 3: Passo 2 (projeto do BO)



Quais variáveis são usadas para armazenar dados?

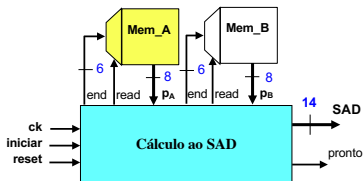
- 6: “pA”, “pB”, “soma”, “SAD_reg”, “end” e “i” (sendo “end” e “i” duas últimas, variáveis de controle).



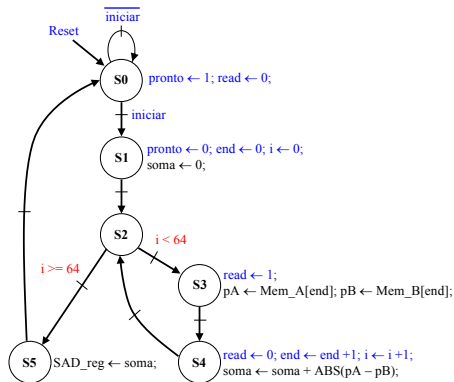
Processadores Dedicados

Projetando um Sistema Digital

Exemplo 3: Passo 2 (projeto do BO)



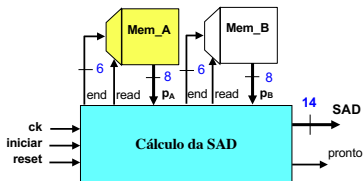
- Porém, note que para cada vez que o laço é executado, as variáveis “end” e “i” possuem o mesmo valor! Então, usaremos **somente um registrador** para ambas variáveis. Chamemo-lo de “i”.



Processadores Dedicados

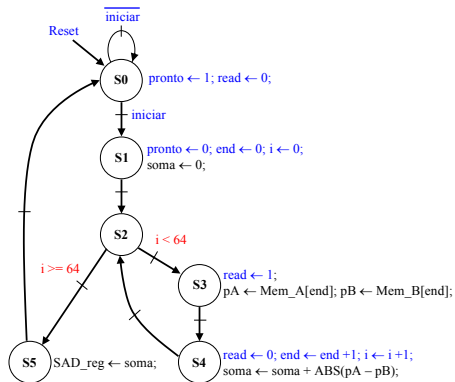
Projetando um Sistema Digital

Exemplo 3: Passo 2 (projeto do BO)



Quais operações são realizadas sobre dados ?

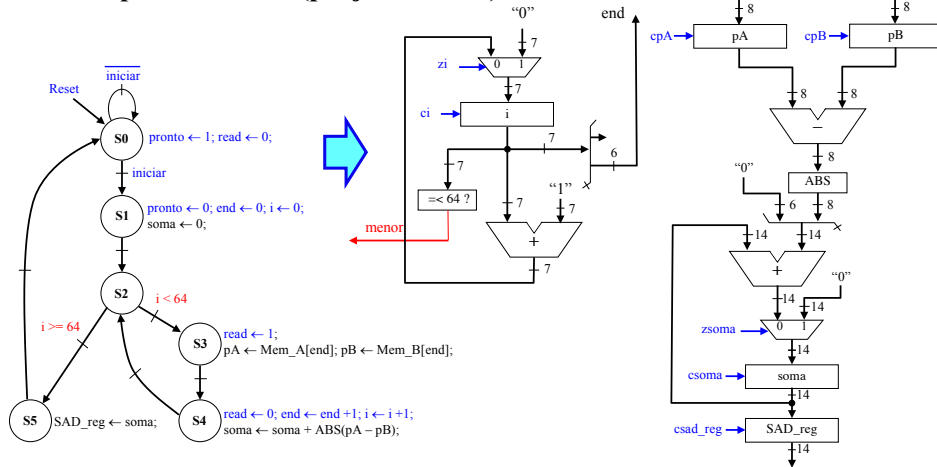
- Em “ $soma \leftarrow soma + ABS(pA - pB)$ ”: uma subtração, uma extração de módulo e uma adição
- Em “ $i \leftarrow i + 1$ ”: incremento (soma 1).



Processadores Dedicados

Projetando um Sistema Digital

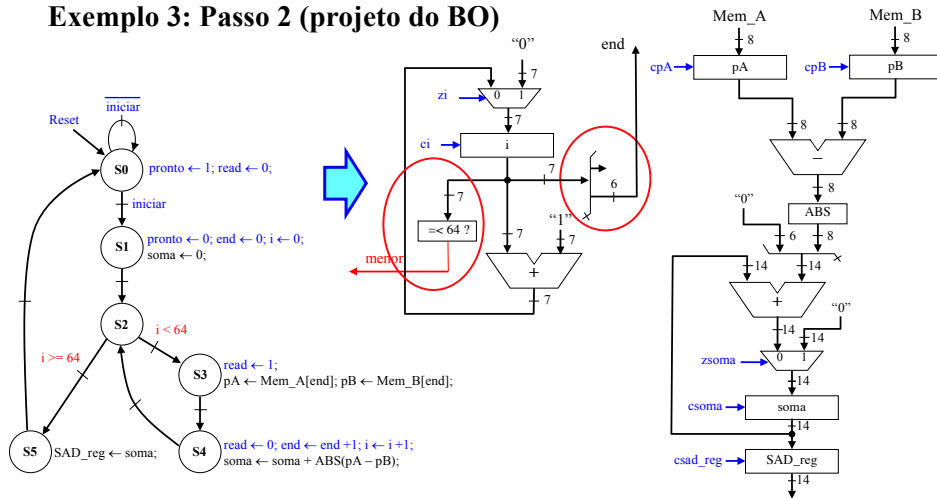
Exemplo 3: Passo 2 (projeto do BO)



Processadores Dedicados

Projetando um Sistema Digital

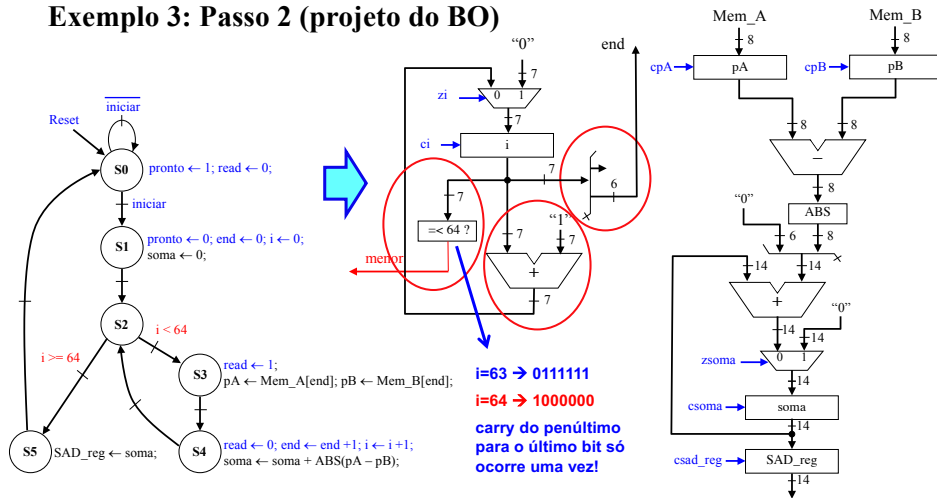
Exemplo 3: Passo 2 (projeto do BO)



Processadores Dedicados

Projetando um Sistema Digital

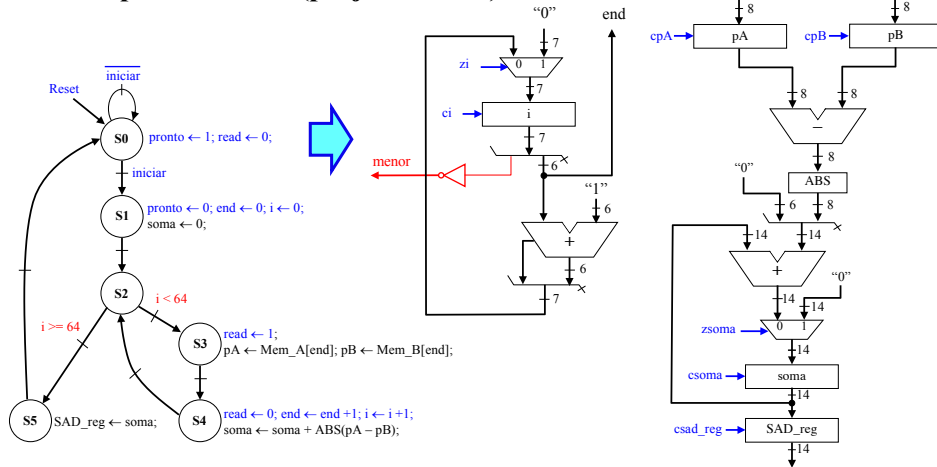
Exemplo 3: Passo 2 (projeto do BO)



Processadores Dedicados

Projetando um Sistema Digital

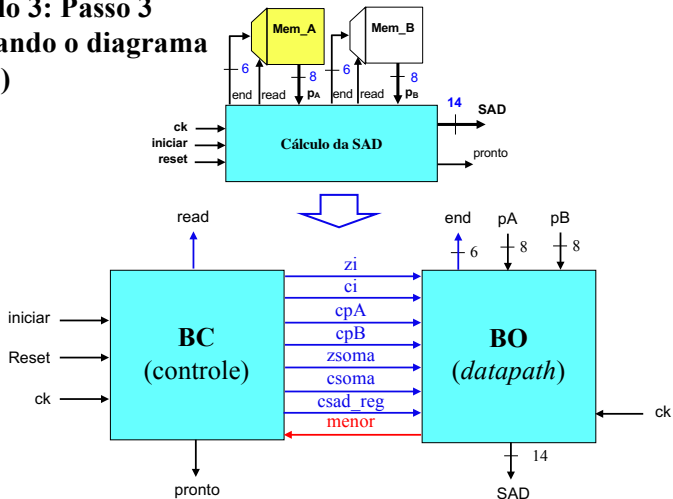
Exemplo 3: Passo 2 (projeto do BO)



Processadores Dedicados

Projetando um Sistema Digital

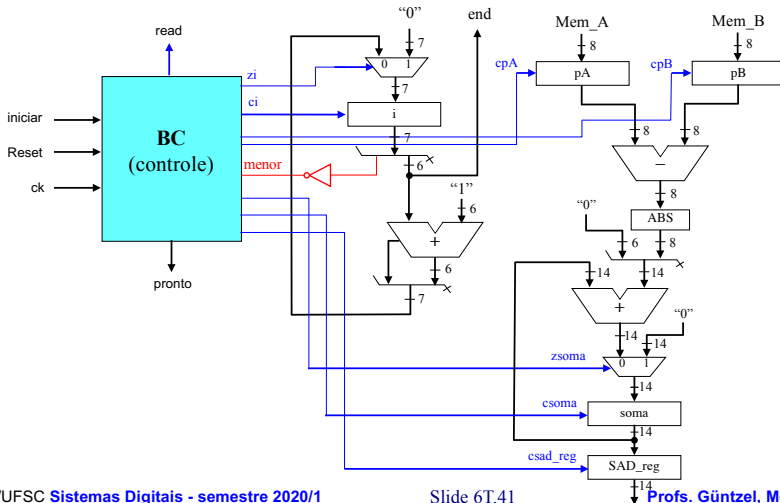
Exemplo 3: Passo 3
(Esboçando o diagrama
BO/BC)



Processadores Dedicados

Projetando um Sistema Digital

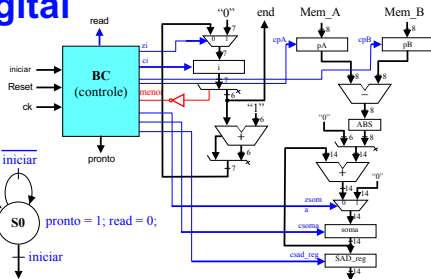
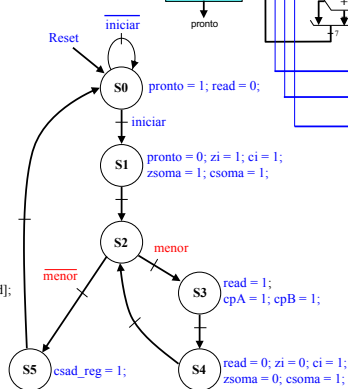
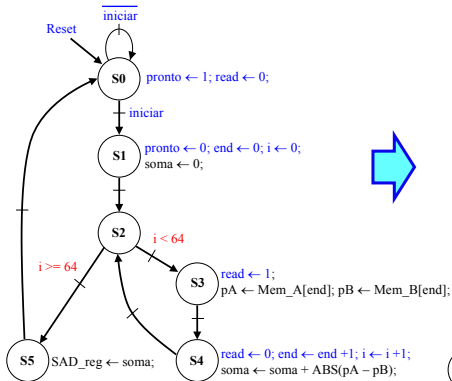
Exemplo 3: 3 (Um diagrama BO/BC mais detalhado...)



Processadores Dedicados

Projetando um Sistema Digital

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



Processadores Dedicados

Projetando um Sistema Digital

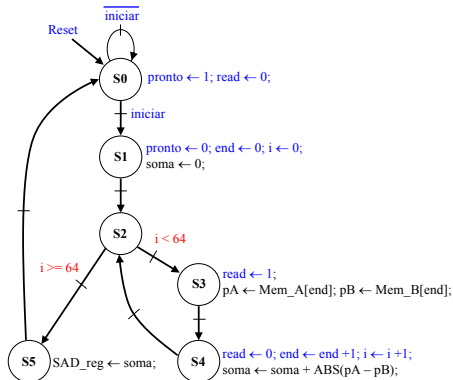
**Quantos ciclos de relógio
são necessários para
calcular um valor de
SAD?**

Resposta:

Para aprontar um cálculo de SAD é
necessário executar a seguinte
sequência de estados:

S1 , 64x(S2, S3, S4) , S2 , S5

o que requer **195 ciclos**



Processadores Dedicados

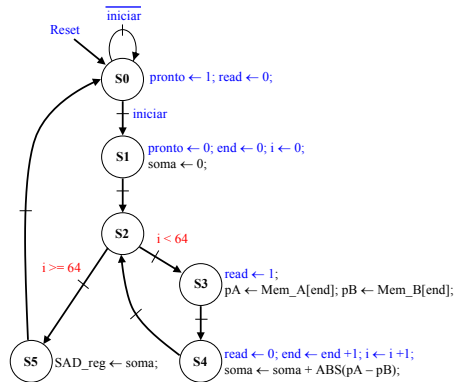
Projetando um Sistema Digital

Qual o tempo de execução de um cálculo de SAD?

Resposta:

$$TE_{SAD} = 195 \text{ ciclos} \cdot T$$

Logo, é preciso estimar o período do relógio!



Processadores Dedicados

Projetando um Sistema Digital

Estimando o período do relógio

Suponha as seguintes características temporais para os componentes do SD.

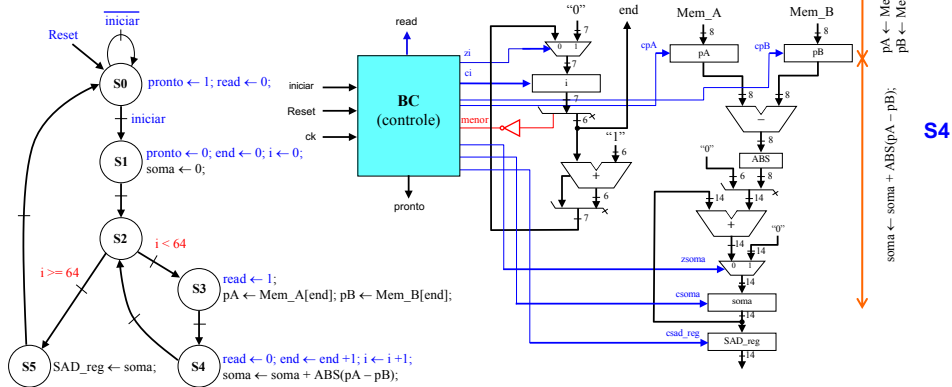
Componente	Característica	Símbolo	Valor
Memórias Mem_A e Mem_B	tempo para leitura de uma linha	tleit	100 ps
Registradores	tempo de setup	tsu	5 ps
Registradores	tempo de carga	tco	5 ps
Registradores	tempo de hold	th	Desprezível (0 ps)
Atraso	Mux2:1	tdmux	2 ps
Somador, subtrator e “ABS”	atraso	tds	10. n ps
Registrador de estados (do controle)	tempos de setup, carga e hold	-	Desprezível (0 ps)*
Lógica de saída do controle	atraso	-	Desprezível (0 ps)*

n = número de bits;

* Isto significa que os sinais de controle não possuem atraso.

Processadores Dedicados

Projetando um Sistema Digital Estimando o período do relógio



Claramente, os estados S3 e S4 são os que possuem operações mais lentas!

Processadores Dedicados

Projetando um Sistema Digital

Estimando o período do relógio

S3 $pA \leftarrow \text{Mem_A}[\text{end}]; pB \leftarrow \text{Mem_B}[\text{end}];$

$$\begin{aligned} \text{td}(\text{S3}) &= \text{tleit}(\text{Mem_A}) + \text{tsu}(pA) = \\ &= \text{tleit}(\text{Mem_B}) + \text{tsu}(pB) = \\ &= 100 \text{ ps} + 5 \text{ ps} = \mathbf{105 \text{ ps}} \end{aligned}$$

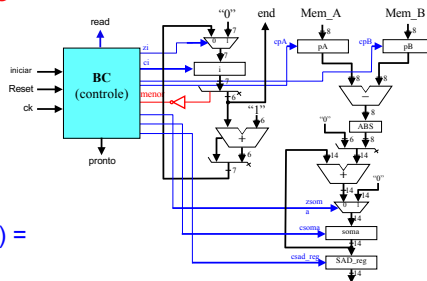
S4: op1 $i \leftarrow i + 1;$ (que inclui $\text{end} \leftarrow \text{end} + 1;$)

$$\begin{aligned} \text{td}(\text{S4:op1}) &= \text{tco}(i) + 6.\text{tds} + \text{tdmux} + \text{tsu}(i) = \\ &= 5 + 6.10 + 2 + 5 = \mathbf{72 \text{ ps}} \end{aligned}$$

S4: op2 $\text{soma} \leftarrow \text{soma} + \text{ABS}(pA - pB);$

$$\begin{aligned} \text{td}(\text{S4:op2}) &= \text{tco}(pA) + 8.\text{tds} + 8.\text{tds} + 14.\text{tds} + \text{tdmux} + \text{tsu}(\text{soma}) = \\ &= \text{tco}(pB) + 8.\text{tds} + 8.\text{tds} + 14.\text{tds} + \text{tdmux} + \text{tsu}(\text{soma}) = \\ &= 5 + 8.10 + 8.10 + 14.10 + 2 + 5 = \mathbf{312 \text{ ps}} \end{aligned}$$

$$\text{td}(\text{S4}) = \max\{ \text{td}(\text{S4:op1}), \text{td}(\text{S4:op2}) \} = \max\{ 72, 312 \} = \mathbf{312 \text{ ps}}$$



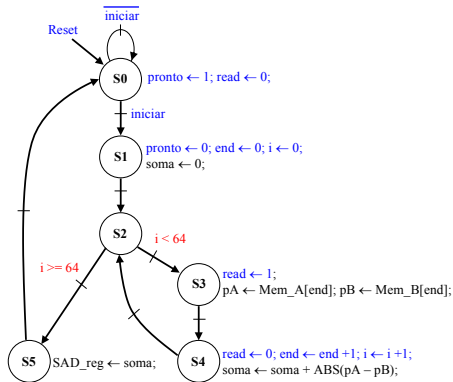
Processadores Dedicados

Projetando um Sistema Digital

Qual o tempo de execução de um cálculo de SAD?

Resposta:

$$\begin{aligned} TE_{SAD} &= 195 \text{ ciclos} \cdot T = \\ &= 195 \text{ ciclos} \cdot 312 \text{ ps/ciclo} = \\ &= 60840 \text{ ps} = 60,84 \text{ ns} \end{aligned}$$



Processadores Dedicados

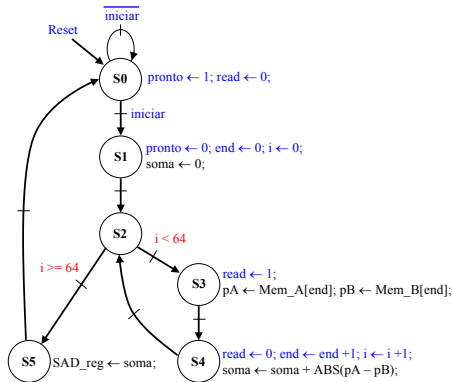
Projetando um Sistema Digital

Qual o tempo de execução de um cálculo de SAD?

Resposta:

$$\begin{aligned} TE_{SAD} &= 195 \text{ ciclos} \cdot T = \\ &= 195 \text{ ciclos} \cdot 312 \text{ ps/ciclo} = \\ &= 60840 \text{ ps} = 60,84 \text{ ns} \end{aligned}$$

Como pode-se reduzir o tempo de execução?

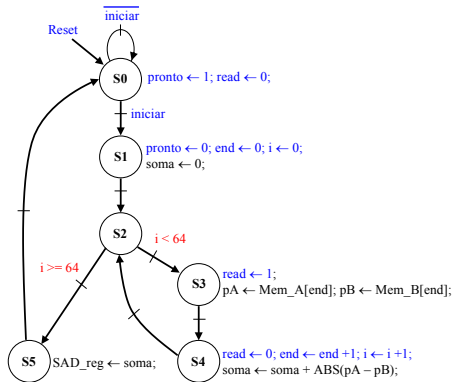


Processadores Dedicados

Projetando um Sistema Digital

Como pode-se reduzir o tempo de execução?

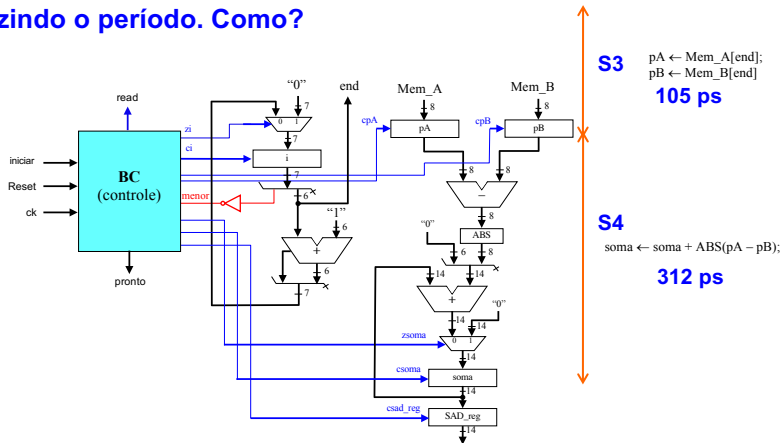
1. Reduzindo o período. Como?
2. Reduzindo o número de ciclos para a execução. Como?



Processadores Dedicados

Projetando um Sistema Digital

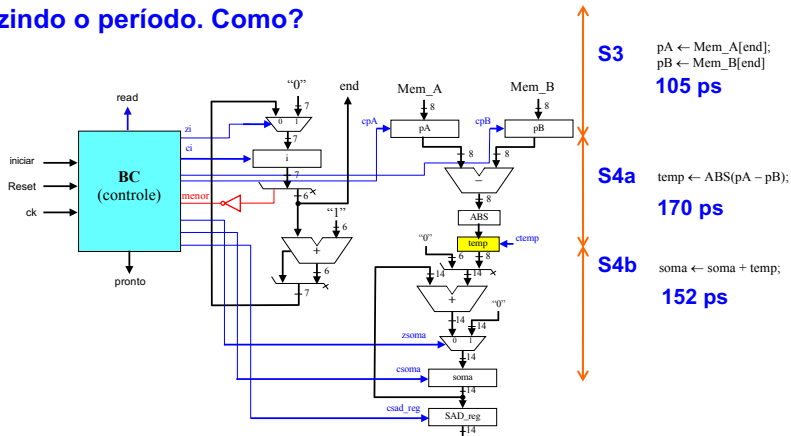
1. Reduzindo o período. Como?



Processadores Dedicados

Projetando um Sistema Digital

1. Reduzindo o período. Como?



Quebrando as operações de S4 em dois ciclos de relógio (i.e., transformando S4 em dois estados, S4a e S4b)

Processadores Dedicados

Projetando um Sistema Digital

1. Reduzindo o período. Como?

Novo tempo de execução:

Sequência de estados para aprontar um cálculo de SAD com este novo B.O.:

S1 , 64x(S2, S3, S4a, S4b) , S2 , S5

o que requer **259 ciclos**

$TE_{SAD} = 259 \text{ ciclos} \cdot T =$

$= 259 \text{ ciclos} \cdot 170 \text{ ps/ciclo} =$

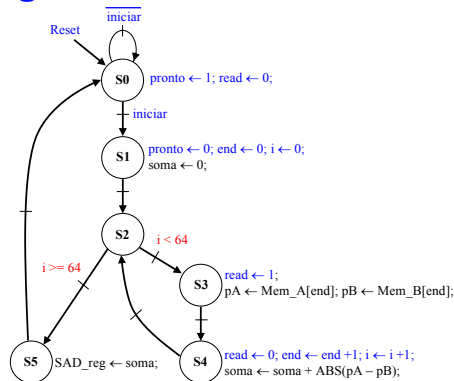
$= 44030 \text{ ps} = 44,03 \text{ ns}$ (antes era 60,84 ns)

Processadores Dedicados

Projetando um Sistema Digital

2. Reduzindo o número de ciclos para a execução. Como?

Seria possível reduzir o número de vezes que o laço é executado?



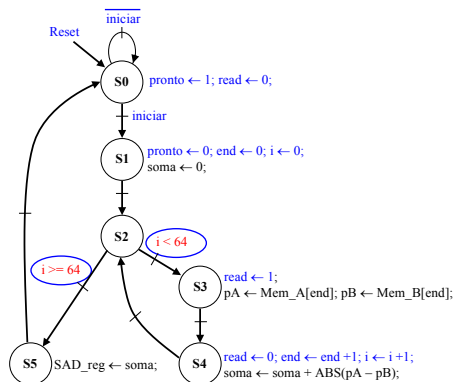
Processadores Dedicados

Projetando um Sistema Digital

2. Reduzindo o número de ciclos para a execução. Como?

Seria possível reduzir o número de vezes que o laço é executado?

Sim! Aumentando o paralelismo do B.O.!



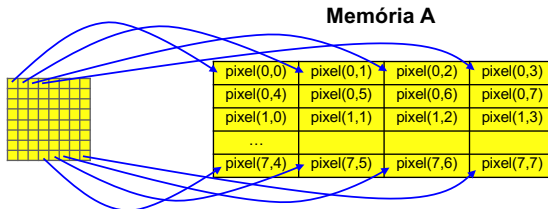
Processadores Dedicados

Projetando um Sistema Digital

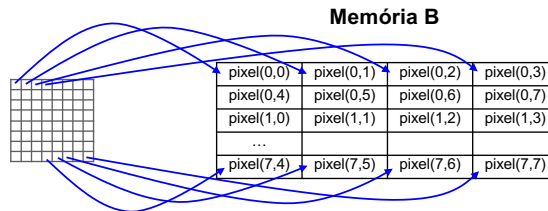
Nova Organização da “Memória A” e da “Memória B” para Explorar Paralelismo no B.O.

Matriz A:

Porção de 8x8 pixels $\in Q_{ref}$



Matriz B: $b_j \in Q_i$



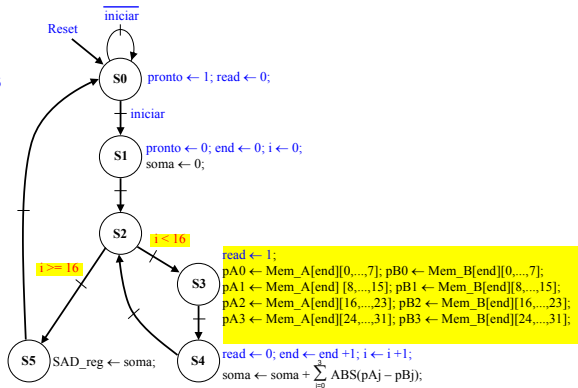
Processadores Dedicados

Projetando um Sistema Digital

2. Reduzindo o número de ciclos para a execução. Como?

Resp.: Explorando Paralelismo no BO

- Processando 4 pares de pixels por ciclo, será necessário executar **16 vezes o laço** para processar todos os 64 pares!
- A memória será acessada somente 16 vezes (cada linha da memória contém 4 pixels);



Processadores Dedicados

Projetando um Sistema Digital

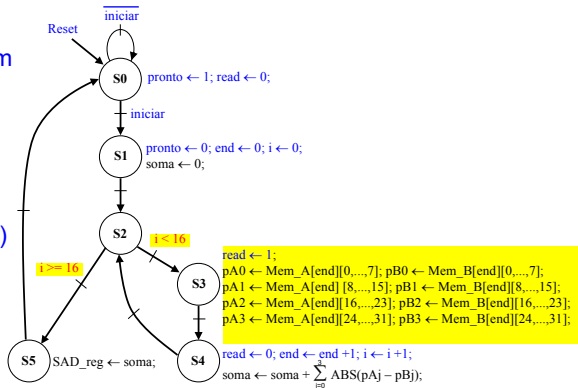
2. Reduzindo o número de ciclos para a execução. Como?

Resp.: Explorando Paralelismo no BO

Sequência de estados para
aprontar um cálculo de SAD com
este B.O. Que explora
paralelismo:

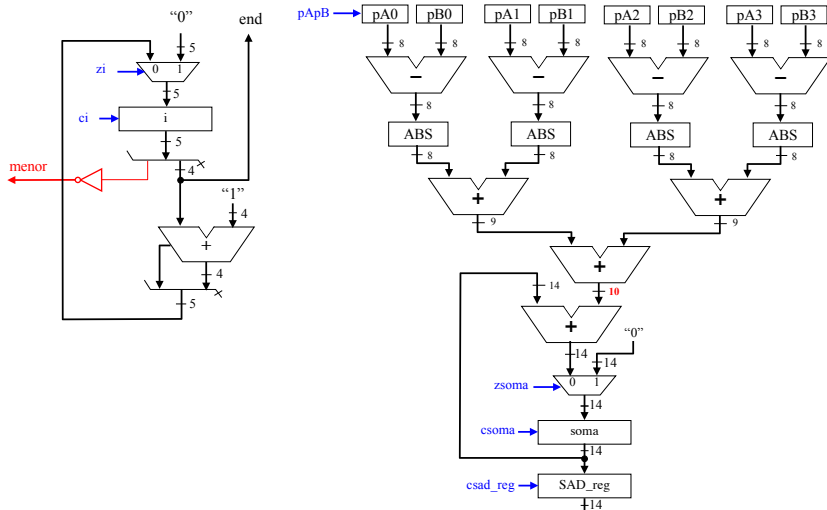
S1 , 16x(S2, S3, S4) , S2 , S5

o que requer **51ciclos** (na
versão original eram **259 ciclos**)



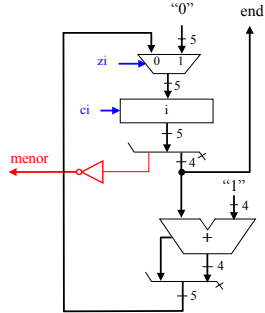
Processadores Dedicados

Projetando um Sistema Digital

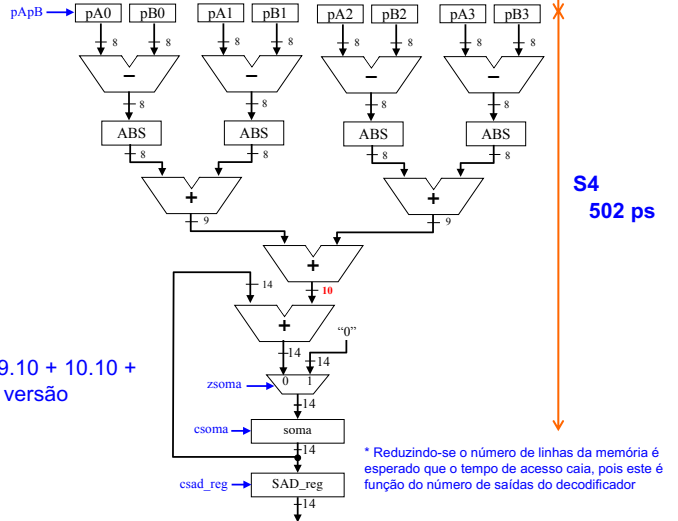


Processadores Dedicados

Projetando um Sistema Digital



td(S4) = 5 + 8.10 + 8.10 + 9.10 + 10.10 + 14.10 + 2 + 5 = 502 ps (na versão original era **312 ps**)



Processadores Dedicados

Projetando um Sistema Digital

2. Reduzindo o número de ciclos para a execução. Como?

Resp.: Explorando Paralelismo no BO

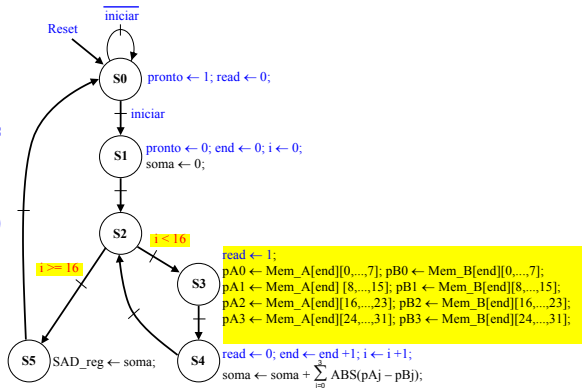
Tempo de execução:

$$TE_{SAD} = 51 \text{ ciclos} \cdot T =$$

$$= 51 \text{ ciclos} \cdot 502 \text{ ps/ciclo} =$$

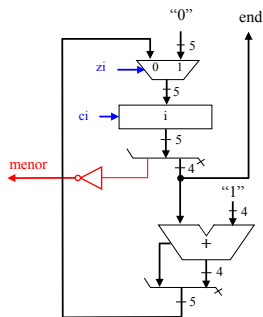
$$= 25602 \text{ ps} = 25,6 \text{ ns}$$

(na versão original era 60,84 ns)



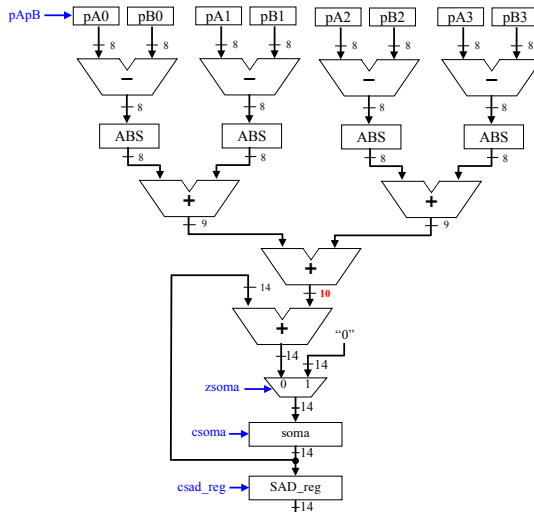
Processadores Dedicados

Projetando um Sistema Digital



Quais as consequências:

- No custo do B.O.?
- No custo do B.C.?



Processadores Dedicados

Projetando um Sistema Digital

2. Reduzindo o número de ciclos para a execução. Como?

$pA \leftarrow \text{Mem_A}[\text{end}]; pB \leftarrow \text{Mem_B}[\text{end}];$

Poderia ser feita em S2 (ao invés de S3)?

Se sim, a execução iria requerer a seguinte sequência de estados:

S1 , 64x(S2, S3) , S2 , S4

Ou seja, **131ciclos**

