

Organização de Computadores

Direct Memory Access - DMA

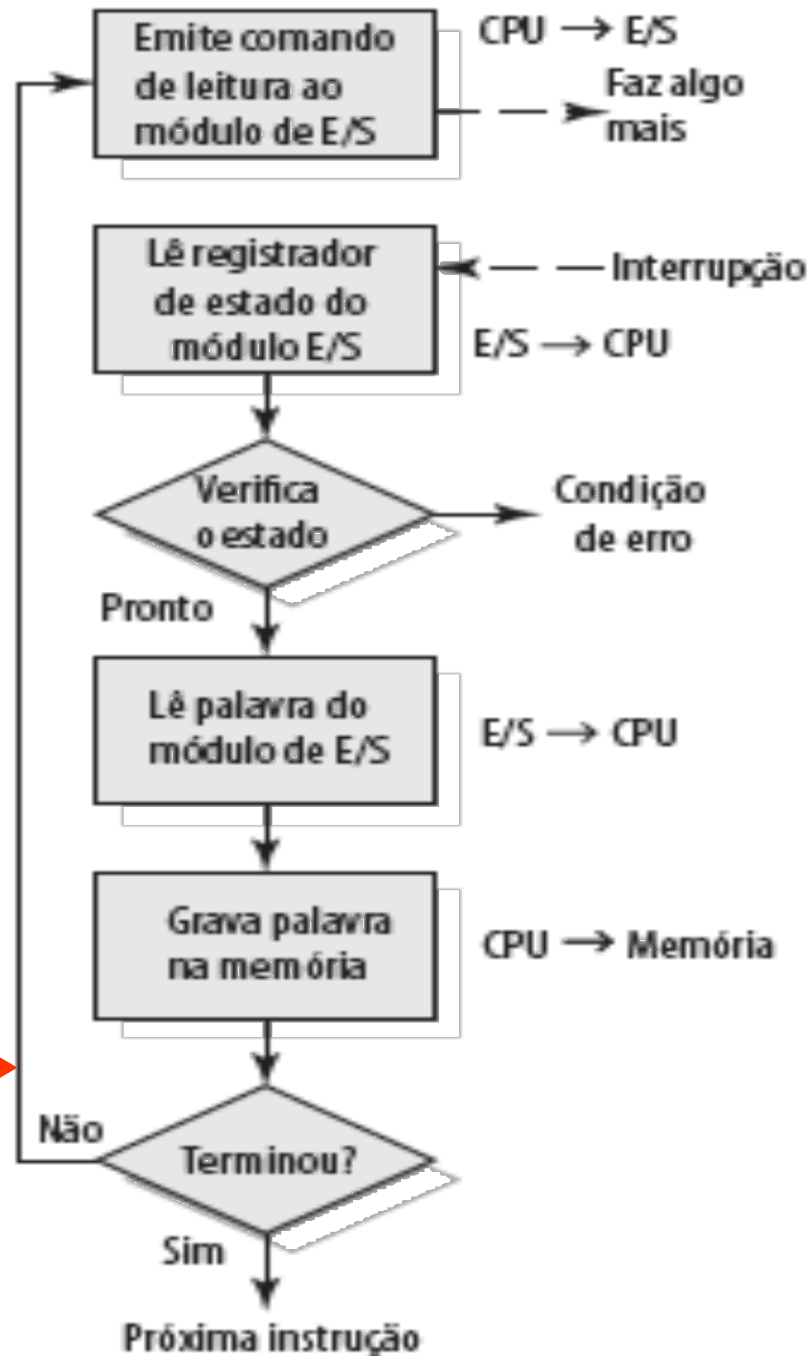
Prof. Jose Paulo . de Oliveira
Eng. da Computação, UPE

jpgo@ecomp.poli.br

Resumo

- Motivação
- Visão geral
- Módulo DMA
- Modos de Operação
- Configurações (de HW)
- Sistemas de alto desempenho

E/S Comandada por Interrupção



Uma interrupção por caractere transferido

Limitações de E/S por ISR

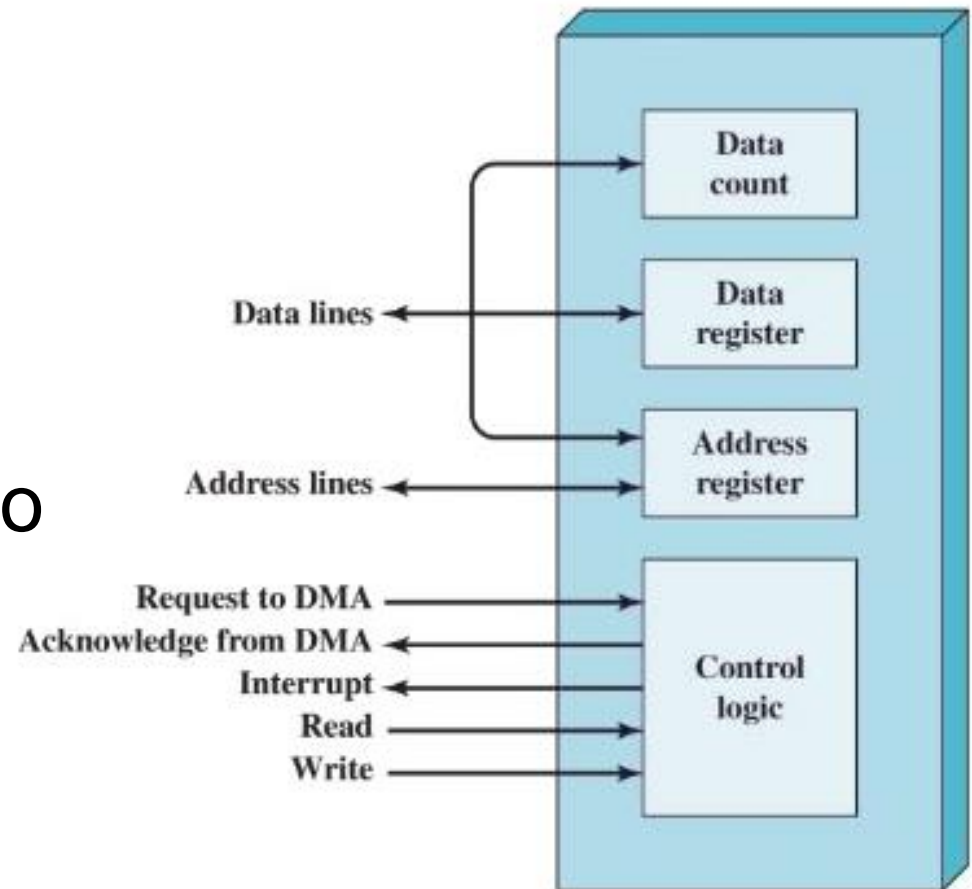
- E/S executada por interrupção e E/S programada exigem intervenção ativa da CPU
 - Taxa de transferência limitada
 - Pela velocidade com que a CPU pode atender a um dispositivo
 - Limitações no uso da CPU
 - Perda de dados

Limitações de E/S por ISR

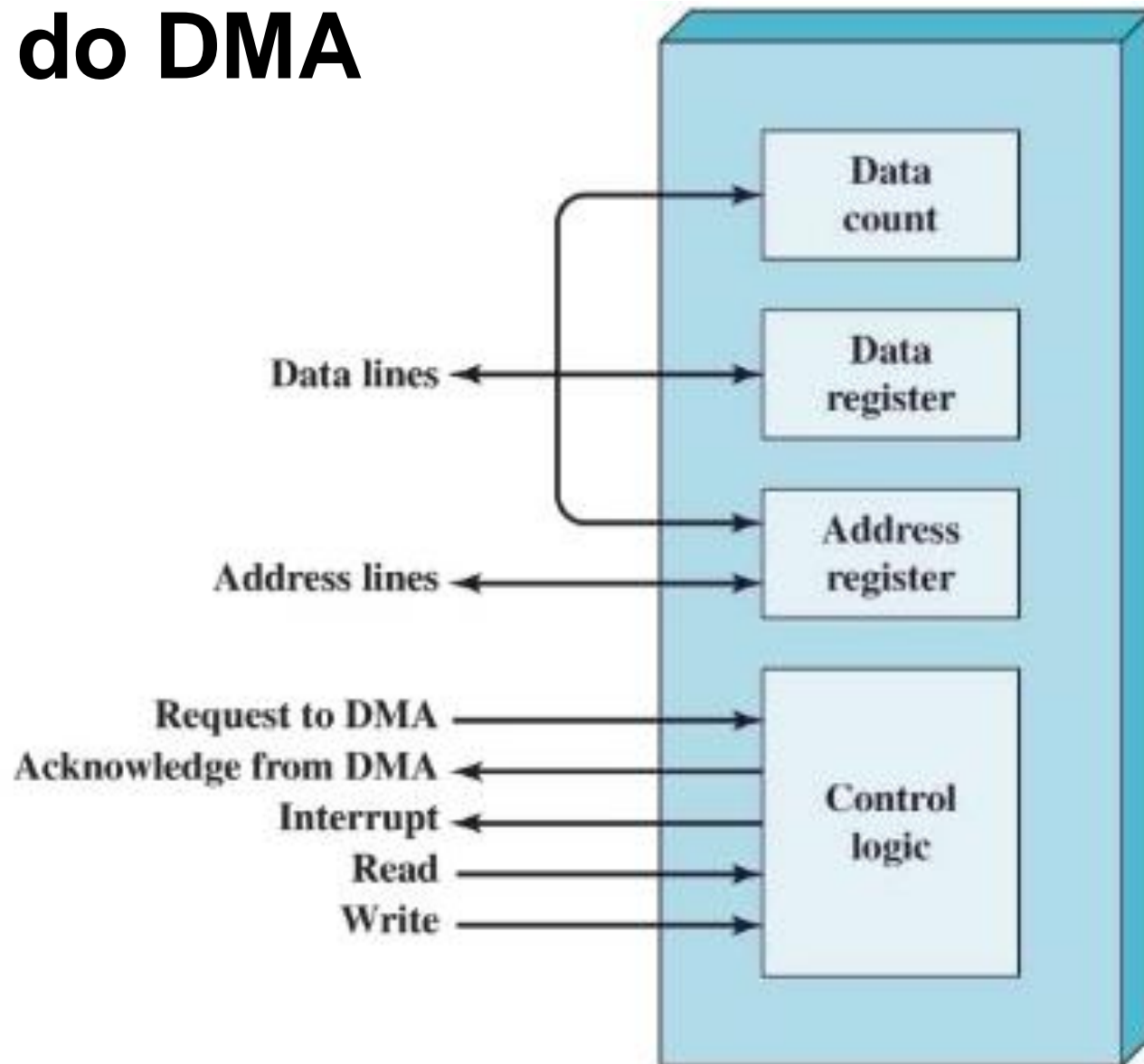
- E/S executada por interrupção e E/S programada exigem intervenção ativa da CPU
 - Taxa de transferência limitada
 - Pela velocidade com que a CPU pode atender a um dispositivo
 - Limitações no uso da CPU
 - Perda de dados
- Solução \Rightarrow **DMA**

Função do DMA

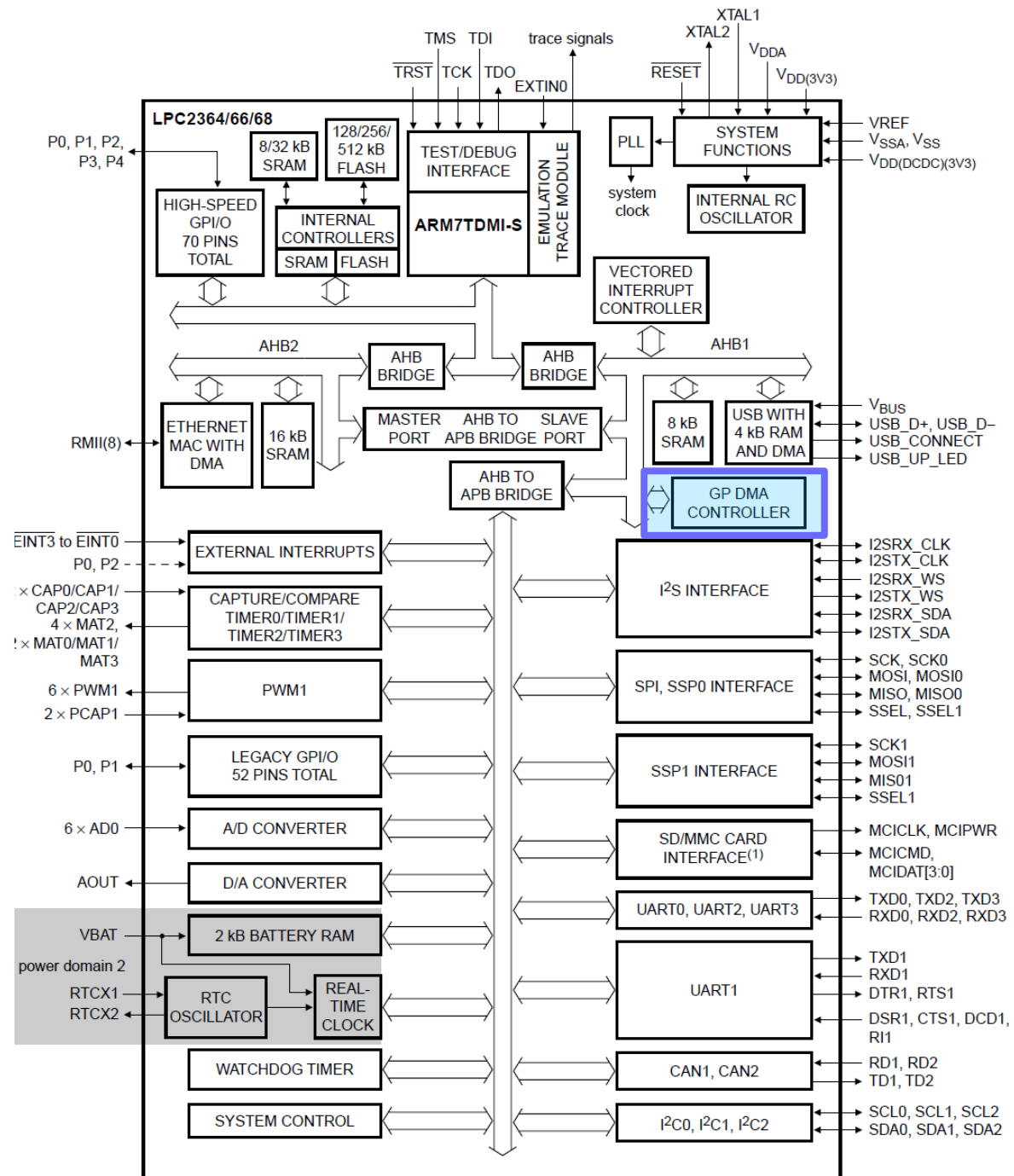
- Módulo adicional (hardware) no barramento
- O controlador de **DMA** assume o lugar da CPU para o dispositivo de E/S



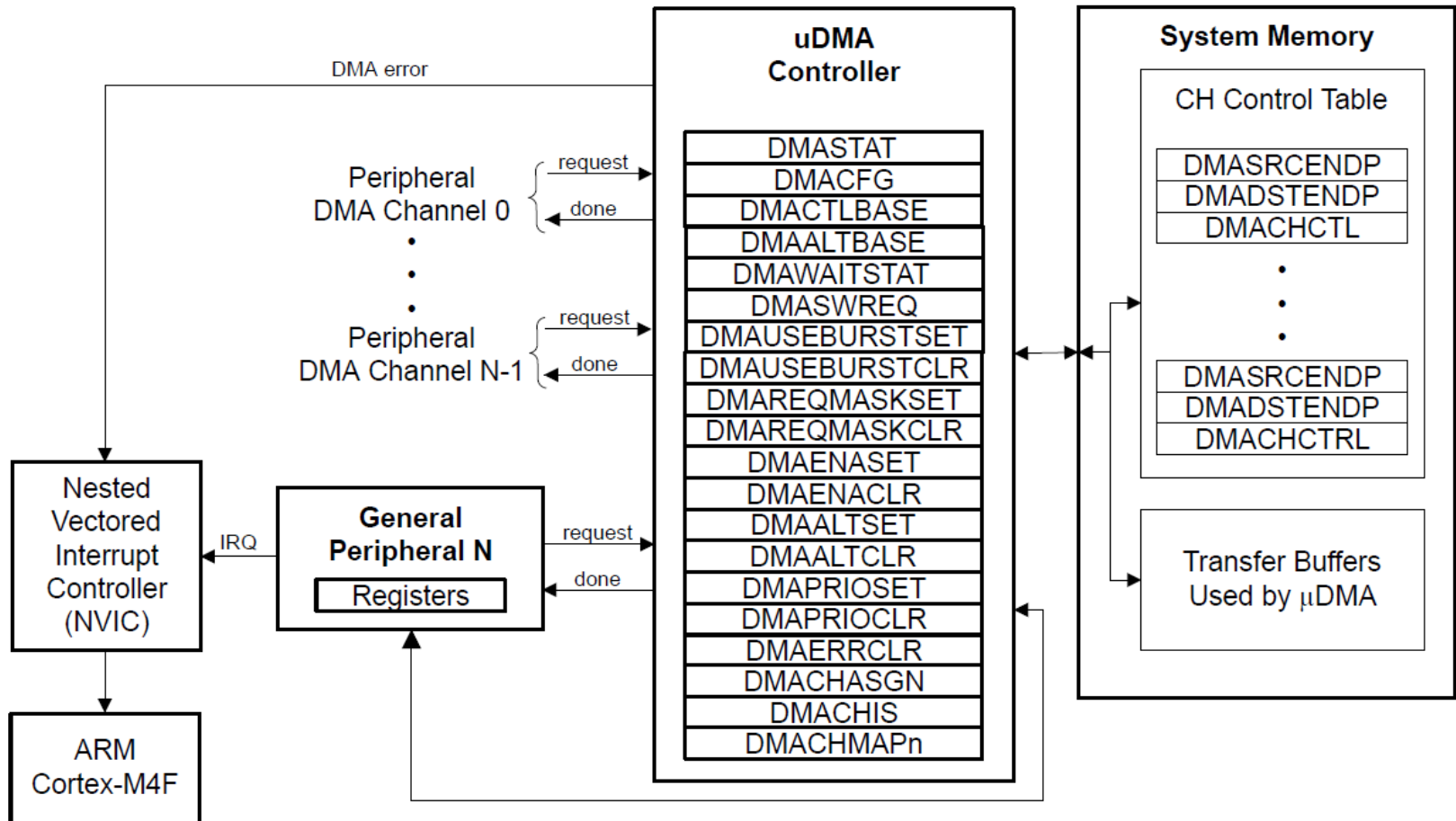
Função do DMA



Ex.: ARM



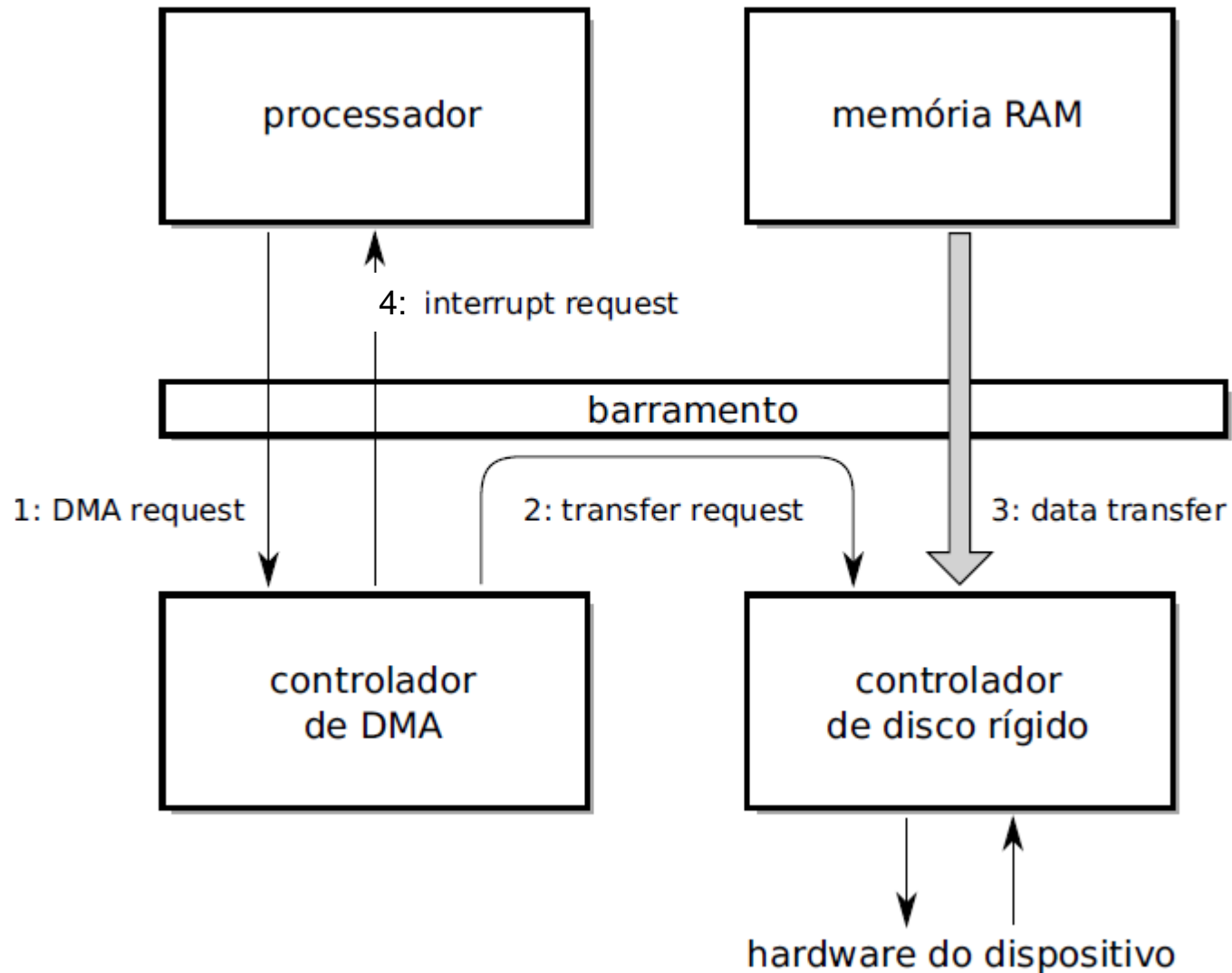
Ex.: ARM



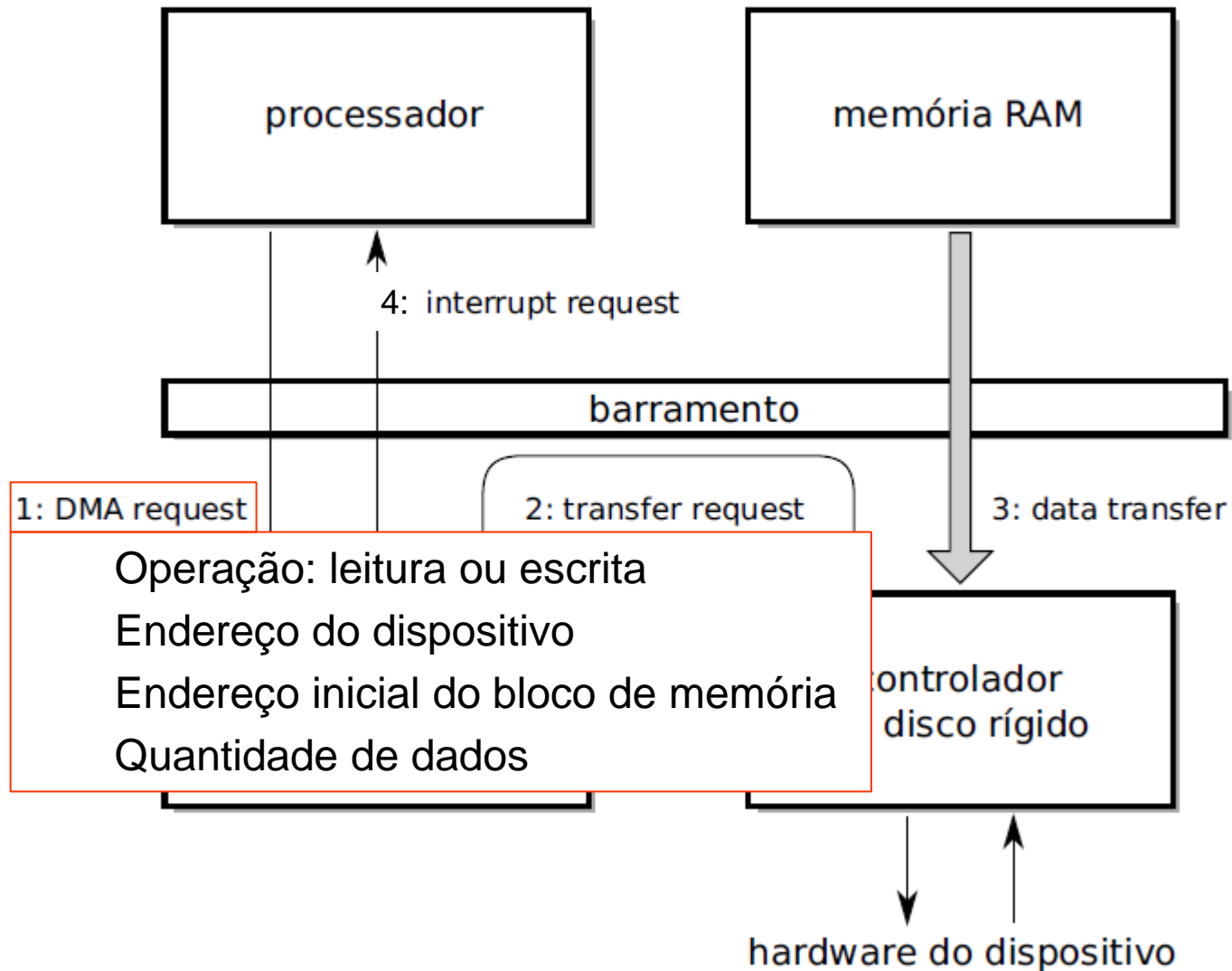
Exemplo: salvando arquivos



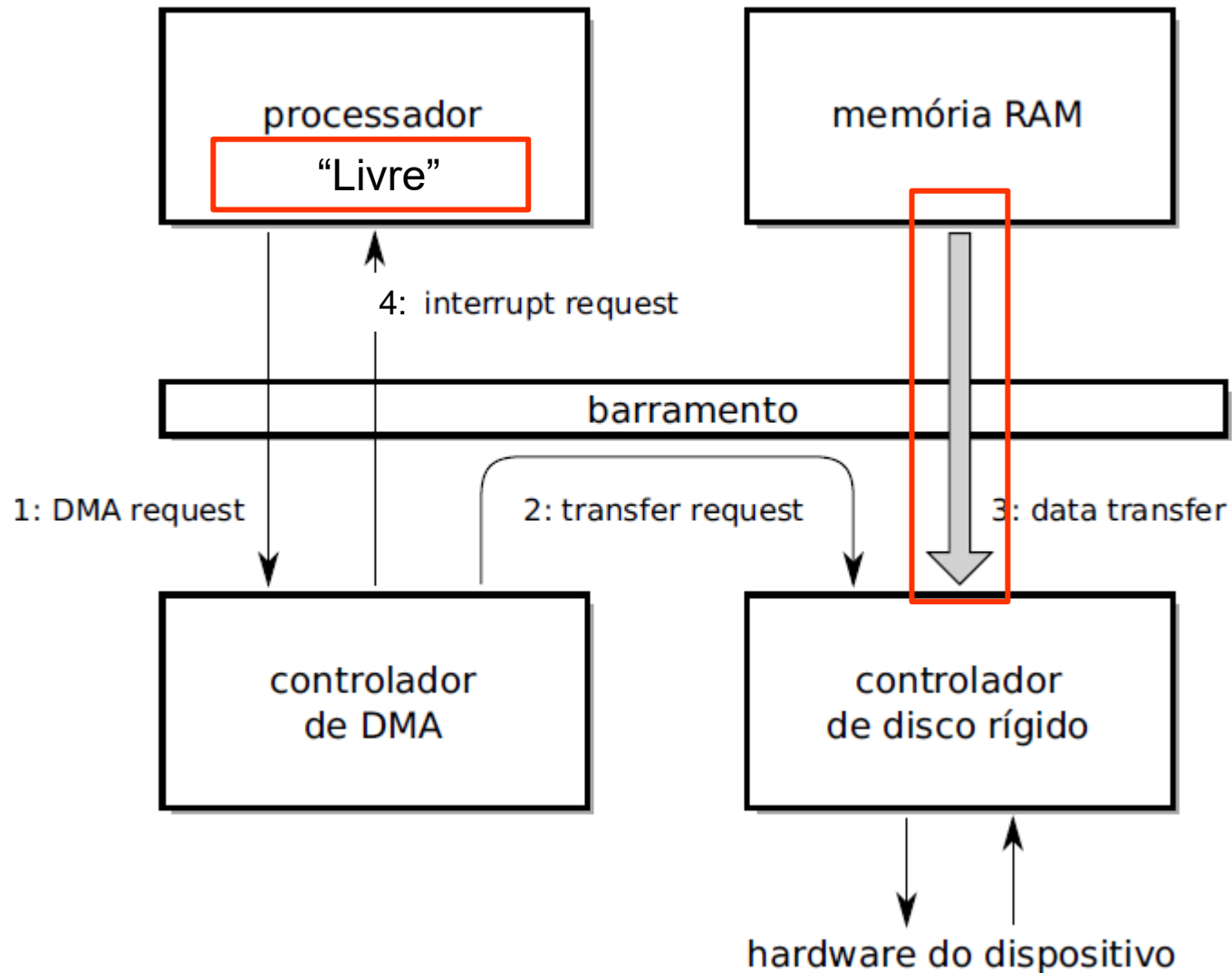
Operação da DMA



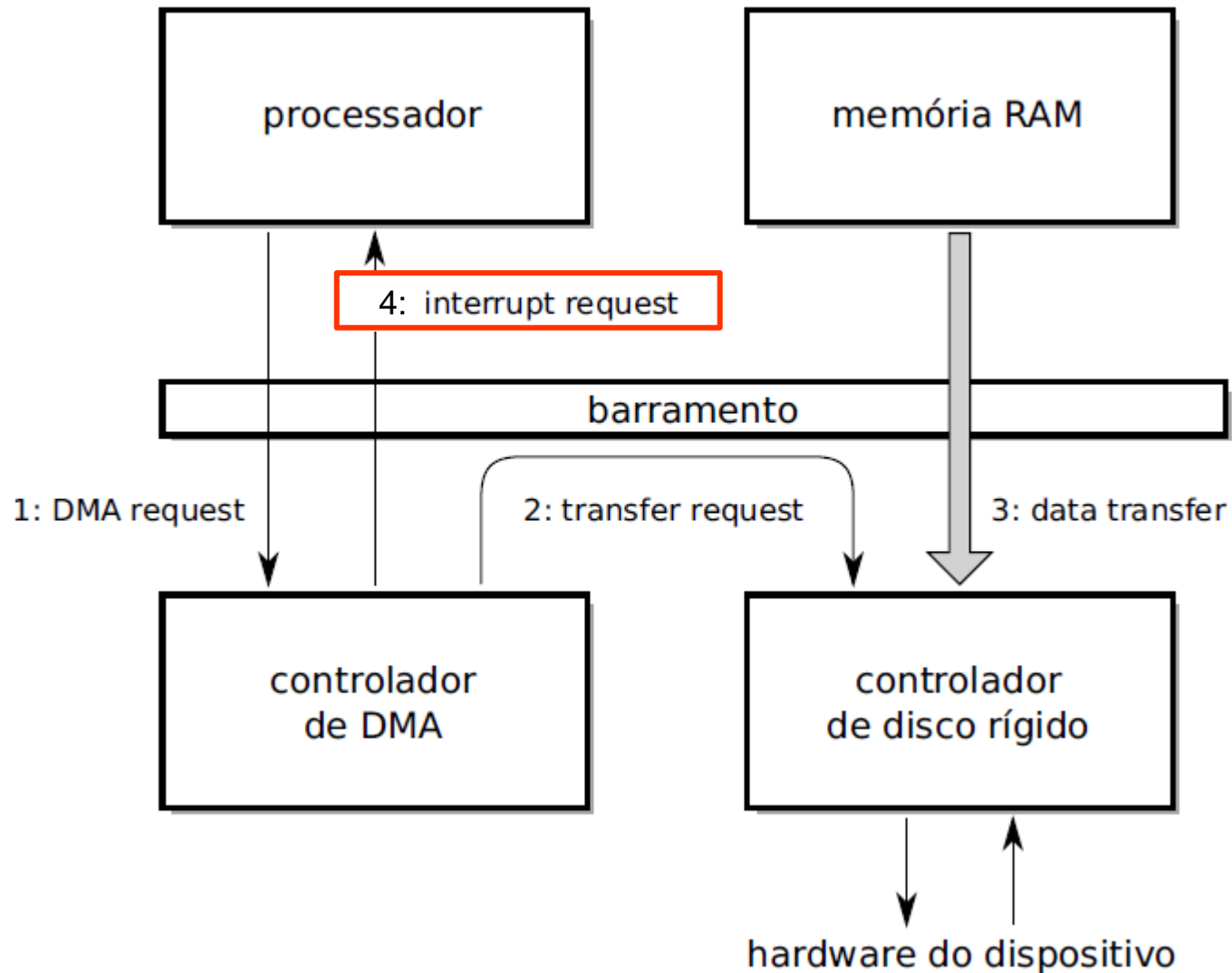
Operação da DMA



Operação da DMA



Operação da DMA

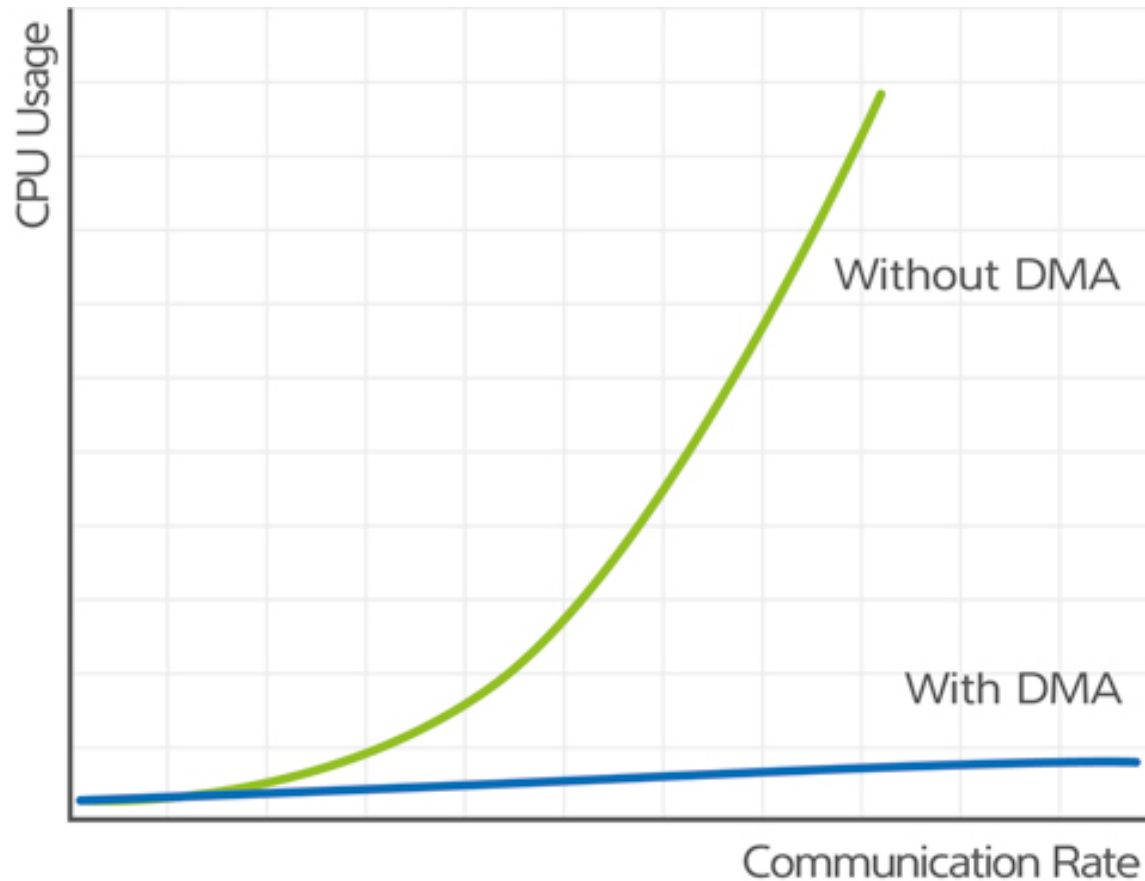


Operação da DMA

- CPU envia comando ao controlador de DMA:
 - Operação: leitura ou escrita
 - Endereço do dispositivo
 - Endereço inicial do bloco de memória para os dados
 - Quantidade de dados a ser transferida
- A CPU passa a fazer outra coisa
- O controlador de DMA cuida de toda a transferência e envia uma interrupção quando termina a tarefa

Desempenho

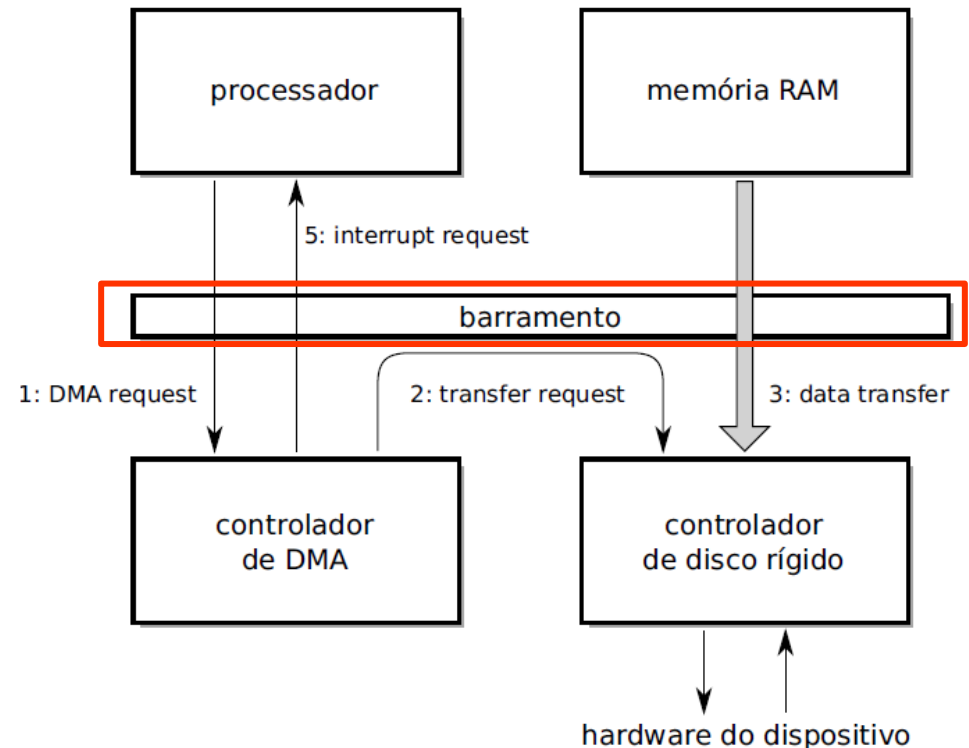
Ex.: AVR XMEGA 4-channel DMA controller



Modos de Operação

Modos de Operação

1. Transferência em bloco (*burst mode*):
2. Modo transparente
3. Roubo de ciclo



Modos de Operação

1. Transferência em bloco (*burst mode*):
 - Bloco completo transferido
 - A CPU fica apenas *inativa*

Modos de Operação

1. Transferência em bloco (*burst mode*):

- Bloco completo transferido
- A CPU fica apenas *inativa*

2. Modo transparente

- Similar ao modo em bloco
- A CPU **só** não pode acessar o barramento
 - Não fica completamente inativa
- Um HW determina quando a CPU não “precisa” acessar o barramento

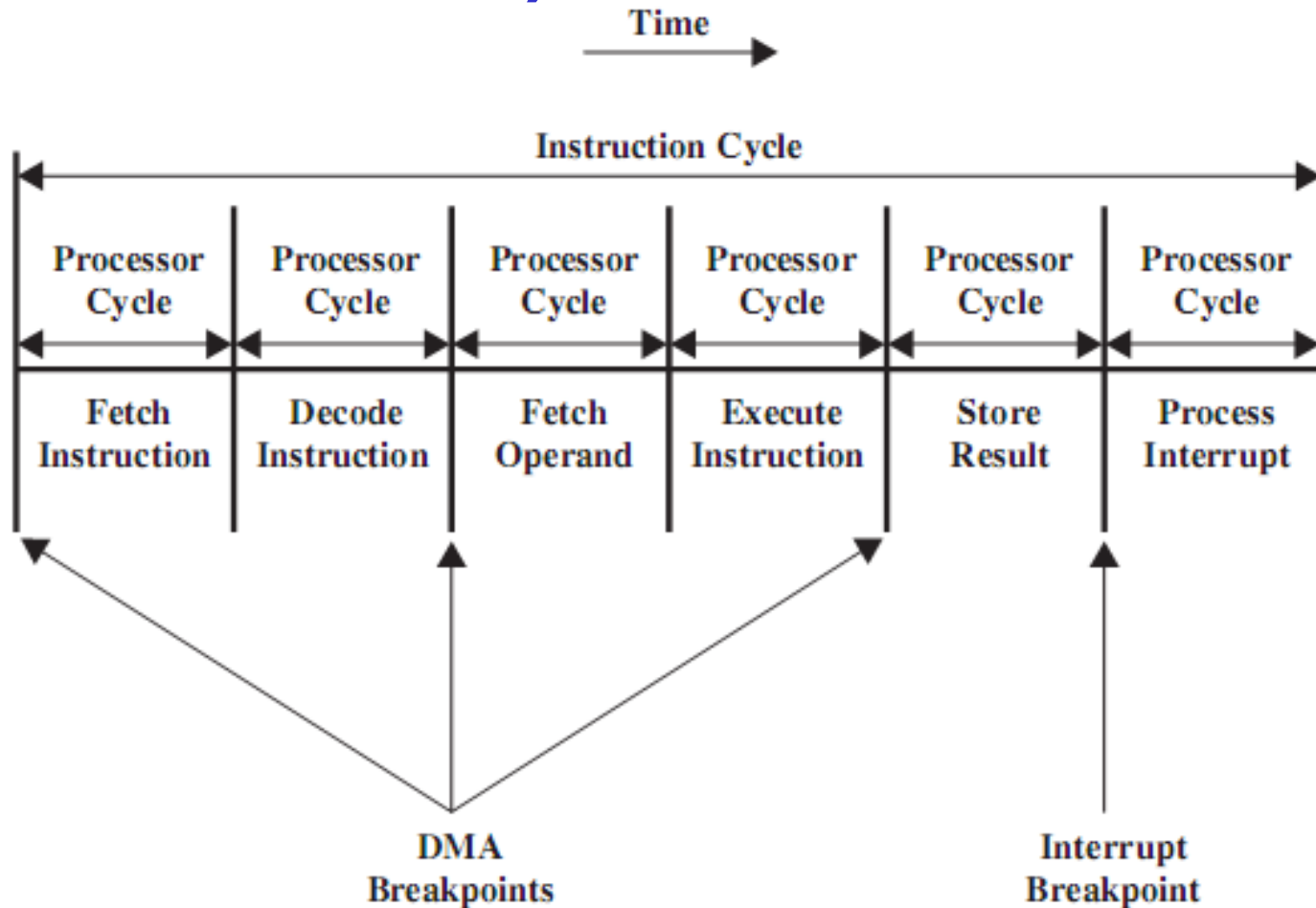
Modos de Operação

1. Transferência em bloco (*burst mode*):
 - Bloco completo transferido
 - A CPU fica inativa
2. Modo transparente
 - Similar ao modo em bloco
 - A CPU só não pode acessar o barramento
 - Não fica completamente inativa
 - Um HW determina quando a CPU não “precisa” acessar o barramento
3. Roubo de ciclo

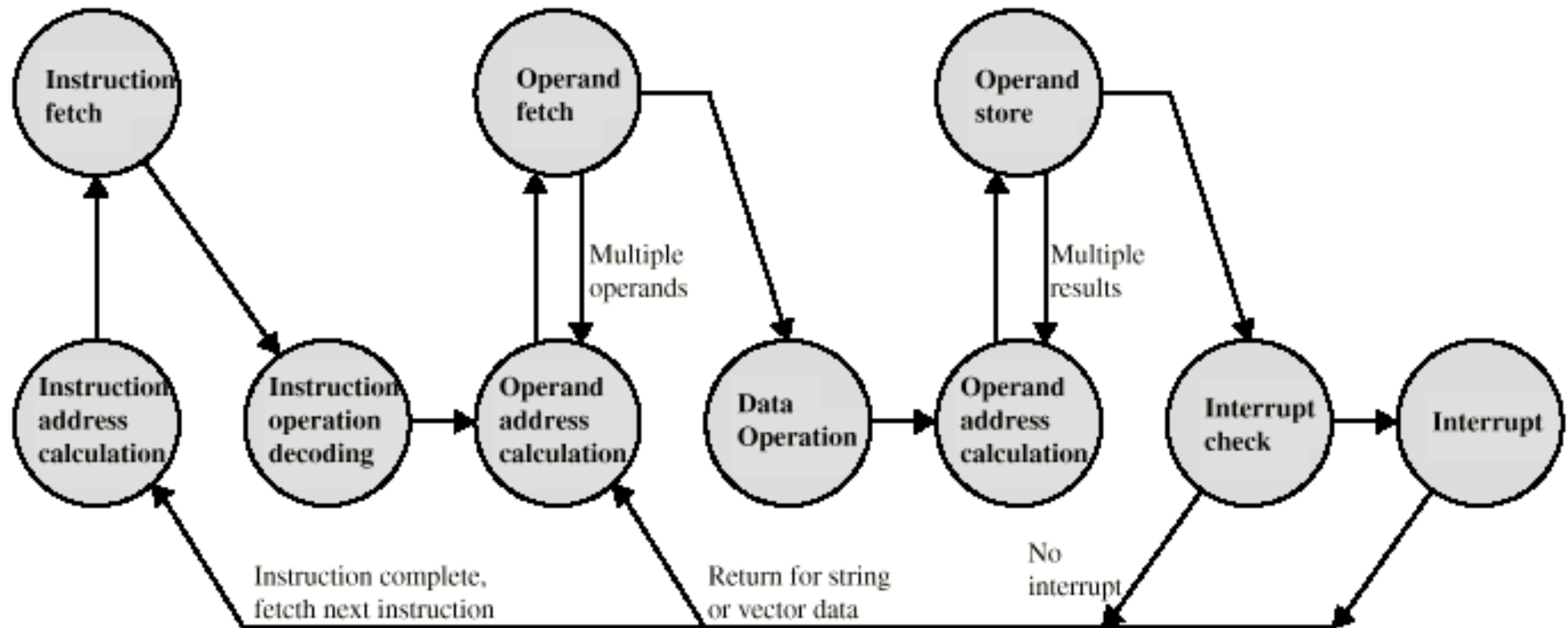
Roubo de Ciclo

- O controlador de DMA assume o controle do barramento por um ciclo (de processador)
 - CPU suspensa
- Transferência de uma **palavra**
 - A CPU não acessa o barramento
- A cada transferência de uma nova palavra, o acesso ao barramento deve ser concedido ao DMA

Transferência por DMA (Roubo de Ciclo)



Ciclo de Instrução



Transferência por DMA

(Roubo de Ciclo)

- A CPU é suspensa apenas antes de acessar o barramento:
 - i.e., antes de uma busca de operandos, de dados, ou de uma escrita
 - nos outros instantes a CPU pode operar enquanto ocorre DMA

Transferência por DMA

(Roubo de Ciclo)

- A CPU é suspensa apenas antes de acessar o barramento:
 - i.e., antes de uma busca de operandos, de dados, ou de uma escrita
 - nos outros instantes a CPU pode operar enquanto ocorre DMA
- Deixa a CPU “mais lenta”, mas, não tanto quanto quando a própria CPU faz a transferência

Transferência por DMA

(Roubo de Ciclo)

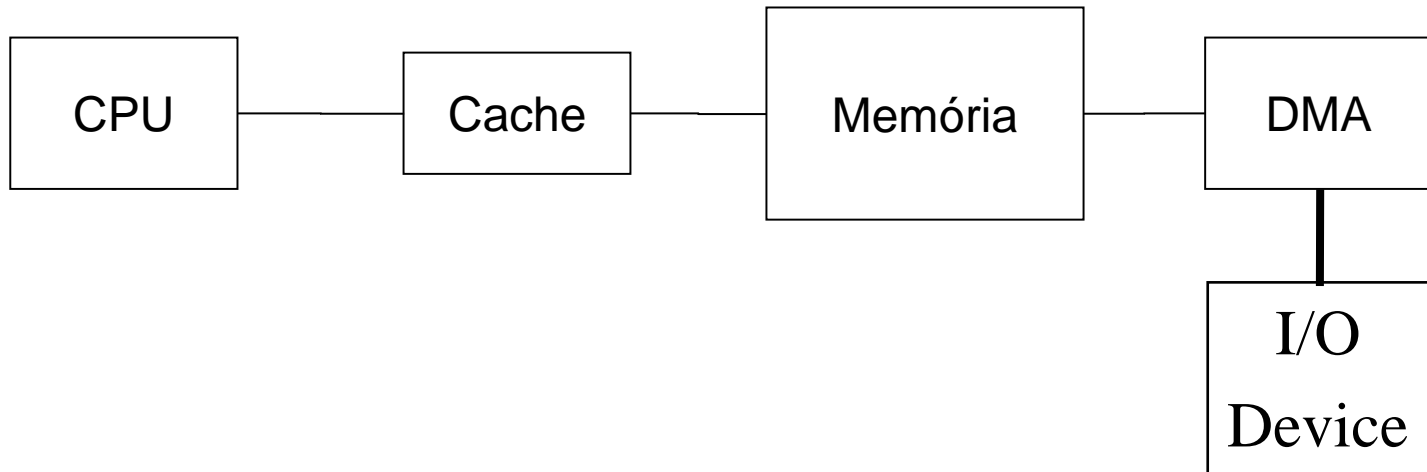
- A CPU é suspensa apenas antes de acessar o barramento:
 - i.e., antes de uma busca de operandos, de dados, ou de uma escrita
 - nos outros instantes a CPU pode operar enquanto ocorre DMA
- Deixa a CPU “mais lenta”, mas, não tanto quanto quando a própria CPU faz a transferência
- Módulo DMA pode operar tão rápido quanto a CPU

Há um “porém” quando se usa DMA

- Qual o efeito da memória *cache* sobre a DMA?

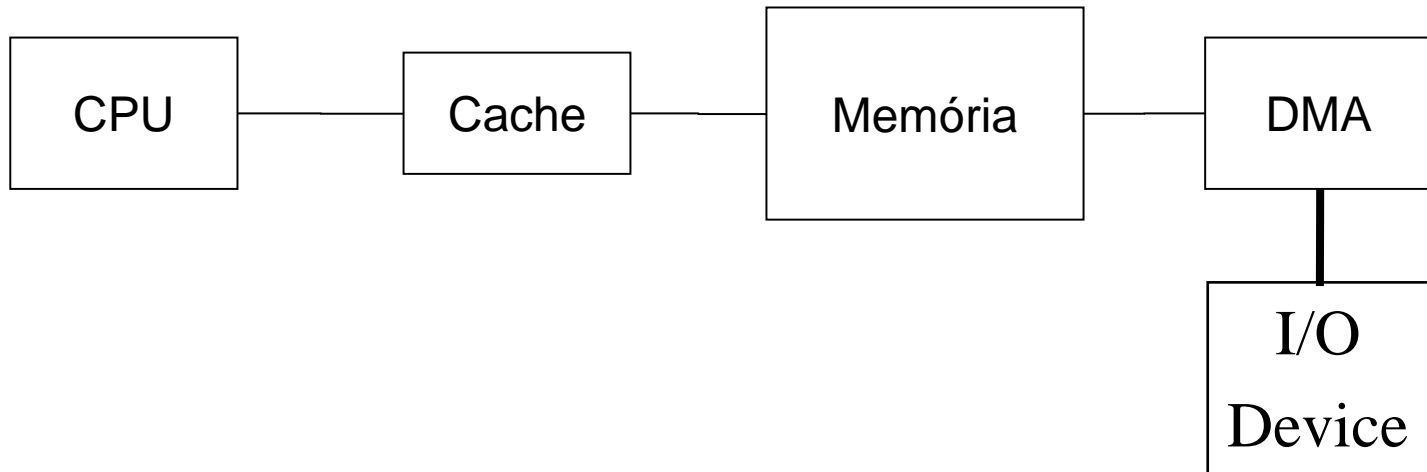
Há um “porém” quando se usa DMA

- Qual o efeito da memória *cache* sobre a DMA?



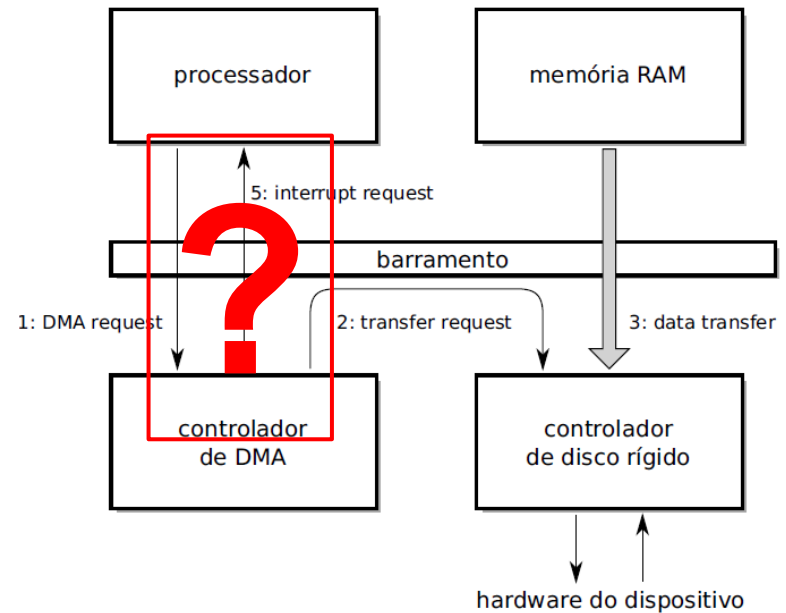
Há um “porém” quando se usa DMA

- Qual o efeito da memória *cache* sobre a DMA?
 - Dados obsoletos podem ser usados



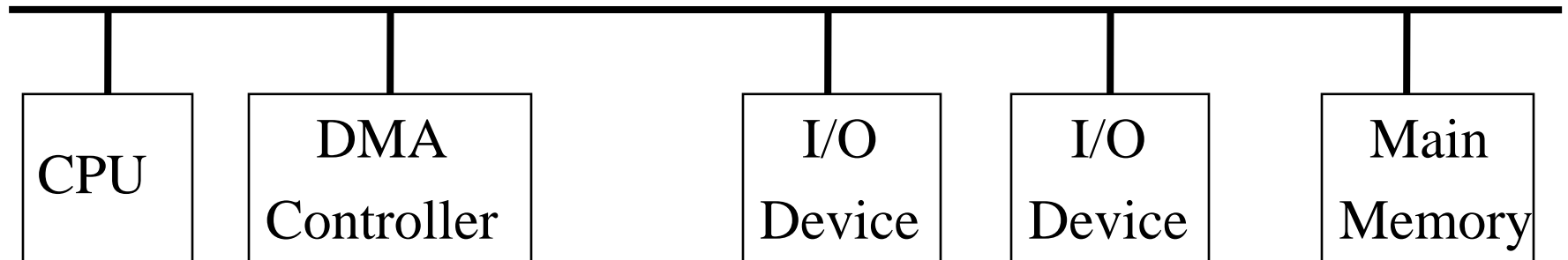
Configurações de DMA

1. Barramento único, DMA separado
2. Barramento único, DMA e E/S integrados
3. Barramento de E/S separado



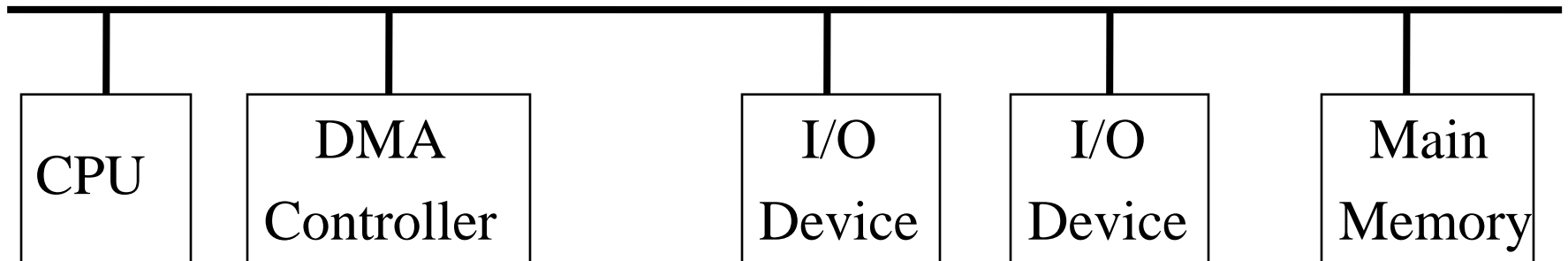
Configurações de DMA

- Barramento único, DMA separado
- Cada transferência usa o barramento duas vezes (Por que?)



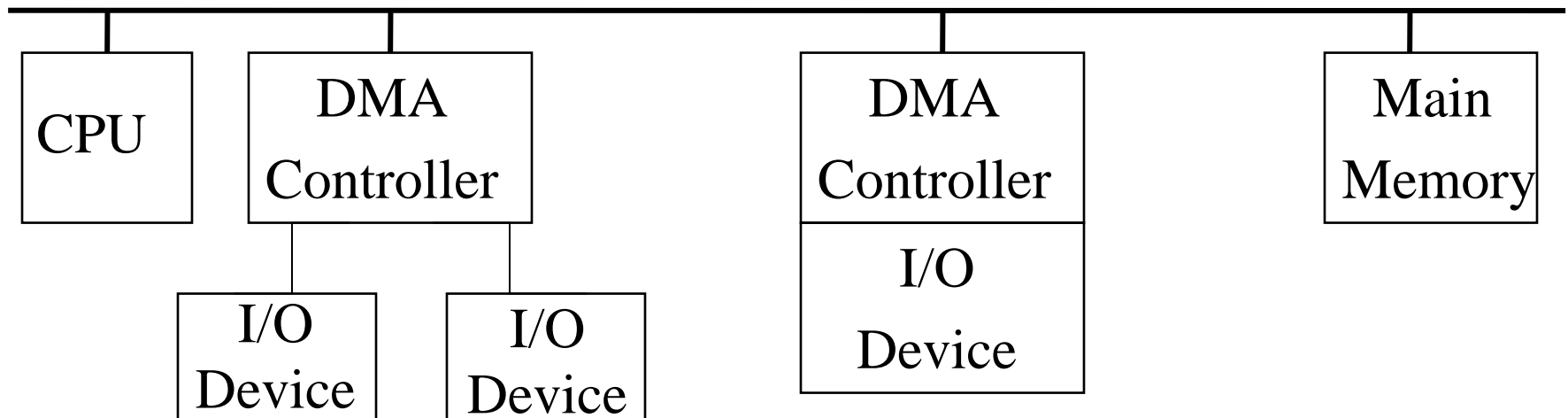
Configurações de DMA

- **Barramento único, DMA separado**
- Cada transferência usa o barramento duas vezes (Por que?)
 - De E/S para DMA e então de DMA para memória
- **⇒ A CPU é suspensa duas vezes**



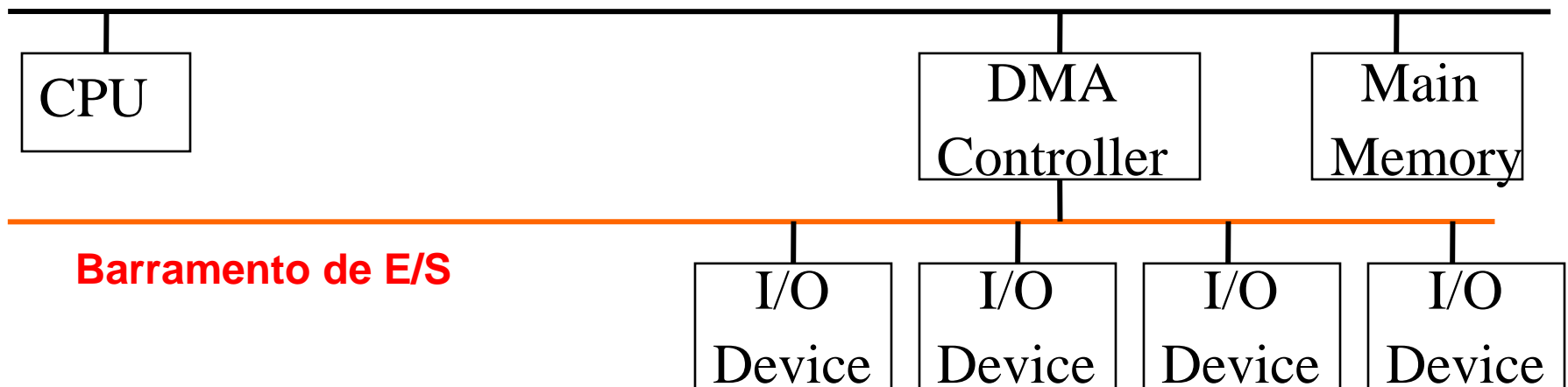
Configurações de DMA

- **Barramento único, DMA e E/S integrados**
- O controlador de DMA suporta diversos dispositivos
- Cada transferência usa o barramento apenas uma vez
 - De DMA para memória
- **⇒ A CPU é suspensa apenas uma vez**



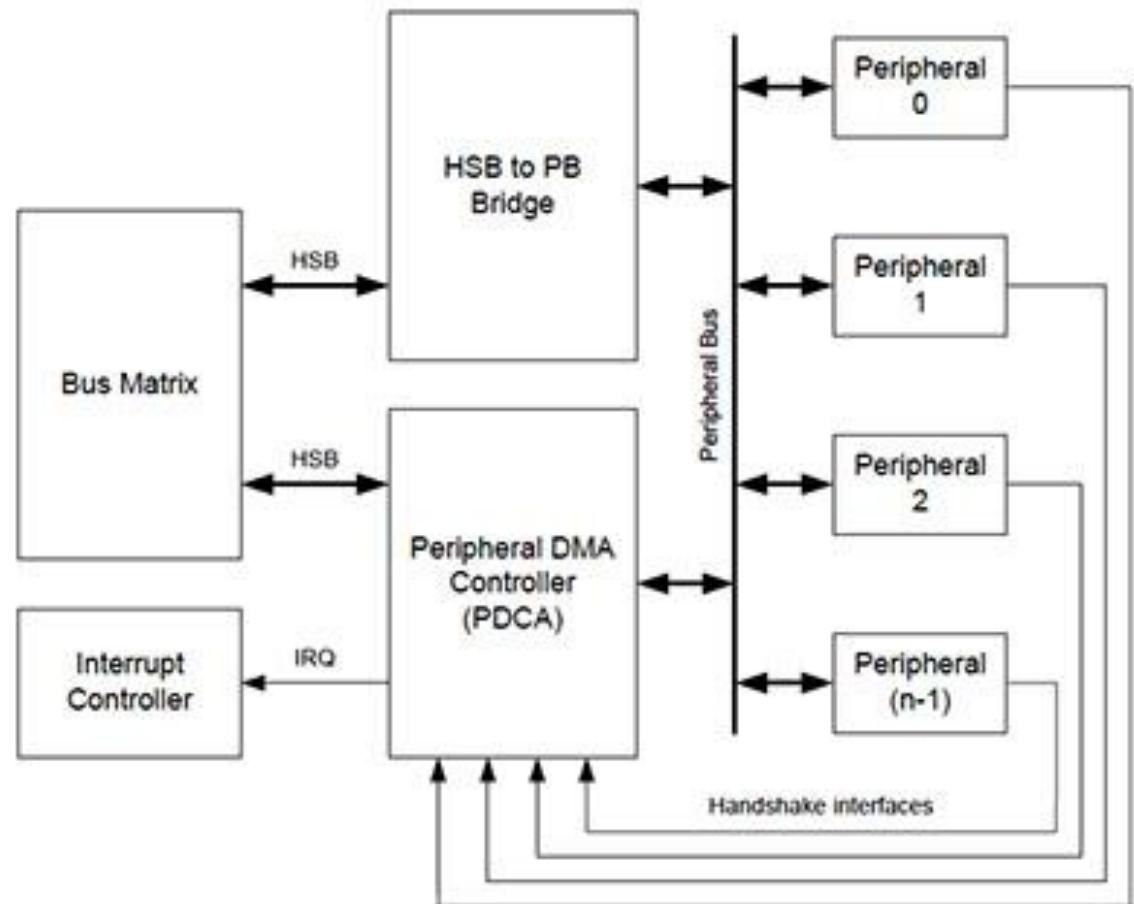
Configurações de DMA

- **Barramento de E/S separado**
- Um Barramento suporta todos os dispositivos habilitados em DMA
- Cada transferência usa o barramento do sistema apenas uma vez
 - De DMA para memória
- **⇒ A CPU é suspensa uma única vez**



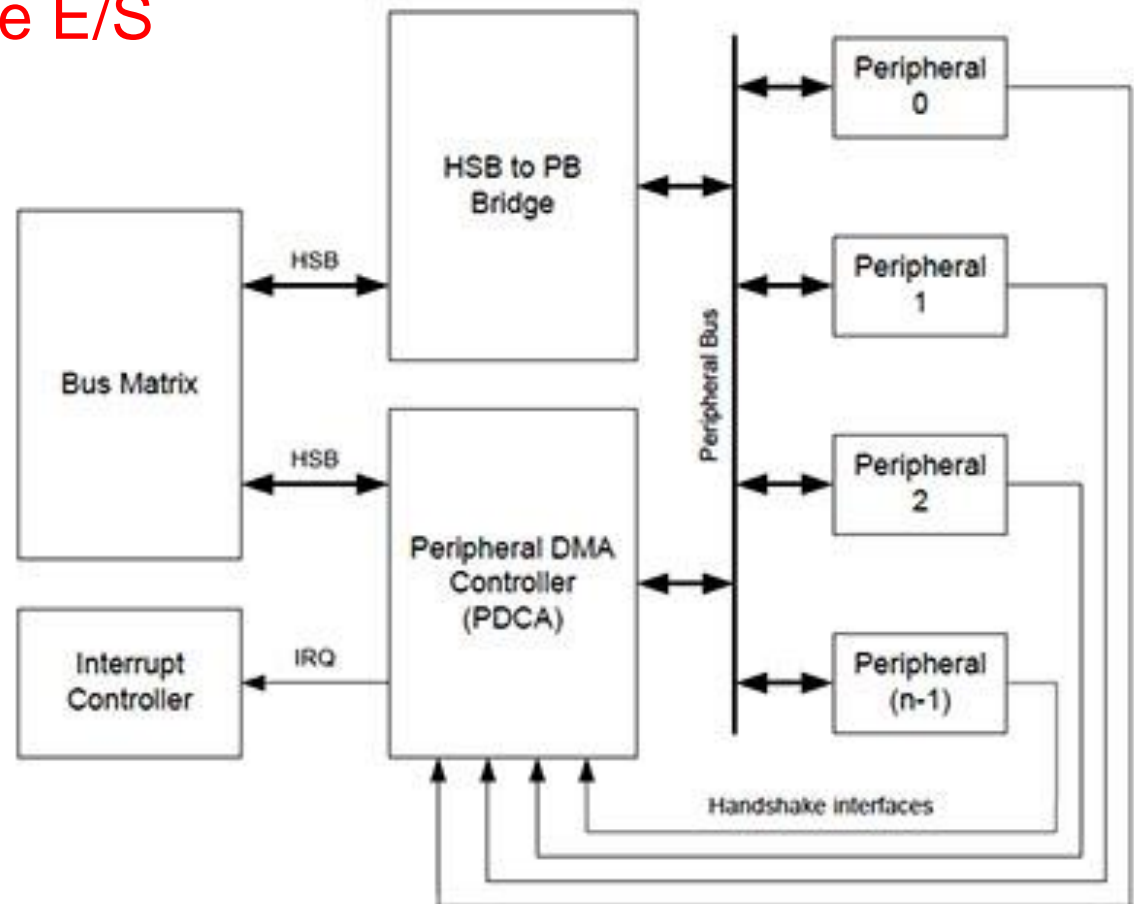
Configurações de DMA

- *Ex.: Atmel AT32UC3A MCU*



Configurações de DMA

- *Ex.: Atmel AT32UC3A MCU*
 - Barramento de E/S separado



Canais e Processadores de E/S



Canais e Processadores de E/S

- Dispositivos de E/S se tornaram mais sofisticados
 - ex.: interfaces de comunicação, placas gráficas 3D

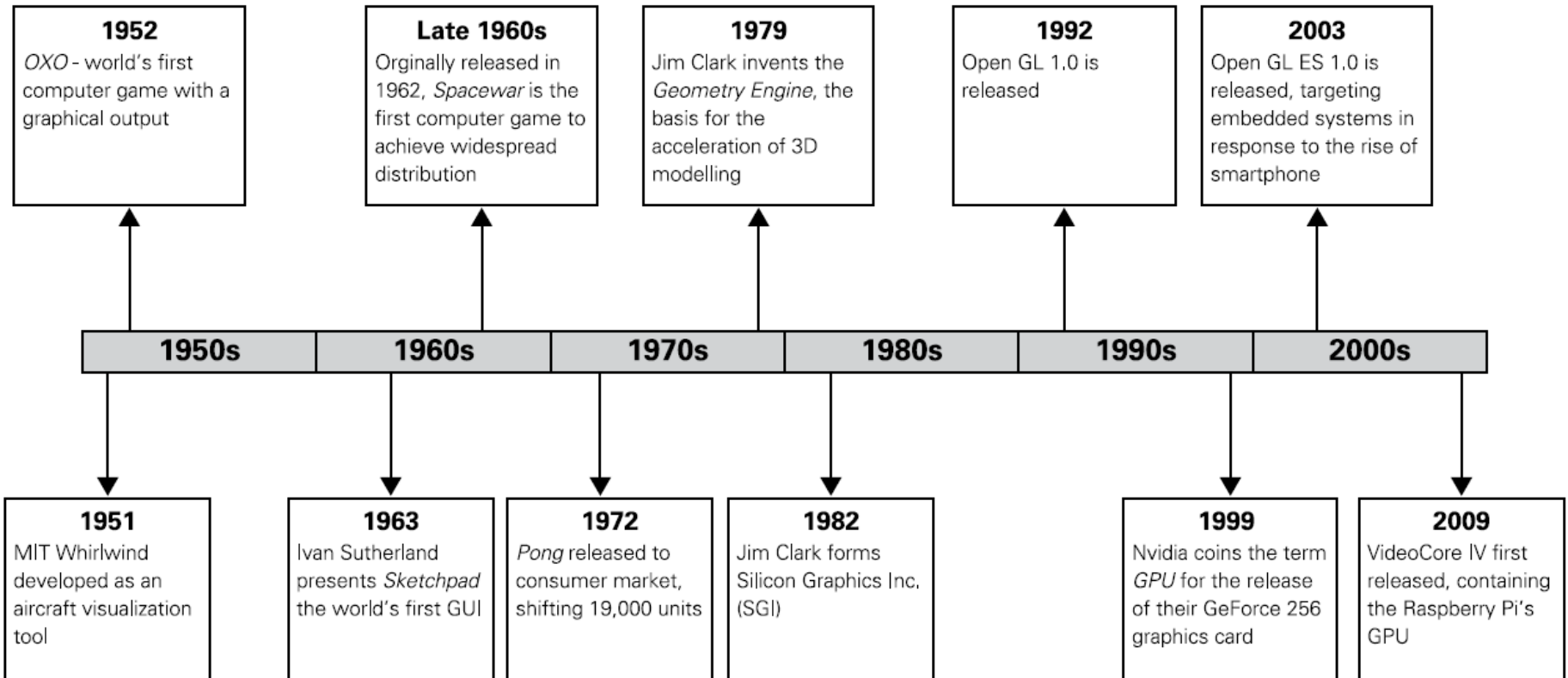
Canais e Processadores de E/S

- Dispositivos de E/S se tornaram mais sofisticados
 - ex.: interfaces de comunicação, placas gráficas 3D
- A CPU instrui o controlador de E/S para que faça a transferência

Canais e Processadores de E/S

- Dispositivos de E/S se tornaram mais sofisticados
 - ex.: interfaces de comunicação, placas gráficas 3D
- A CPU instrui o controlador de E/S para que faça a transferência
- O controlador faz toda a transferência
- Aumenta a velocidade
 - Deixa a CPU mais livre para outras tarefas
 - Processador dedicado é uma solução mais rápida

Evolução de sistemas gráficos



Evolução

1. CPU controla E/S diretamente (muito primitivo)
2. Módulo de E/S acrescentado
3. Módulo de E/S + operação de E/S com interrupção
4. Implementação com DMA

Evolução

1. CPU controla E/S diretamente (muito primitivo)
2. Módulo de E/S acrescentado
3. Módulo de E/S + operação de E/S com interrupção
4. Implementação com DMA
5. Módulo de E/S aprimorado → Processamento
 - Executa instruções de E/S
 - CPU direciona o módulo a executar um programa de E/S
 - Módulo executa instruções sem intervenção da CPU
 - A CPU é interrompida só ao final do processamento

Evolução

1. CPU controla E/S diretamente (muito primitivo)
2. Módulo de E/S acrescentado
3. Módulo de E/S + operação de E/S com interrupção
4. Implementação com DMA
5. Módulo de E/S aprimorado → Processamento
 - Executa instruções de E/S
 - CPU direciona o módulo a executar um programa de E/S
 - Módulo executa instruções sem intervenção da CPU
 - A CPU é interrompida só ao final do processamento
6. Módulo de E/S implementado com memória própria
 - Módulo → Computador independente
 - Um grande conjunto de E/S pode ser controlado
 - Envolvimento da CPU mínimo

Evolução

1. CPU controla E/S diretamente (muito primitivo)
2. Módulo de E/S acrescentado
3. Módulo de E/S + operação de E/S com interrupção
4. Implementação com DMA
5. Módulo de E/S aprimorado → Processamento

Executa instruções de E/S



A CPU é interrompida só ao final do processamento

6. Módulo de E/S implementado com memória própria

Módulo → Computador independente

Um grande conjunto de E/S pode ser controlado

Envolvimento da CPU mínimo

Evolução

1. CPU controla E/S diretamente (muito primitivo)
2. Módulo de E/S acrescentado
3. Módulo de E/S + operação de E/S com interrupção
4. Implementação com DMA
5. Módulo de E/S aprimorado → Processamento

Executa instruções de E/S

CP **Módulo → Canal de E/S** de E/S
Mó PU

A CPU é interrompida só ao final do processamento

6. Módulo de E/S implementado com memória própria

Mó **Módulo → Processador de E/S**
Un

Envolvimento da CPU mínimo

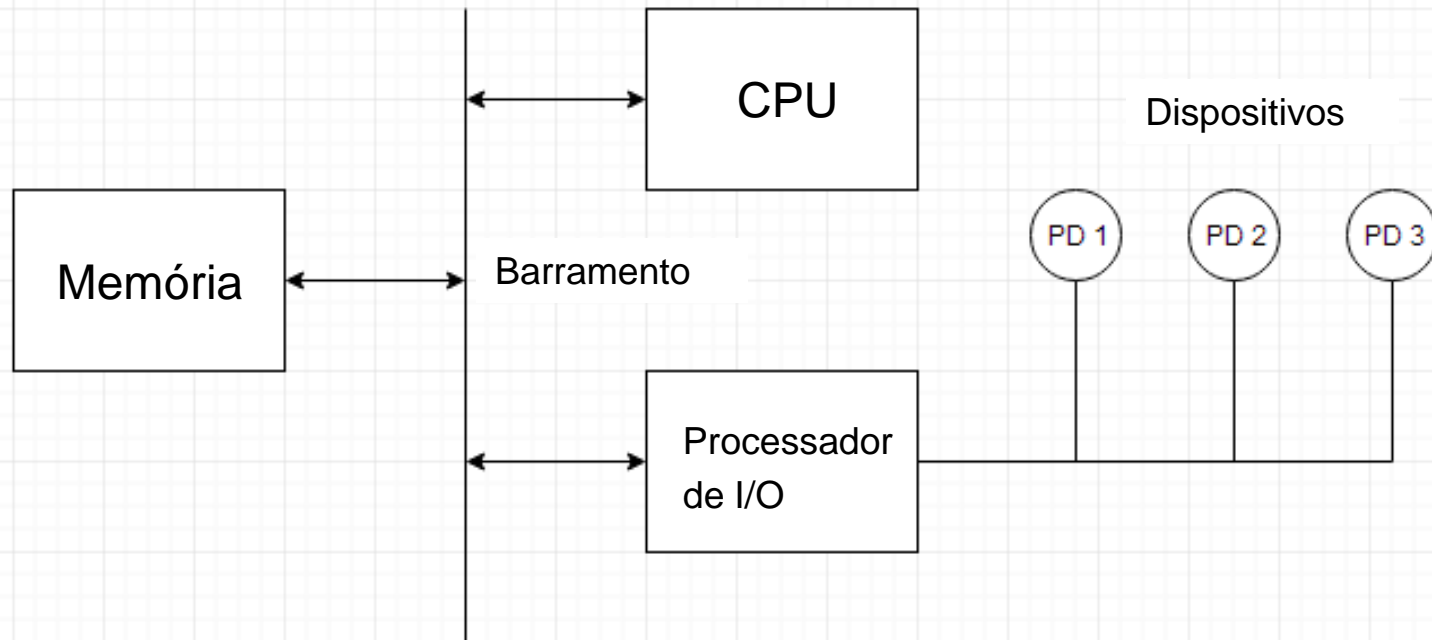
Canais (processadores) de E/S

- Extensão do conceito DMA
- CPU inicia E/S instruindo o canal
- Canal executa instruções de E/S não a CPU
- Instruções armazenadas em memória

{Canais} {processadores} de E/S

- Extensão do conceito DMA
- CPU inicia E/S instruindo o canal
- Canal executa instruções de E/S não a CPU
- Instruções armazenadas na memória {principal} {própria}
 - Especifica o(s) dispositivo(s)
 - Local de armazenamento (ou leitura) na memória
 - Prioridade
 - Ações em caso de erro

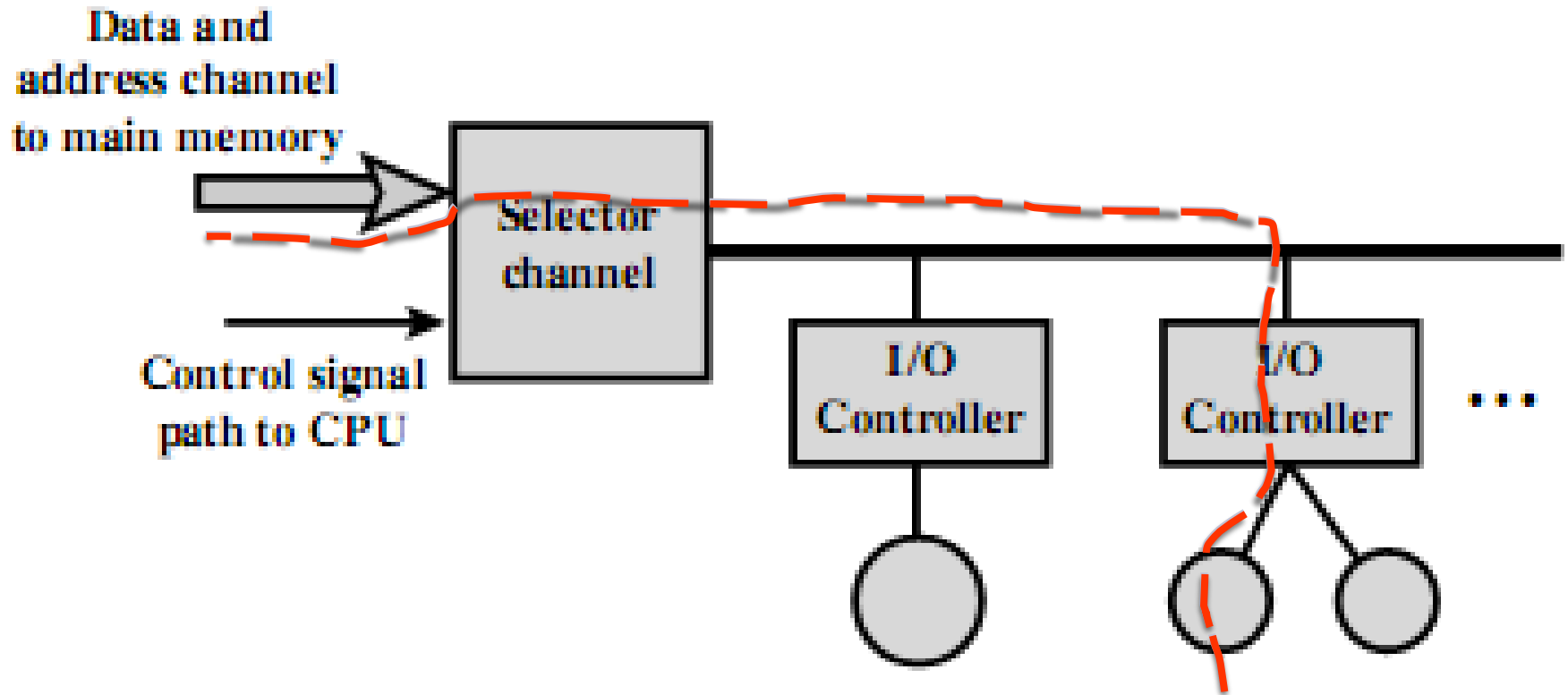
Canal de E/S



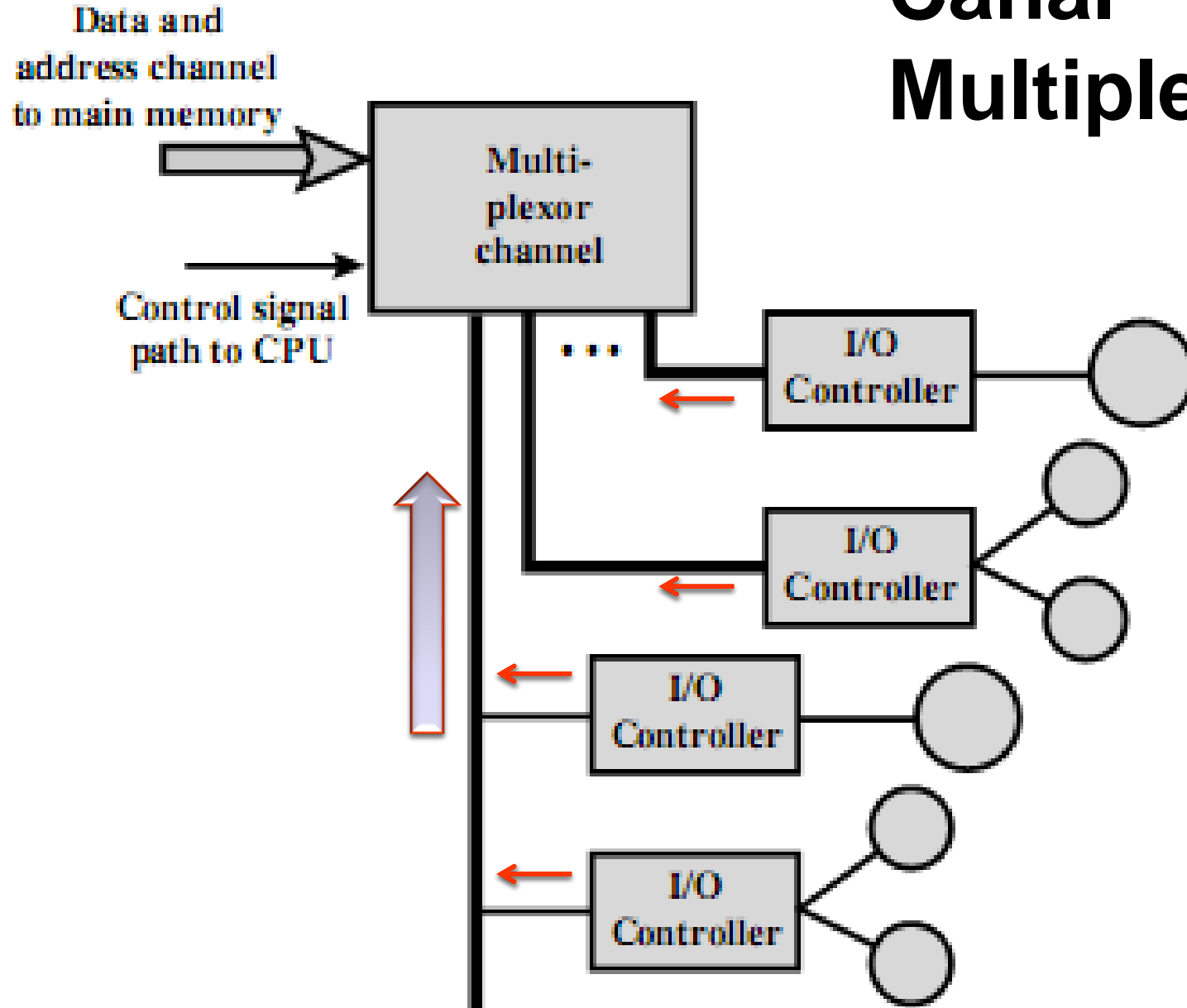
Arquiteturas

1. Canal Seletor
2. Canal Multiplexador

Canal Seletor



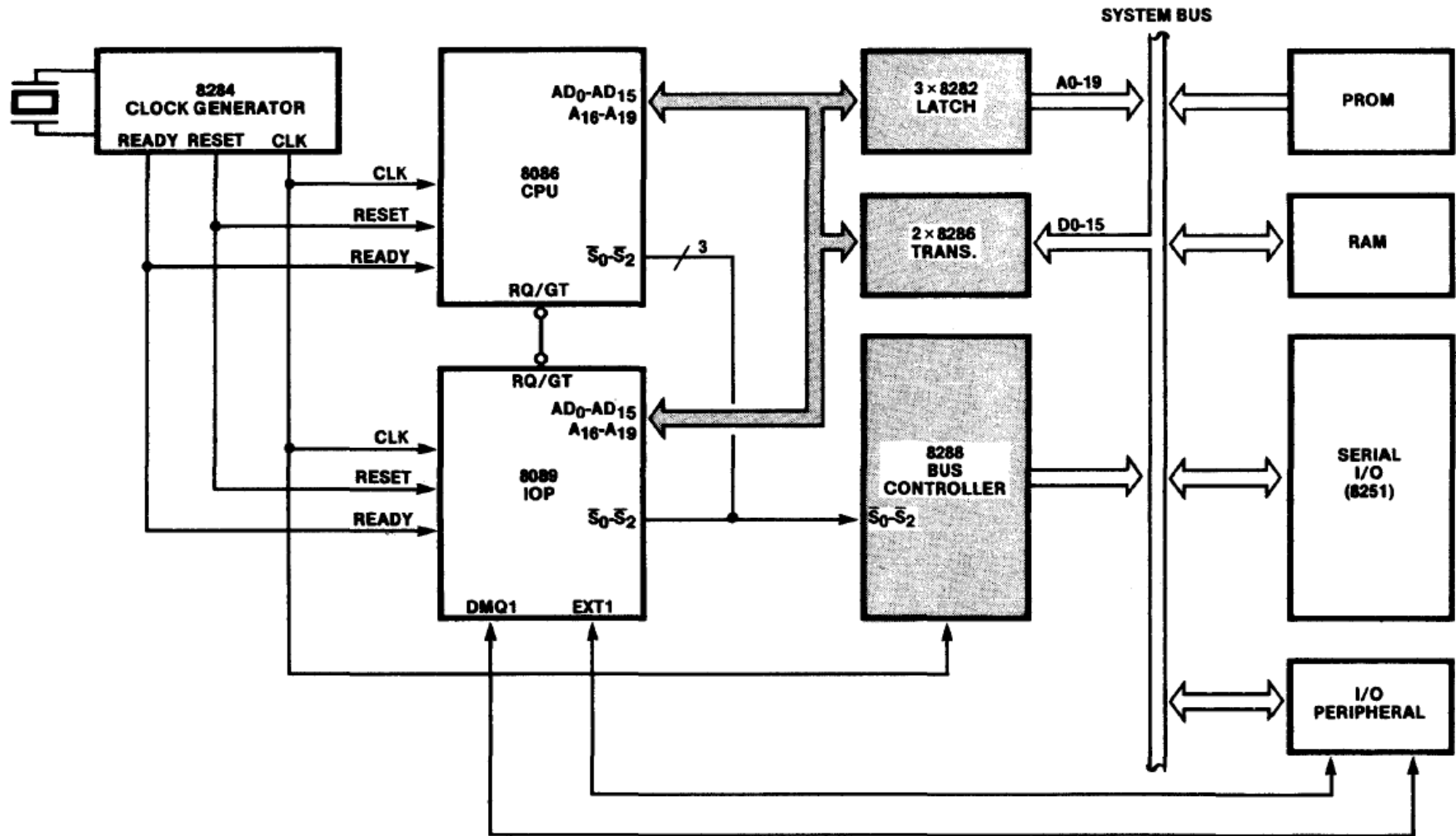
Canal Multiplexador



Co-processador de E/S da Intel



Co-processador de E/S da Intel

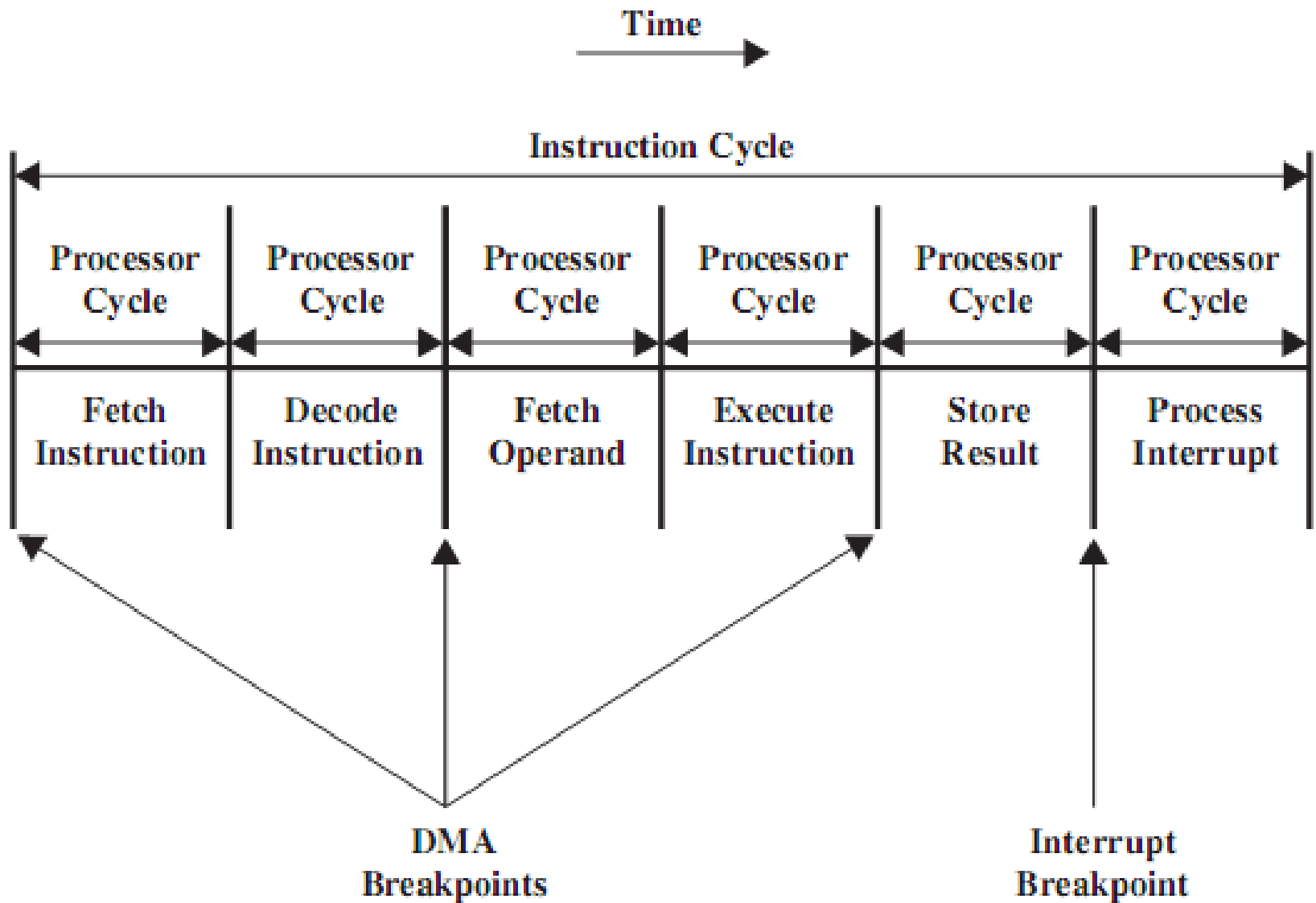


Exercício

Um computador contém um processador e um dispositivo **D** de **E/S** conectados à memória principal **M** por meio de um barramento compartilhado com uma **largura** do barramento de dados de **uma palavra**.

O processador pode executar no máximo **10^6 instruções por segundo**. Uma instrução requer, em média, **cinco ciclos de máquina**, **três** dos quais **usam o barramento de memória (dados)**.

Uma operação de leitura ou de escrita na memória gasta **um** ciclo de máquina. Suponha que o processador execute programas continuamente, que consomem **95%** da sua taxa de execução de instruções que não envolvem qualquer **instrução de E/S**. Suponha, ainda, que o ciclo do processador tem a mesma duração do ciclo de barramento e que o dispositivo de **E/S** é usado para transferir grandes blocos de dados entre **M** e **D**.

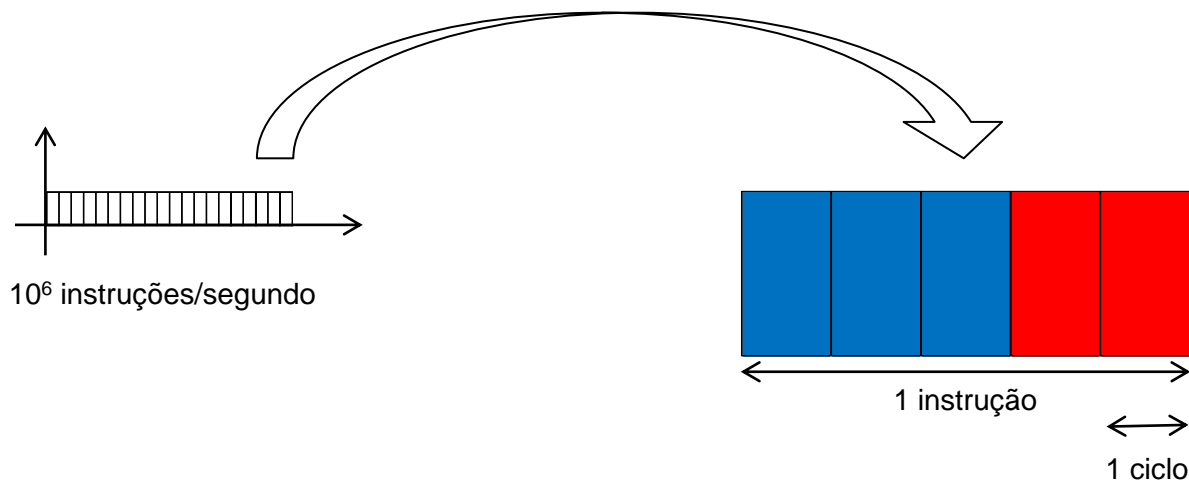


Exercício

Um computador contém um processador e um dispositivo **D** de **E/S** conectados à memória principal **M** por meio de um barramento compartilhado com uma **largura** do barramento de dados de **uma palavra**.

O processador pode executar no máximo **10^6 instruções por segundo**. Uma instrução requer, em média, **cinco ciclos de máquina**, **três** dos quais **usam o barramento de memória (dados)**.

Uma operação de leitura ou de escrita na memória gasta **um** ciclo de máquina. Suponha que o processador execute programas continuamente, que consomem **95%** da sua taxa de execução de instruções que não envolvem qualquer **instrução de E/S**. Suponha, ainda, que o ciclo do processador tem a mesma duração do ciclo de barramento e que o dispositivo de **E/S** é usado para transferir grandes blocos de dados entre **M** e **D**.



Exercício

Um computador contém um processador e um dispositivo **D** de **E/S** conectados à memória principal **M** por meio de um barramento compartilhado com uma **largura** do barramento de dados de **uma palavra**.

O processador pode executar no máximo **10^6 instruções por segundo**. Uma instrução requer, em média, **cinco ciclos de máquina**, **três** dos quais **usam o barramento de memória (dados)**.

Uma operação de leitura ou de escrita na memória gasta **um** ciclo de máquina. Suponha que o processador execute programas continuamente, que consomem **95%** da sua taxa de execução de instruções que não envolvem qualquer **instrução de E/S**. Suponha, ainda, que o ciclo do processador tem a mesma duração do ciclo de barramento e que o dispositivo de **E/S** é usado para transferir grandes blocos de dados entre **M** e **D**.

a) Se a **E/S programada** for usada e cada transferência de E/S de uma palavra exigir que o processador execute **duas instruções**, estime a taxa de dados de E/S média, em palavras por segundo, possível por meio de D.

b) Estime a taxa se o **DMA** for utilizado. Assuma que o módulo DMA pode usar todos os ciclos em que a CPU não acessa o barramento e ignore os tempos de setup e de checagem de estado.

Solução

- a. O processador dedica apenas 5% do seu tempo a operações de E/S. Logo a máxima taxa de execução de instruções de E/S é $10^6 \times 0,05 = 50.000$ instruções por segundo. Usando DUAS instruções para transferir UMA palavra. A taxa de transferência de E/S é, portanto, 25.000 palavras por segundo.

Solução

- b. O número de ciclos de máquina, por segundo, disponíveis para o controle do DMA é:

$$10^6 (0,05 \times 5 + \underline{0,95 \times 2}) = 2,15 \times 10^6 \text{ ciclos.}$$

Se assumirmos que o módulo DMA pode usar todos esses ciclos, e ignorar os tempos de setup e de checagem de estado, então esse valor é a taxa de transferência de E/S: $2,15 \times 10^6$ palavras/segundo.