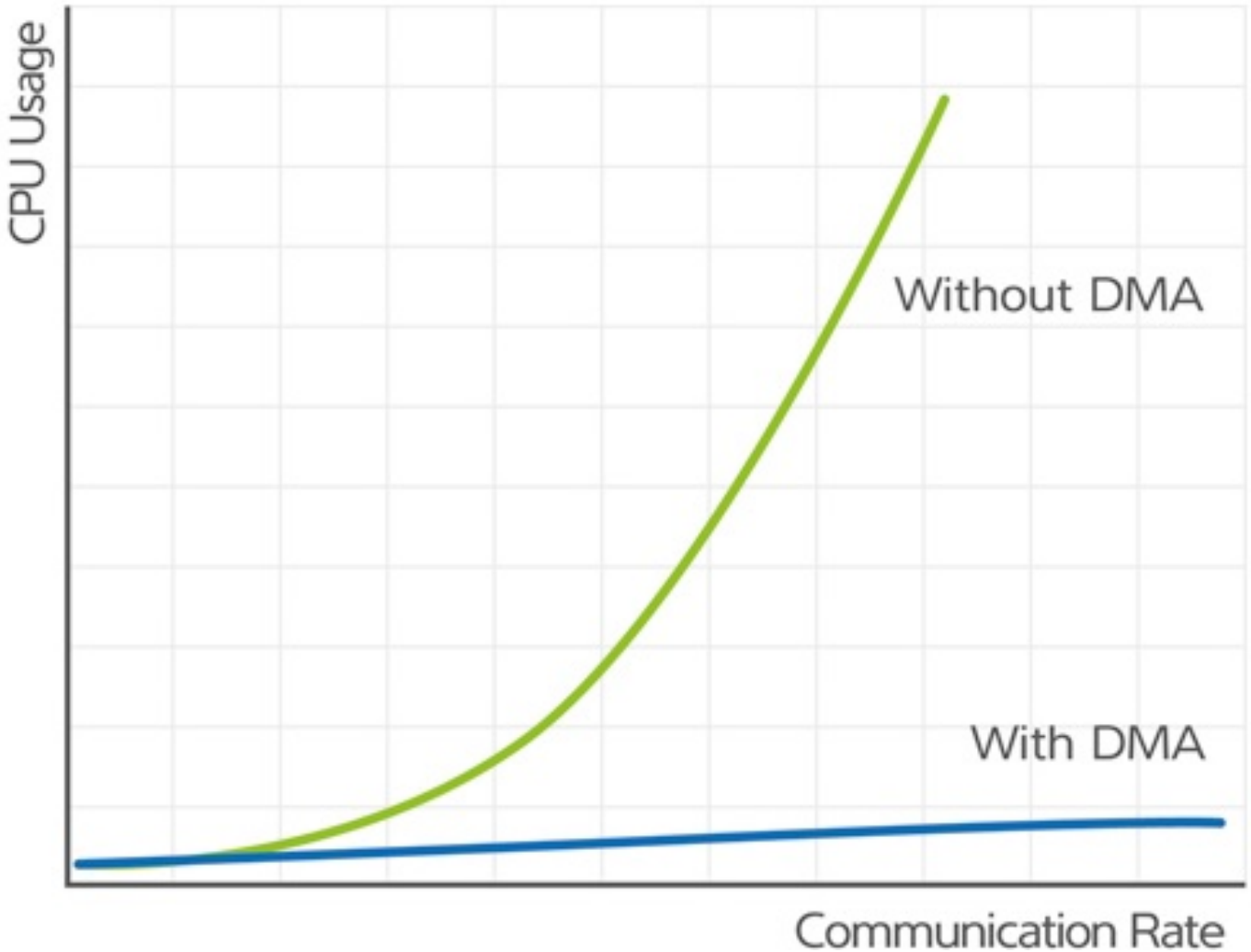
Aula 11 - Acesso Direto à Memória
Arquitetura de Computadores I

Prof. Wagner Guimarães Al-Alam

Universidade Federal do Ceará
Campus de Quixadá

2020-1

Por Que Precisamos de DMA



Por Que Precisamos de DMA

- I/O orientado a Polling e interrupção concentram a transferência de dados entre o processador e o dispositivo de I/O.
- Uma instrução para transferir dados (mov datain,R0) somente ocorre após o processador verificar que o dispositivo de I/O está pronto
  - Tanto verificando a flag de status na interface do dispositivo
  - Quanto aguardando o dispositivo enviar uma requisição de interrupção
- Uma sobrecarga considerável é encontrada, pois muitas instruções de programas devem ser executadas para cada palavra de dados transferida.
- Instruções são necessárias para incrementar o endereço de memória e manter o contador do trabalho.
- Com interrupções, uma sobrecarga adicional é associada por salvar e restaurar o contador de programa e outras informações de estado.

Direct Memory Access (DMA)

- Para transferir largos blocos de dados em alta velocidade, o DMA é uma alternativa a ser usada.
- Blocos de dados são transferidos entre um dispositivo externo e a memória principal, isso tudo sem a intervenção contínua do processador.

Controlador DMA

- Controlador DMA é parte da interface de I/O.
- Efetua as funções que normalmente seriam realizadas pelo processador no momento de acesso à memória principal. Para cada palavra transferida, ele prove o endereço de memória e todos os sinais de barramento de controle que controlam a transferência de dados.

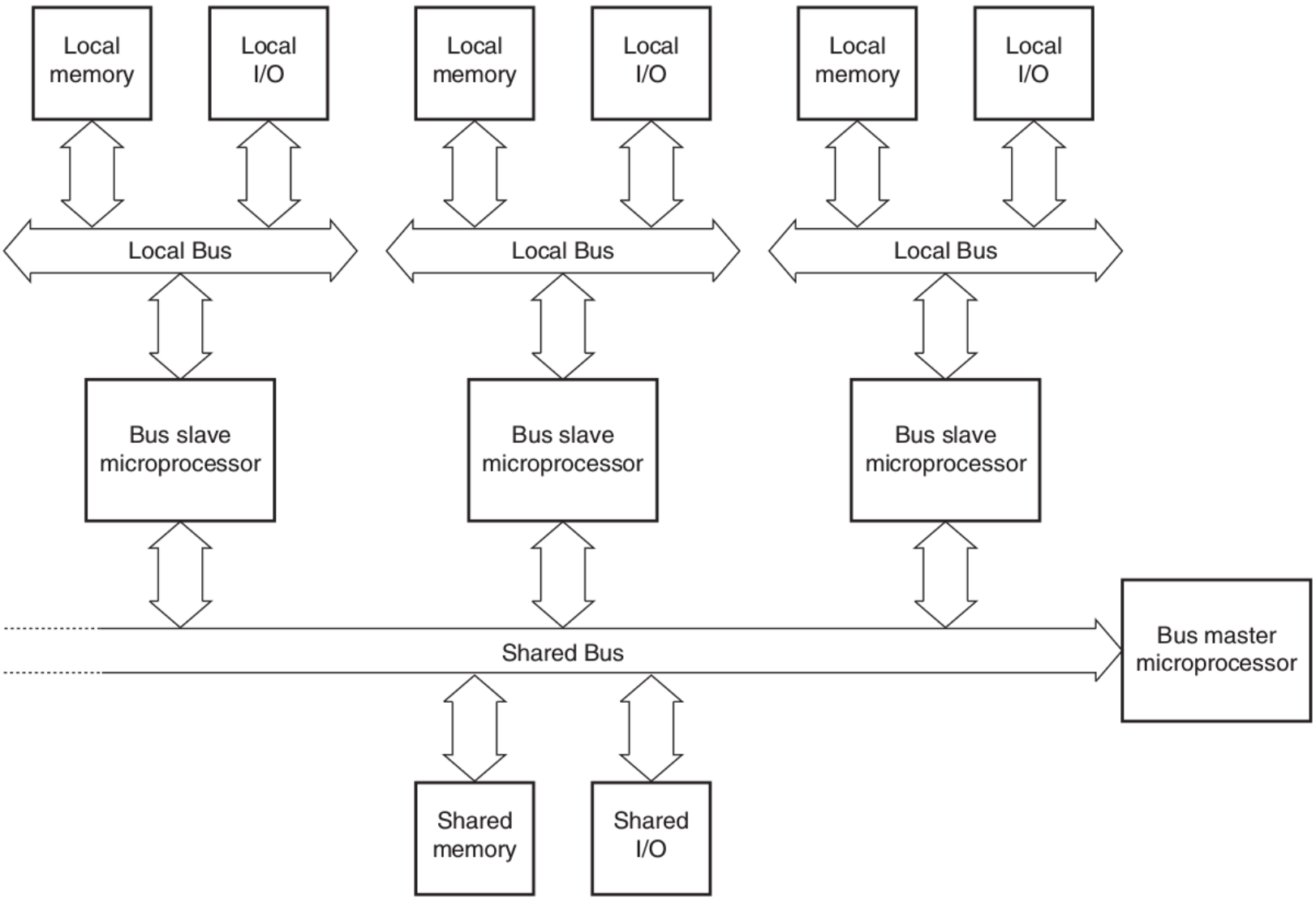
Controlador DMA

- O dispositivo que deseja executar DMA lança o sinal de solicitação do barramento dos processadores.
- O processador completa o ciclo de barramento e lança o sinal de confirmação *ack* no barramento.
- O dispositivo então lança o sinal de confirmação *ack* no barramento.
- O dispositivo DMA efetua a transferência da origem para o endereço de destino.
- Uma vez que as operações de DMA são completadas, o dispositivo libera o barramento lançando um sinal de release no barramento.
- O processado reconhece o sinal de release no barramento e continua o ciclo de barramento a parti de onde parou anteriormente.

Como o Sistema Operacional está Envolvido

- Operações de I/O são sempre feitas pelo SO em resposta a uma requisição de uma aplicação.
- O SO também é responsável por suspender a execução de um programa e iniciar outro.
  - O SO coloca o programa que está requisitando a transferência no estado bloqueado,
  - inicia uma operação DMA,
  - inicia a execução de outro programa.
- Quando a transferência estiver completa, o controlador DMA informa o processador enviando uma requisição de interrupção.
  - O SO coloca o programa suspenso no estado de pronto e agora pode se selecionado pelo agendados de processos para continuar a execução.

Arquitetura de Memória



Problema da Coerência de Cache

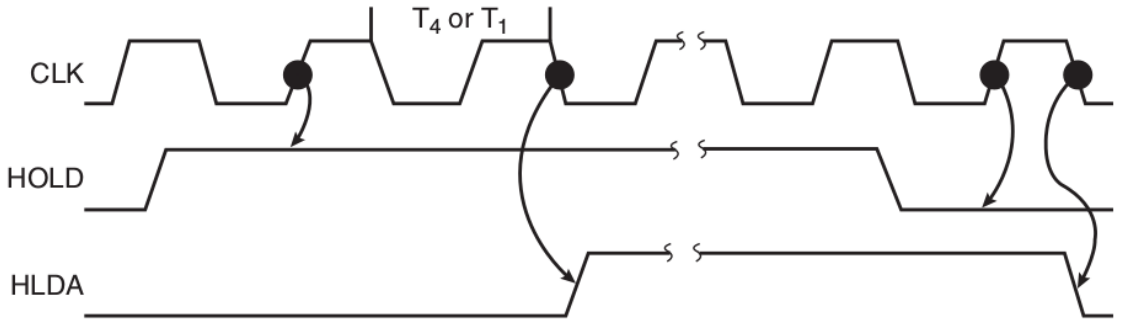
- Imagine um CPU equipado com uma memória cache e uma memória externa que pode ser acessada diretamente pelos dispositivos usando DMA. Quando a CPU acessa a localização X na memória, o valor corrente é armazenado na cache. Operações subsequentes em X irão atualizar a cópia da cache, mas não a versão na memória externa, assumindo que a estratégia de coerência de cache é write-back. Se a cache não for descarregada na memória antes do próxima vez que um dispositivo tente acessar X, o dispositivo irá receber o valor desatualizado.

Arbitração

- Arbitração de barramento é necessária para resolver conflitos entre mais de um dispositivo(DMA e processador, etc..) que tentam usar o barramento para acessar a memória principal.

Video

Como o DMA é Implementado

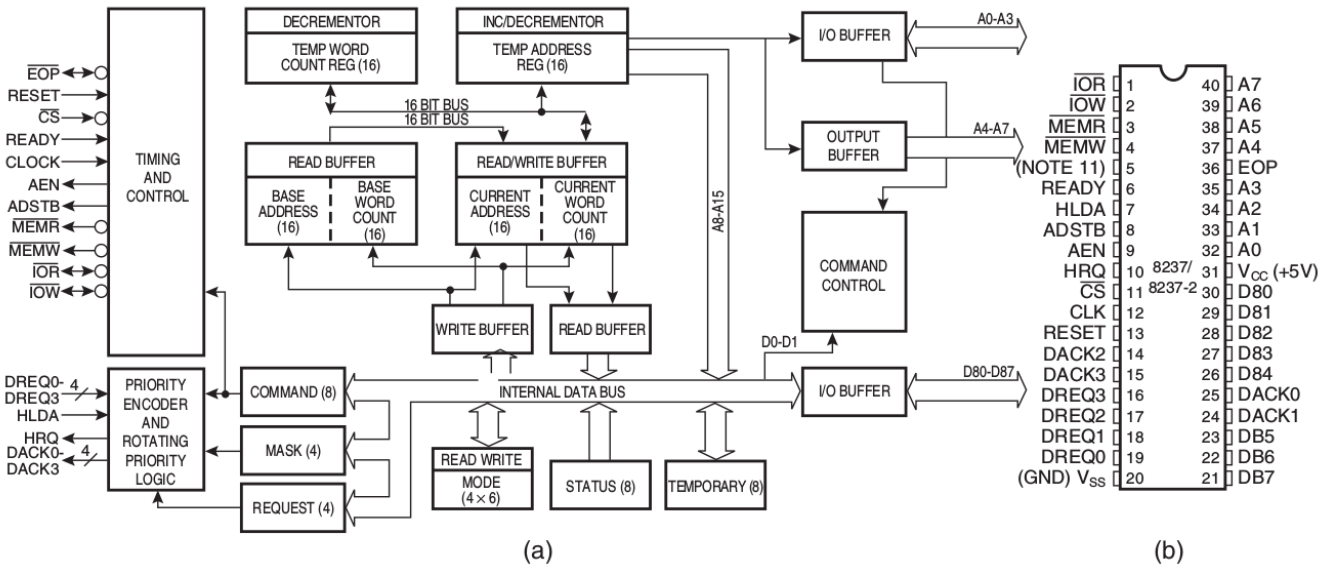


- HOLD: Controlador DMA requisita requisita o barramento ao processador com o sinal 1 no bit HOLD
  - HOLD pode ser executado durante o ciclo de instrução do CPU, diferente de interrupções que ocorrem somente no final.
- HLDA (hold acknowledge): Após alguns ciclos, o processador suspende a execução do programa e coloca os barramentos de controle, endereços e dados em modo de alta impedância.
  - Coloca os barramentos do processador no modo de alta impedância.
  - Permite que dispositivos externos possam ter acesso aos barramentos do sistema e a memória pode ser acessada diretamente.

Operações do DMA

- Operações do DMA:
  - DMA read: Transfere dados da memória para o dispositivo de I/O.
  - DMA write: Transfere dados do dispositivo de I/O para a memória.
- O controlador DMA ativa os sinais de controle necessários para uma operação em conjunto com a memória.

Controlador DMA Programável 8237A-5



(a) Diagrama de bloco e (b) pinagem do controlador

ISP (integrated system peripheral controller)

Nos computadores modernos, não encontramos o componente "controlador de DMA", mas ele está presente no chip controlador do sistema. No ISP há dois controladores DMA programáveis (como o 8237) e também provê um par de controladores de interrupção programáveis (como o 8259A).

Estado Atual

- Nos computadores modernos ocorreu a mudança para comunicação seral nas transferências de dados.
- O barramento PCI Express, que é serial, transfere dados a taxas que excedem as transferências DMA (podem chegar a 20Gbps).
- A interface SATA (serial ATA) para drives de disco usa transferência a taxas de 300 Mbps.

Referências

- BREY, Berry B.. The Intel Microprocessors. 8 edição. Prentice Hall. ISBN 9780135026458
- IRVINE, Kip R. Assembly Language for x86 Processors; 6 edição. Prentice Hall. ISBN 978- 0136022121
- `http://www.ece.tufts.edu/ee/107/`