

# Extras - DMA

## E/S com DMA

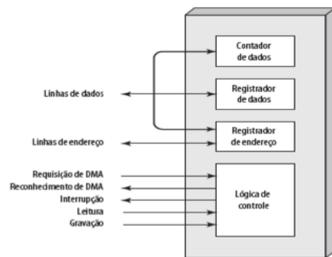
- Dados trocados entre Memória e dispositivos não precisam passar pela CPU
  - Ex: Memória Principal e Disco
- Com quantidade de dados elevada tem-se ganho de desempenho com DMA
- Exige lógica adicional razoavelmente complexa: processador dedicado à DMA que substitui a CPU nas operações de E/S com DMA
- Módulo adicional (hardware) conectado ao barramento



2

## E/S com DMA

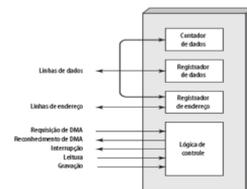
- Controlador de DMA ou Módulo de DMA



3

## E/S com DMA

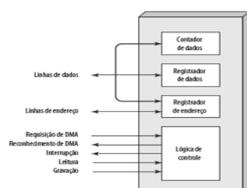
- Processador envia:
  - indicação de operação de E/S pela linha de Leitura ou Escrita
  - endereço do dispositivo de E/S pelas linhas de endereço
  - Endereço inicial de memória pela linha de dados -> Registrador de Endereço
  - Número de palavras a serem lidas /escritas pela linha de dados -> Contador de Dados



4

## E/S com DMA

- DMA:
  - Transfere o bloco de dados, uma palavra por vez, diretamente, de ou para a memória principal
  - Sinal de Interrupção



5

## E/S com DMA

- CPU programa a DMA para transferir bloco de dados
  - CPU continua processamento
  - DMA executa transferência
- DMA encerra transferência
  - Envia interrupção ao processador
  - CPU lê estado do Módulo de DMA
- Processador executa próxima instrução

6

## E/S com DMA

- Durante a transferência feita pela DMA, é como se a CPU fosse “desativada”
- O Controlador de DMA (o “processador dedicado”) tem controle do barramento de endereços, dados e controle, durante o tempo em que a CPU executa operações internas.

{ 7 }

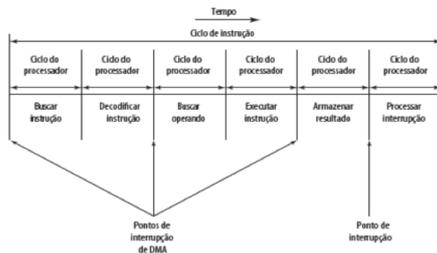
## E/S com DMA

- Duas maneiras de implementar uma DMA
  - 1. Força o uso do barramento, suspendendo o processador temporariamente
    - Técnica conhecida como Roubo de Ciclo
    - Redução no processamento
    - Não há interrupção - CPU não troca de contexto.
    - Atrasa a CPU, mas não tanto quanto a CPU fazendo transferência.

{ 8 }

## E/S com DMA

- Roubo de Ciclo
  - CPU suspensa logo antes de acessar o barramento.



{ 9 }

## E/S com DMA

- 2. Usando de modo transparente o barramento
  - Transferência é mais lenta
  - Usa quando o processador não quer

{ 10 }

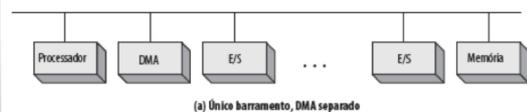
## E/S com DMA

- Transferências por DMA usam as mesmas vias de dados e endereços utilizadas pelo processador
  - Barramento: Como é organizado?

{ 11 }

## E/S com DMA

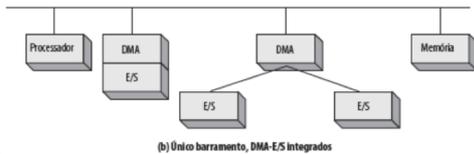
- Barramento único
  - DMA atua tal como um processador para todos dispositivos de E/S
  - Cada transferência usa barramento duas vezes - E/S para DMA e DMA para memória
  - Esta configuração é de menor custo, mas ineficiente



{ 12 }

### E/S com DMA

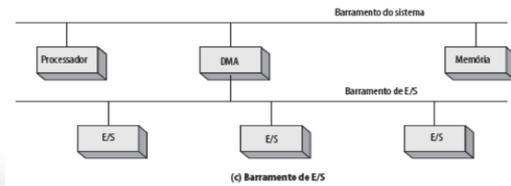
- Barramento único, DMA – E/S integrados
  - Há várias linhas entre dispositivos e DMA que não passam pelo barramento do sistema
  - Cada transferência usa barramento uma vez - DMA para memória



13

### E/S com DMA

- Barramento único, DMA – E/S integrados
  - Há várias linhas entre dispositivos e DMA que não passam pelo barramento do sistema
  - Cada transferência usa barramento uma vez - DMA para memória



14

### E/S com DMA

- As duas últimas técnicas usam somente uma vez o barramento do sistema
  - Isso permite a comunicação entre DMA e dispositivo em paralelo com o uso do barramento do sistema pelo processador

15

### E/S com DMA

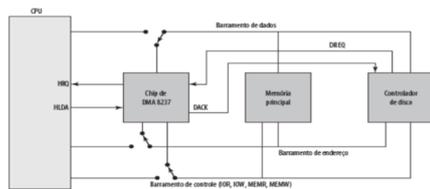
#### Exemplo

- Controlador de DMA Intel 8237A
  - Interfaces com família 80x86 e DRAM
  - Quando o módulo de DMA precisa de barramentos, ele envia sinal HOLD ao processador
  - CPU responde HLDA (hold acknowledge)
  - Módulo de DMA pode usar barramentos

16

### E/S com DMA

Ex: transferir dados da memória para o disco



DACK = DMA acknowledge (reconhecimento de DMA)  
 DREQ = DMA request (requisição de DMA)  
 HLDA = HOLD acknowledge (reconhecimento de HOLD)  
 HRQ = HOLD request (requisição de HOLD)

17

### E/S com DMA

1. Dispositivo requisita serviço de DMA levantando DREQ (requisição de DMA)
2. DMA levanta sua linha HRQ (hold request)
3. CPU termina ciclo de barramento e levanta linha HLDA (hold acknowledge). HOLD permanece ativo pela duração do DMA
4. DMA ativa DACK (DMA acknowledge), para dispositivo iniciar transferência.
5. DMA inicia transferência:
  - endereço do primeiro byte no barramento de endereço
  - ativa MEMR
  - ativa IOW para escrever no periférico.
6. DMA decrementa contador e incrementa ponteiro de endereço. Repete até contagem chegar a zero.
7. DMA desativa HRQ, retornando o controle do barramento de volta à CPU.

18