Universidade de São Paulo - São Carlos,SP
Instituto de Ciências Matemáticas e de Computação

USP - ICMC - SSC
SSC 0510 - Informática - 2o. Semestre 2010

Disciplina de Arquitetura de Computadores

Prof. Fernando Santos Osório
Email: fosorio [at] { icmc. usp. br , gmail. com }
Página Pessoal: http://www.icmc.usp.br/~fosorio/

Estagiário PAE Maurício Dias - Email: acdias29 [at] yahoo.com.br

Material on-line: COTEIA - http://coteia.icmc.usp.br

Aula 09

Aula 06 - Tópicos Abordados

Conteúdos Abordados:

1. **CPU: Técnicas de Otimização**
   - > Técnicas de Pipeline
   - > Pré-Fetch de Instrução
2. **Memória - Organização da Memória**
   - > Hierarquia de Memória
   - > Registradores
   - > Memória Cache
   - > Memória Principal
   - > Memória Secundária
   - > Memória Virtual
3. **Entrada e Saída**
   - > Bancos de Memória de I/O
   - > Barramentos e DMA
   - > E/S por Polling e por Interrupção

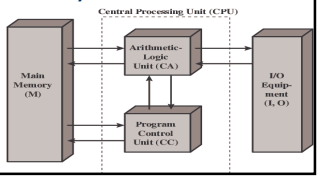
1. CPU: Técnicas de Otimização

CPU: Otimizações

Memória:
Registradores x Memória Principal x Memória Secundária
(Interno a CPU) (RAM/ROM) (Disco / Fita)
☺

Intruções / Execução:
- Pipeline de Execução
- Conjunto amplo de Registradores Internos
- Cache On Board
- Cache On board: L1 & L2
- Pré-Carga (*Pre-Fetch*) e Predição de Desvios (*Branch prediction*)
- Análise de Fluxo de Execução (*Data flow analysis*)
- Execução Especulativa (*Speculative execution*)

E/S:
- Interrupções
- DMA (*Direct Memory Access*)



USP - SSC-510 - Informática T1 / 2009-2
Arquitetura de Computadores
Prof. Fernando Osório

1. CPU: Técnicas de Otimização

CPU: Otimizações

Intruções / Execução:
- Pipeline de Execução - Explorando o paralelismo na execução

Neander: Instrução LDA (carrega acumulador)

Simbólico: LDA end

RT: AC ← MEM(end)

Passos no nível RT:

Busca da Instrução
Incrementa o PC

Busca:	REM ← PC
	Read; PC ← PC + 1
	RI ← RDM
Execução:	REM ← PC
	Read; PC ← PC + 1
	REM ← RDM
	Read
	AC ← RDM; atualiza N e Z

1. CPU: Técnicas de Otimização

CPU: Otimizações

Intruções / Execução:
- Pipeline de Execução - Explorando o paralelismo na execução

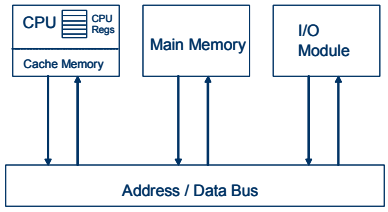
Paralelismo entre etapas da execução de uma instrução

> Busca da Instrução
> Incrementa o PC
> Busca do Operando
> Incrementa o PC
> Armazena o Resultado

> Decodifica Instrução
> Executa Instrução (ULA, E/S, ...)

> Busca da próxima instrução

Registrador
Memória
Cache



Pipeline de Execução

1. CPU: Técnicas de Otimização

Pipeline

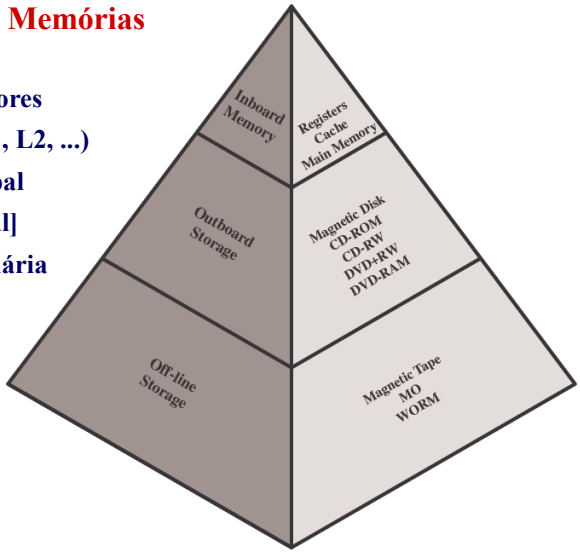
Ciclos de Operação da CPU
Estágios do Pipeline
Pre-Fetch
Previsão de Desvio

Material complementar
William Stallings
Computer Organization and Architecture (Book)
Chapter 12 [Trad. E.Simões / F.Osório]

2. Memória: Organização da Memória

Hierarquia de Memórias

CPU - Registradores
CPU - Cache (L1, L2, ...)
Memória Principal
[Memória Virtual]
Memória Secundária



2. Memória: Organização da Memória

Performance

- Tempo de Acesso / *Access time*
  - Tempo entre apresentar o endereço e obter o dado válido.
- Ciclo de Memória / *Memory Cycle time*
  - Tempo que pode ser necessário para a memória se "recuperar" antes de um próximo acesso.
  - Tempo de ciclo = *access* + *recovery*
- Taxa de Transferência / *Transfer Rate*
  - Taxa na qual os dados são movidos.

Tipos Físicos de Memórias

- Semicondutor
  - RAM, ROM, Flash
- Magnética
  - Disco e Fita
- Ótica
  - CD & DVD
- Outros
  - Bubble memory
  - Holográfica

Memória RAM e ROM

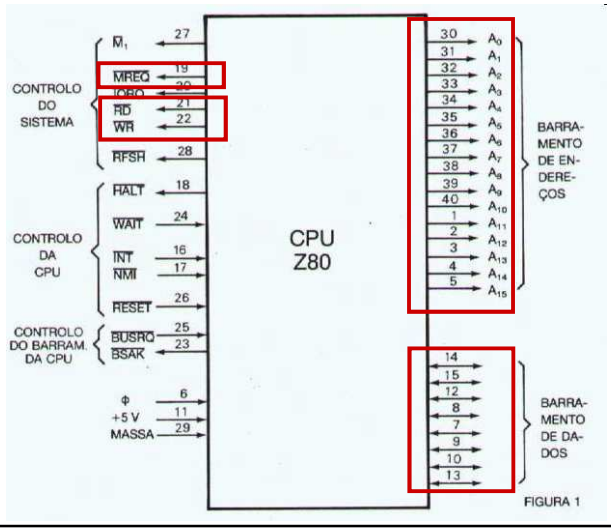


FIGURA 1

ZX Spectrum
Z80

2. Memória: Organização da Memória

Características Físicas

DRAM
SRAM
ROM
PROM
FLASH

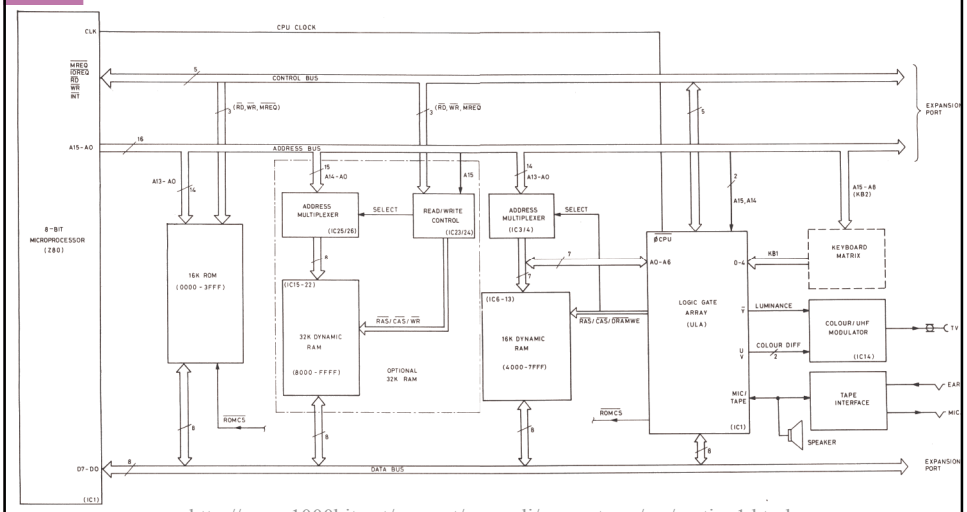
- Decaimento (*Decay*)
- Volatilidade (*Power off*)
- Gravável (*Erasable*)
- Consumo de energia
- Velocidade de acesso
- Capacidade de armazenamento

USP - SSC-510 - Informática T1 / 2009-2
Arquitetura de Computadores
Prof. Fernando Osório

2. Memória: Organização da Memória

DRAM + Refresh + I/O (Video)

ZX Spectrum
Z80



http://www.1000bit.net/support/manuali/zxspectrum/sm/section1.html

Set. 2009

2. Memória: Organização da Memória

DRAM + Refresh + I/O (Video)

ZX Spectrum
Z80

Dynamic Memory Refresh

The CPU incorporates built-in **dynamic RAM refresh circuitry**. As part of the instruction OP code fetch cycle, the CPU performs a memory request after first placing the refresh address on the lower eight bits of the address bus. At the end of the cycle the address is incremented so that over 255 fetch cycles, each row of the dynamic RAM is refreshed.
This mechanism only applies to the optional 32k expansion RAM in the the 48k Spectrum.
An alternative refresh method is adapted for the standard 16k RAM.

MEMORY ORGANISATION

In the standard 16k Spectrum there are 32k bytes of addressable memory equally divided between **ROM** and **RAM**.
The lower 16k bytes of memory (addresses 0000 - 3FFF) are implemented in a **ROM** (IC5) which holds the monitor program. This program is a complex Z80 machine code program divided broadly into three parts one each covering the input/output routines, the BASIC interpreter and expression handling.
The upper 16 bytes of memory (addresses 4000 - 7FFF) are implemented using eight 16k bit **dynamic RAMs** (IC6-IC13). Approximately half of this space is available to the user for writing BASIC or machine code programs.
The remainder is used to hold the system variables including 6k bytes reserved for the memory mapped **display area**.

USP - SSC-510 - Informática T1 / 2009-2
Arquitetura de Computadores
Prof. Fernando Osório

2. Memória: Organização da Memória

DRAM + Refresh + I/O (Video)

ZX Spectrum
Z80

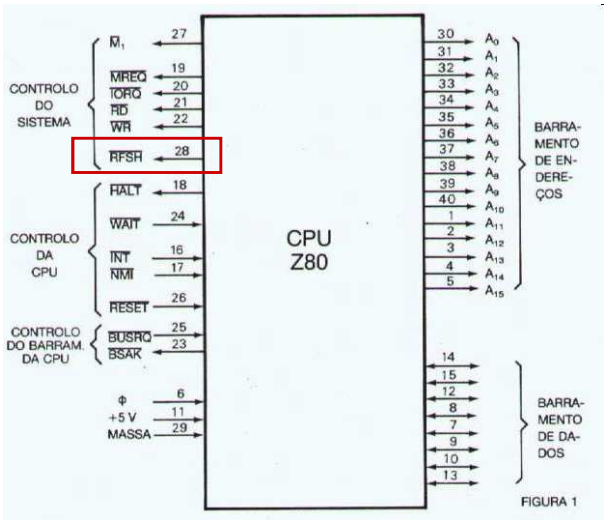


FIGURA 1

Organização

- Arranjo físico dos bits em palavras
- Nem sempre é óbvia

  e.g. memória entrelaçada (*interleaved*)

  páginas de memória

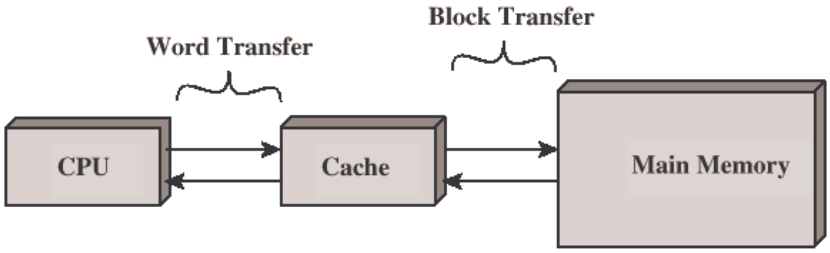
  memória em planos de bits (*m*k x *n*bits)

Lista de Hierarquia

- Registradores
- Cache L1
- Cache L2
- Memória principal
- Cache de Disco
- Disco
- Ótica
- Fita

Cache

- Pequena quantidade de memória mais rápida
- Se localiza entre a CPU e a Memória Principal
- Pode fazer parte do chip da CPU ou de um módulo



Operação do Cache – Visão Geral

- CPU requer o conteúdo de uma posição de memória
- Verifica se o dado deste endereço está no cache
- Se tem no cache, localiza e lê os dados, lê da memória cache (+ rápida)
- Se não tem no cache, lê os dados (bloco de dados) da memória principal para a memória cache
- Repassa os dados do cache para a CPU
- O cache inclui tags (marcas) para identificar qual bloco da memória principal que está em qual bloco (slot) slot do cache.

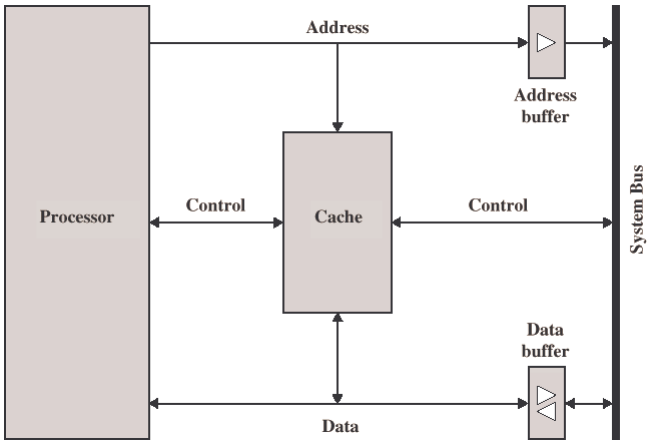
Projeto do Cache

- Tamanho
- Função de Mapeamento
- Algoritmo de Atualização
- Política de escrita
- Tamanho do bloco
- Número de caches

Size does matter ("Tamanho faz diferença")

- Custo
  - O custo de ter mais memória de cache é elevado.
- Velocidade
  - Mais cache => Mais rápido (até certo ponto);
  - Verificar os dados presentes no cache consome um certo tempo.

Organização Típica de Cache



I/O – Tipos de E/S: Comandos

- Dois métodos são usados para acessar os dispositivos de I/O
  - I/O mapeado em memória (Acesso a memória de I/O – LDA, STA).
  - Instruções especiais de I/O (CPU Instruction Set: InPort, OutPort).
- **I/O Mapeado em Memória:**
  - Reserva uma parte do endereçamento de memória para acesso aos dispositivos de I/O. Escreve e lê dados nesta área. Exemplo: 0xF000 (área reservada para memória de vídeo)
  - Comandos de acesso a memória são interpretados como I/Os.
- **I/O por Instruções Especiais:**
  - Instruções Especiais de I/O identificam o número do dispositivo de I/O que se deseja acessar. Exemplo: InPort 0x01FF (área reservada para status do dispositivo)

I/O – Tipos de E/S: Métodos

*** Programado:**

- Baseado em uma rotina que é responsável pela transferência dos dados para o dispositivo de I/O;

*** Por Interrupção:**

- Baseado em um conjunto de Hardware+Software que são usados para controlar a transferência dos dados. O hardware sinaliza quando pode receber/enviar dados e uma rotina de software "atende" o pedido (interrupção), realizando o I/O;

*** Por Acesso Direto a Memória: DMA**

- A transferência dos dados é realizada diretamente entre o dispositivo de I/O e a memória, sem a necessidade de uma interferência direta da CPU.

I/O - E/S por Polling e por Interrupção

O Sistema Operacional precisa saber quando
- Um dado está disponível para transferência (leitura ou escrita);
- O dispositivo de completou a operação requisitada;
- Ocorreu um erro de I/O durante uma operação.

Estas operações podem ser feitas de 2 modos principais:

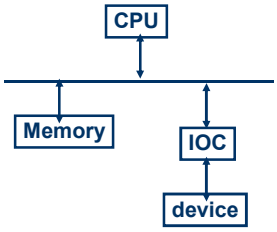
1. Polling:

- O dispositivo de I/O coloca informações em um registrador de status;
- O sistema operacional verifica periodicamente o status do registrador.

2. I/O Interrupt:

- Quando o dispositivo de I/O precisar da atenção do processador, ele irá gerar um sinal de interrupção (Interrupt – IRQ) para o processador.
- O sistema operacional deve prover uma rotina que irá atender esta solicitação de interrupção (Interrupt Handler)

I/O - E/S por Polling

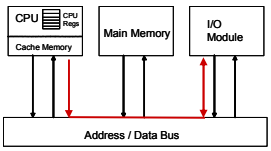


busy wait loop is an inefficient way to use the CPU unless the device is very fast!

but checks for I/O completion can be dispersed among computation intensive code

- Vantagem
  - Simples: o processador está controlando o I/O e faz todo o trabalho.
- Desvantagem:
  - Sobrecarga do Polling que consome muito tempo de CPU.

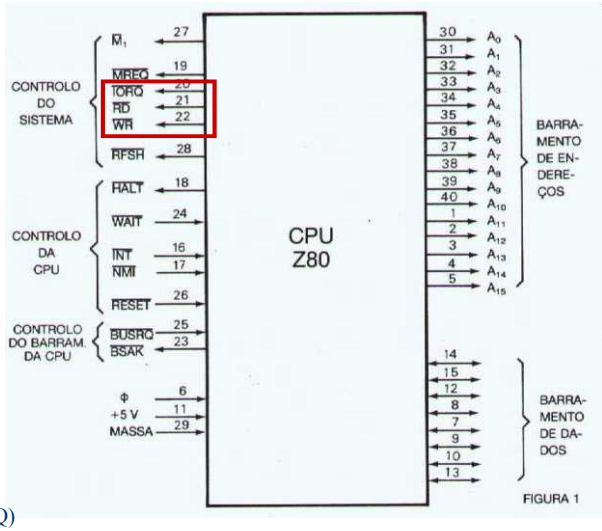
I/O - E/S por por Polling



Porta de I/O:

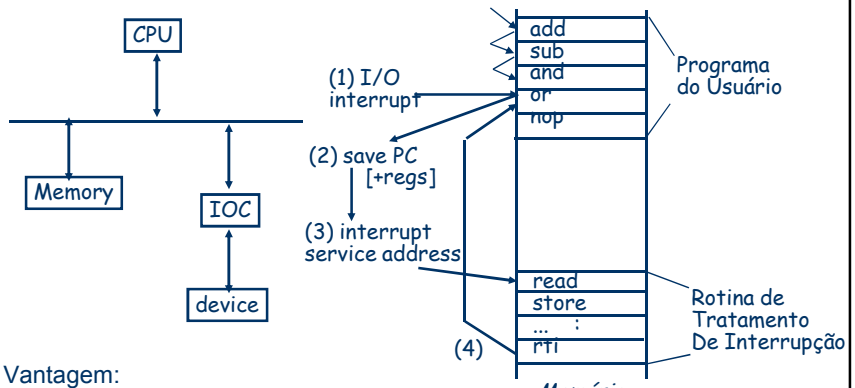
-Acessa um endereço de I/O

-Status:
Dado disponível
Dado Indisponível



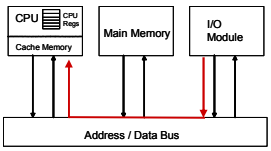
Controles: I/O Request (IORQ)

I/O - E/S por por Interrupção



- Vantagem:
  - O programa do usuário só é parado no momento em que a transferência realmente ocorre.
- Desvantagem:
  - Necessidade de um hardware especial para gerenciar os eventos:
  - Gerar a interrupção (I/O device) / Detectar a interrupção (CPU)

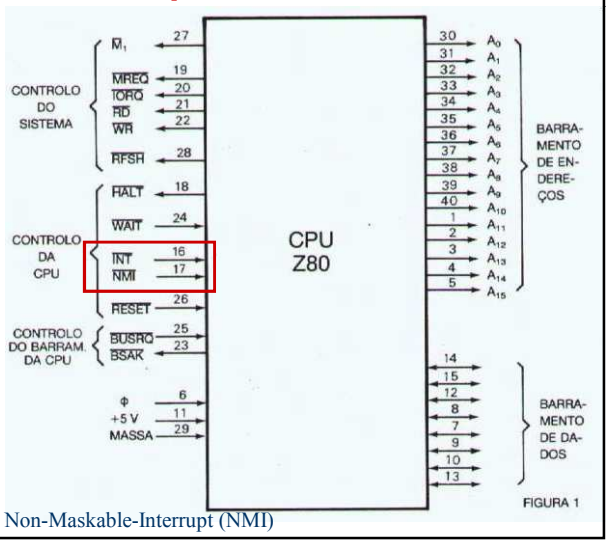
I/O - E/S por por Interrupção



Interrupt (Int):

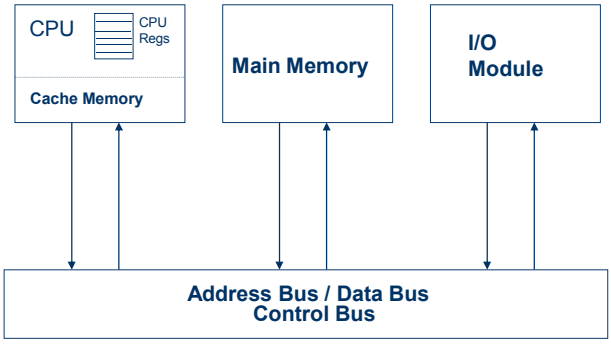
-Interromper a CPU avisando que tem evento sinalizando I/O

-Int: Pegue o dado, Próximo dado.



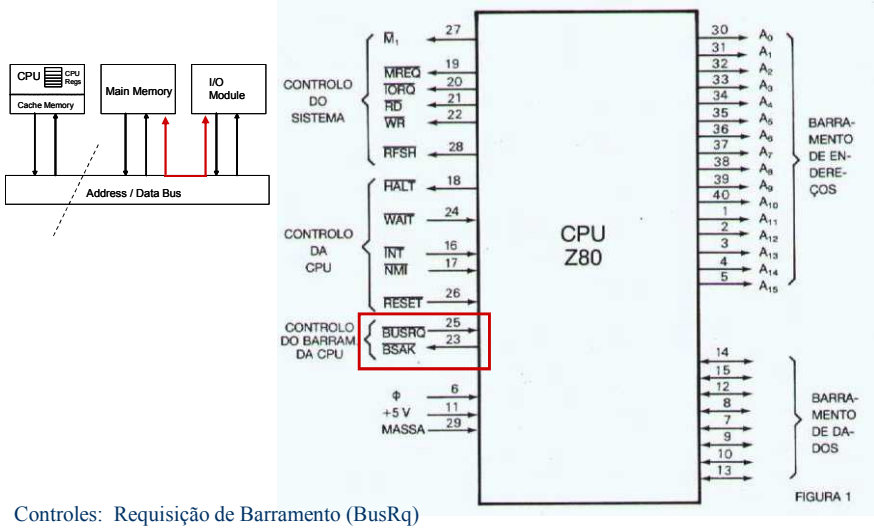
Controles: Interruption (Int) / Non-Maskable-Interrupt (NMI)

I/O - Barramento e DMA



DMA: De um modo geral o DMA consiste na realização de uma transferência de dados entre um determinado dispositivo de I/O e a memória principal, praticamente sem intervenção da CPU.
Controle: DMA controller => Requisição de Barramento (BusRq)

I/O - Barramento e DMA



Controles: Requisição de Barramento (BusRq)

INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO
Web institucional: http://www.icmc.usp.br/ssc/
Página pessoal: http://www.icmc.usp.br/~fosorio/
E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

Disciplina de Arquitetura de Computadores / Informática
Estagiário PAE: Maurício A. Dias
Web disciplina: COTEIA - Http://coteia.icmc.usp.br
> Programa, Material de Aulas, Critérios de Avaliação,
> Lista de Exercícios, Trabalhos Práticos, Datas das Provas

Visite regularmente o site da CoTeia - Material em permanente atualização