

**USP - ICMC - SSC  
SSC 0610 - Eng. Comp. - 2o. Semestre 2010**

## **Disciplina de Organização de Computadores I**

**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: [maccddias \[at\] gmail.com](mailto:maccddias@gmail.com)**

**Material on-line Wiki ICMC - <http://wiki.icmc.usp.br/index.php/Ssc-610>**

*Aula 08s*

## **Agenda - Tópicos Abordados**

### **Conteúdos Abordados:**

- 1. CPU: Entrada e Saída**
- 2. Tipos de E/S**
  - E/S por Pooling
  - Interrupção
  - DMA

### 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 – Instruções tipo LDA, STA
  - Instruções especiais de I/O  
CPU Instruction Set: InPort, OutPort

### I/O – Tipos de E/S: Comandos

- Dois métodos são usados para acessar os dispositivos de I/O]

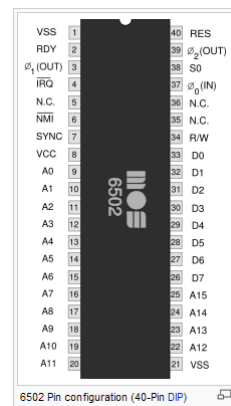
- I/O mapeado em memória  
Acesso a memória de I/O  
Instruções tipo LDA, STA

*Instruction Set:*

Sem Instruções Específicas para I/O

Exemplo: 6502

- > Não possui instruções especiais de I/O
- > Não possui hardware específico para I/O



**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  
 Instruções tipo LDA, STA

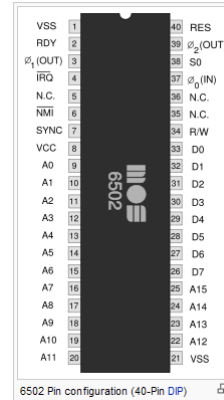
Exemplos:

APPLE – Acesso ao “Speaker”  
 End. \$C030

Simulador 6502

Acesso ao Vídeo: \$E000,\$E002,\$E003

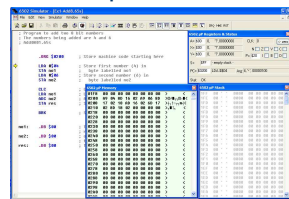
Acesso ao Teclado: \$E004



**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  
 Instruções tipo LDA, STA



\$E000 Clear terminal window

\$E001 Ascii code will be sent to terminal screen and displayed as a character

\$E002 Ascii code will be sent to screen and displayed as above, but CR/LF will be ignored

\$E003 Ascii code will be sent to screen and displayed as a hexadecimal number

\$E005 Controls horizontal location of cursor

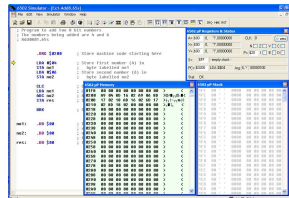
\$E006 Controls vertical location of cursor.

## Entrada e Saída

### 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  
 Instruções tipo LDA, STA



\$E004 If a character has been typed into the terminal screen, it will be copied to this location

Teclado...

ASCII, echo, Press-Release, Leitura múltipla, Scan-Codes,

```

inicio:  LDA #$00
        STA $E000 ; Limpa Tela
Loop:   LDA $E004
        CMP #00 ; Espera tecla
        BEQ Loop
        STA $E002 ; Exibe Echo
Pula:   CMP #$0D ; Enter: FIM
        BNE Loop
        BRK

; Leitura de teclado por "Pooling"
    
```

## Entrada e Saída

### I/O – Tipos de E/S: Comandos

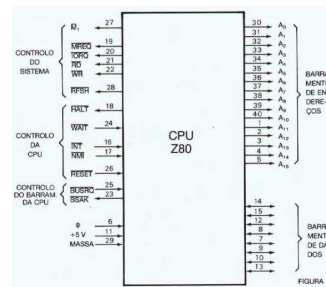
- Dois métodos são usados para acessar os dispositivos de I/O]

- I/O mapeado em memória  
 Acesso a memória de I/O – Instruções tipo LDA, STA

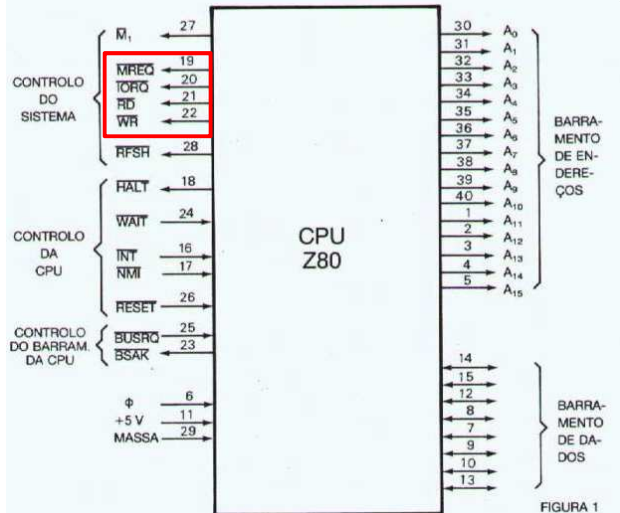
- Instruções especiais de I/O  
 CPU Instruction Set: InPort, OutPort

Exemplo de Instruções:

Z80 => IN A,(n) [Porta n]  
 OUT (n),A  
 8086 => IN data, port  
 OUT port, data



### I/O – Tipos de E/S: Comandos



**Z80:** In the instructions IN A,n and OUT n, A, the I/O device address n appears in the lower half of the address bus (A7-A0)

#### Pooling Problems:

- Sync (síncrono)
- Overrun (perda de dados)
- CPU Overhead (alto custo computacional)

### 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: (ex. Pooling)

- 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:

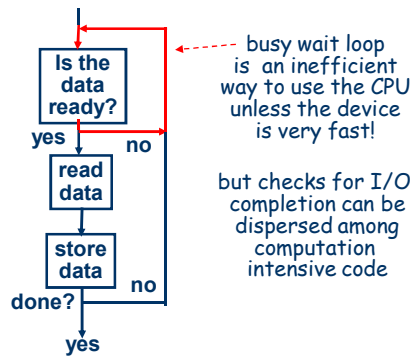
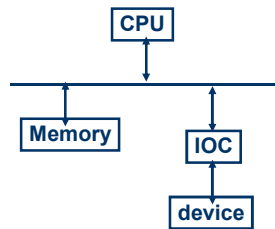
- O dispositivo de I/O coloca informações em um registrador de status;
- O sistema operacional verifica periodicamente o status do registrador.

#### 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)

## Entrada e Saída

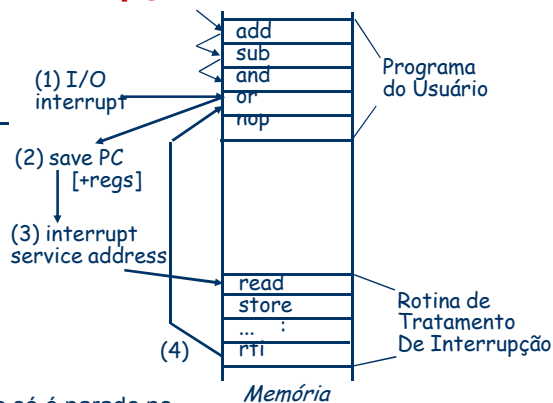
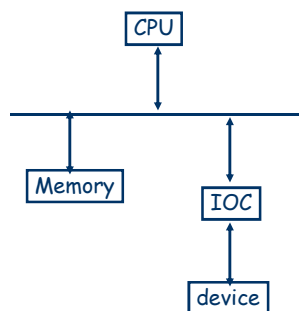
### I/O - E/S por Polling



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

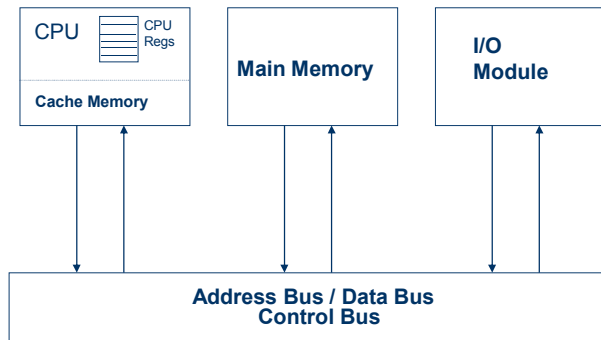
## Entrada e Saída

### I/O - E/S por Interrupção



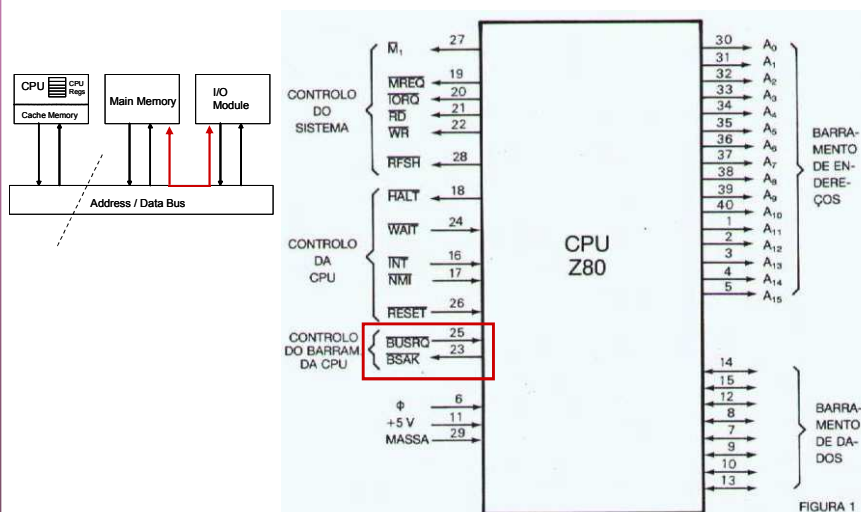
- Vantagem:
  - O programa do usuário só é parado no momento em que a transferência realmente ocorre.
- Desvantagem:
  - Necessidade de um hardware especial para gerenciar os eventos:
  - Gerar a interrupção (I/O device) / Detectar a interrupção (CPU)

**I/O - 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)

FIGURA 1





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](mailto:fosorio@icmc.usp.br) ou [fosorio \[at\] gmail. com](mailto:fosorio@gmail.com)**

**Disciplina de Organização de Computadores I / Eng. Comp.**

**Estagiário PAE: Maurício A. Dias**

**Web disciplina: <http://wiki.icmc.usp.br/index.php/Ssc-610>**

**> Programa, Material de Aulas, Critérios de Avaliação,**

**> Lista de Exercícios, Trabalhos Práticos, Datas das Provas**