

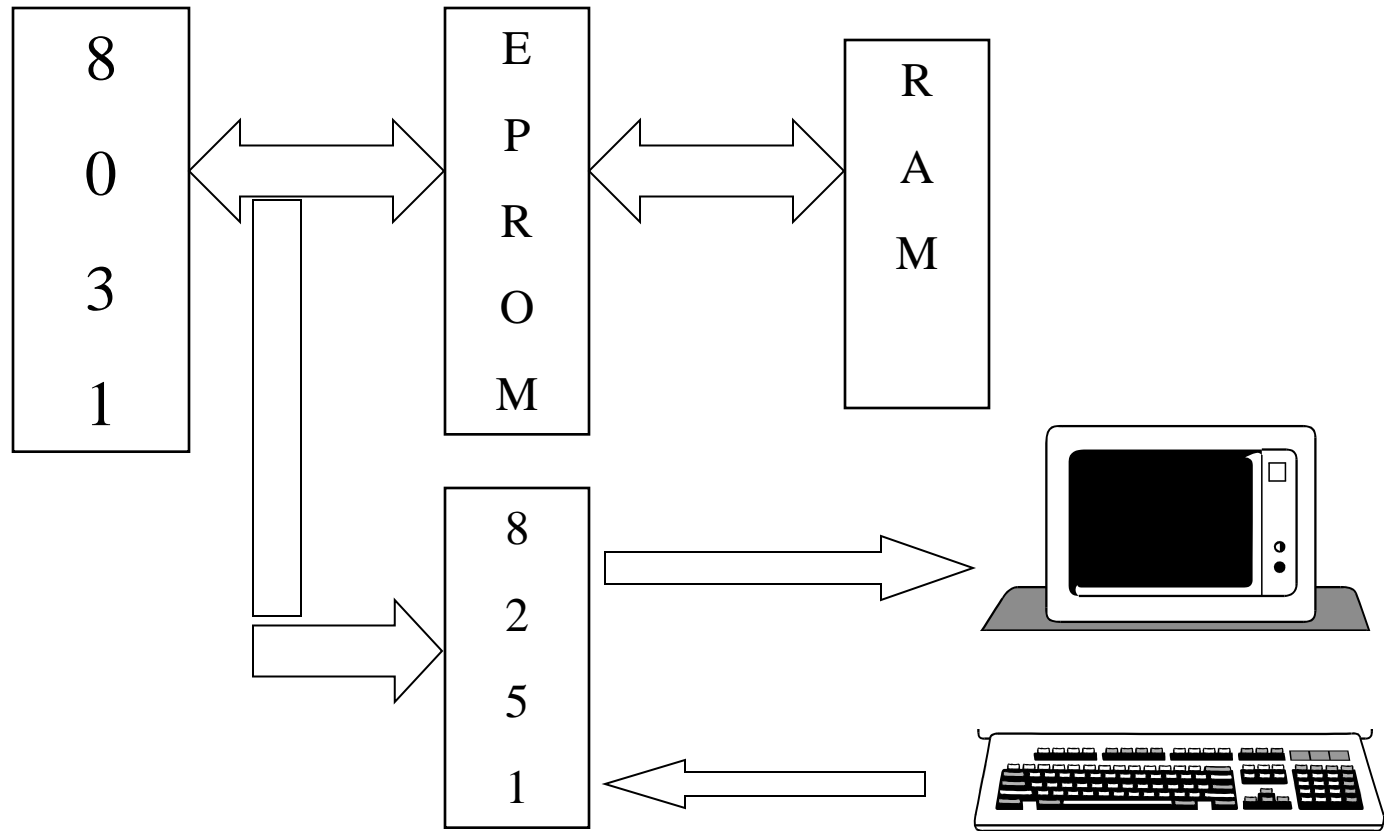
Universidade de São Paulo

Organização de Computadores

Dr. Jorge Luiz e Silva

Cap 7

Entrada e Saída



Acesso à E/S

```
MOV [1000H],AX
```

```
MOV AX,[1000H] → Memória
```

Instruções de Entrada e Saída

```
IN AL,END
```

```
IN AL,DX
```

```
OUT END,AL
```

```
OUT DX,AL
```

Entrada e Saída - Sincronismo

Tempo de execução `MOV AX,[1000H]` = t

Tempo de execução `IN AL,DX` >> t

Informações são externas ao sistema

Executar a instrução `IN AL,DX` o dado pode não estar pronto \Rightarrow Sincronismo do sistema para acessar os dados externos.

END (como fica a questão do endereço ou DX)

Entrada e Saída - Sincronismo - END

DX – 16 bits \Rightarrow 65535 endereços possíveis

porto – 8 bits \Rightarrow 256 endereços diferentes

SINCRONISMO – IN AL,DX

se dado não pronto \Rightarrow esperar o dado ficar pronto
(disponível) só então utilizá-lo

8251 – 2 registradores (2 buffers E/s e 1 flag)

E = 0 escrita não pronta

E = 1 escrita pronta

L = 0 leitura não pronta

L = 1 Leitura pronta

Entrada e Saída - Porto

IN AL,DX onde DX endereço do status

Em AL verificar se está pronto para ler ou escrever, então endereçar o dado a se lido ou escrito.

O conjunto STATUS, BUFFER de DADOS E/S é chamado PORTO de Entrada e Saída.

Entrada e Saída - dados

O que enviar pela E/S?

Trabalhar com dados de 16 bits e temos normalmente que Multiplicar, Somar, Dividir, etc.

Os dados trocados entre microprocessadores e qualquer dispositivo periférico são codificados em ASCII (American Standar Code for Information Interchange)

Ex: A – 01000001b

 B – 01000010b

 7 – 00110111b

 9 – 00111001b

Código ASCII é entendido pelo 8251 e entendido pelo terminal de vídeo (8251).

Entrada e Saída - Linha Serial

Dados com 7 bits que são enviados ou recebidos pelo 8088 através de um FIO \Rightarrow O dado é enviado ou recebido em série no protocolo “RS 232” ou outro qualquer.

8251 monta o RS232 e o envia \Rightarrow terminal que o recebe desmonta o RS232 e pega o código ASCII fazendo em seguida o teste de paridade. Qualquer anormalidade é sinalizado no flag do 8251.



11 bits de dados

Entrada e Saída - Exemplo 1

Endereço para dados FAH - Endereço para status FBH

Ler um dado do terminal

Verificar o status – se pronto ler o dado senão ficar esperando até que pronto

```
T1:  MOV  DX,0FBH
      IN   AL,DX
      TEST AL,2
      JZ   T1
      IN   AL,0FAH
      MOV  [1000H],AL
```

O dado em ASCII está na posição 1000h da memória

Entrada e Saída - Exemplo 2

Limpendo o bit de paridade

```
T1:  MOV  DX,0FBF
      IN   AL,DX
      TEST AL,2
      JZ   T1
      IN   AL,0FAH
      AND  AL,7FH
      MOV  [1000H],AL
```

Entrada e Saída - Exemplo 3

Escrever um dado no terminal

Verificar se o status de escrita está disponível.

Se pronto então escrever senão esperar até que pronto

```
T1:      MOV      DX,0FBH
         IN       AL,DX
         TEST    AL,1
         JZ     T1
         MOV    AL,[1000H]
         MOV    DX,0FAH
         OUT   DX,AL
```

Entrada e Saída - Exemplo 4

Escrever um programa ECO supondo o sistema acima

```
T1:      MOV  DX,0FBH
         IN   AL,DX
         TEST AL,2
         JZ   T1
         IN   AL,0FAH
         AND  AL,7FH
         MOV  [1000H],AL
```

Entrada e Saída - Exemplo 4

cont.

```
T2:      MOV  DX, 0FBH
         IN   AL,DX
         TEST AL,1
         JZ   T2
         MOV  AL,[1000H]
         MOV  DX,0FAH
         OUT  DX,AL
         JMP  T1
```

Entrada e Saída

Interrupções do DOS

INT 21H

Parâmetros: passados em AH

AH - 1 leitura com ECO, resultado em AL

AH - 2 escrita de dados de DL

AH - 9 escrita de string endereçada em DX

AH - 4CH volta para DOS

INT 10H

Parâmetros: AX

AX - 2 apaga tela

Entrada e Saída

Interrupções do DOS

Exemplo 1

Fazer um programa que leia um conjunto de dados do terminal e os coloque na Memória.

Elaborar o programa em classe usando TASM, TLINK