

Entrada e Saída

Prof. Alexandre Beletti

Cap. 3 – Stallings, Cap. 10 – Monteiro,
Cap. 2 – Tanenbaum, Cap. 15 - Weber

Tópicos de E/S

- Introdução
- Tipos de E/S
 - E/S por Programa (Programada)
 - Interrupção
 - DMA

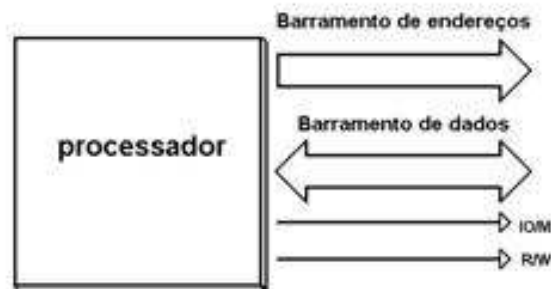
Dispositivos de E/S

- Entrada:
 - Teclado
 - Mouse
 - Scanner
- Saída:
 - Monitor
 - Impressora

Interfaces de Comunicação

Dispositivo	Interface
Monitor	Placa de vídeo
Teclado	Interface de teclado
Alto falante	Interface de alto falante
Impressora	Interface paralela ou USB
Mouse	Interface serial, PS/2 ou USB
Drive de disquete	Interface para drives de disquete
Disco rígido IDE	Interface IDE
Disco rígido SCSI	Interface SCSI
Joystick	Interface para jogos ou USB
Scanner	Placa de interface de scanner, paralela ou USB
Câmera digital	Interface serial, paralela ou USB
ZIP Drive	Existem modelos SCSI, paralelos, USB e IDE.

Barramentos e Sinais de Controle



Barramento de Endereços

- Address Bus
- Nos 386DX e 486 é um conjunto de 32 sinais digitais, representados por 32 terminais do processador
- Especifica o endereço de memória ou E/S que se deseja ter acesso
- É do tipo Unidirecional (trafegam em uma única direção = seta da figura anterior)

Barramento de Dados

- Data Bus
- Nos 386DX e 486 este barramento possui 32 bits (32 terminais do processador)
- No Pentium possui 64 bits
- Através desse barramento trafegam os dados que o processador lê e escreve na memória e nas interfaces
- É do tipo Bidirecional (figura anterior)

I/O

- Input Output Memory
- Com esse sinal o processador avisa se está acessando:
 - Posição de Memória
 - Posição de E/S

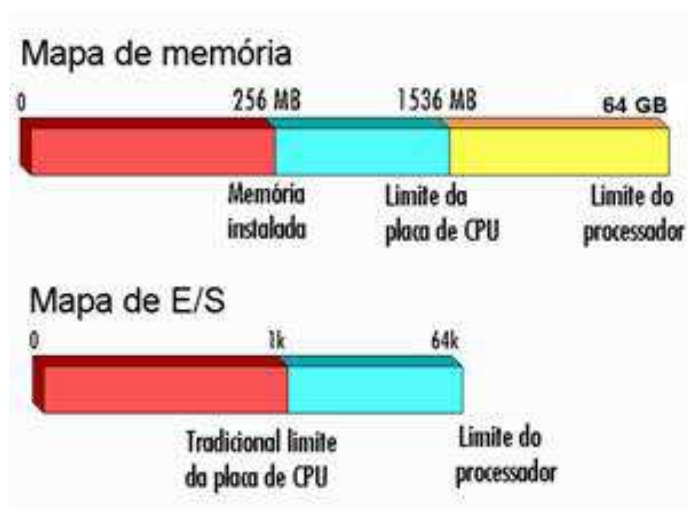
R/W

- Read / Write
- Com esse sinal o processador indica se está realizando uma operação de leitura ou de escrita

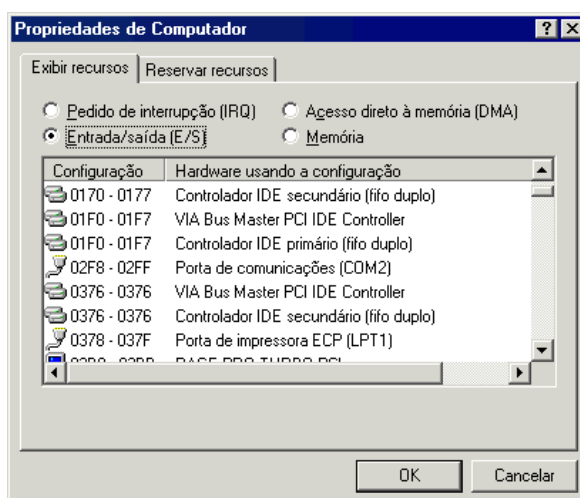
Exemplos

- Exemplo de Leitura de E/S
 - Recepção de um tecla pressionada no teclado
- Exemplo de Escrita de E/S
 - Transmissão de um caractere para a impressora

Mapa de Memória PC – 256Mb



Mapa de E/S



Endereços Mapa de E/S

Endereços	Interface que os utiliza
000-01F	Controlador de DMA (placa de CPU)
020-03F	Controlador de interrupções (placa de CPU)
040-05F	Timer (placa de CPU)
060-06F	Controlador de teclado do AT
070-07F	Chip CMOS
080-09F	Registro de página de DMA (placa de CPU)
0A0-0BF	Segundo controlador de interrupções (CPU)
0C0-0DF	Segundo controlador de DMA (placa de CPU)
0F0-0F1	CLEAR e RESET do coprocessador
170-177	Controladora IDE secundária
1F0-1F7	Controladora IDE primária
200-207	Interface de joystick
278-27F	Porta paralela
2E8-2EF	Porta serial COM4
2F8-2FF	Porta serial COM2
370-377	Interface de drives secundária
378-37F	Porta paralela
3B0-3BF	Placa de vídeo MDA e HERCULES
3C0-3CF	Placa VGA
3D0-3DF	Placas CGA e VGA
3E8-3EF	Porta serial COM3
3F0-3F7	Interface de drives primária
3F8-3FF	Porta serial COM1

Métodos de Comunicação E/S

- Entrada/Saída por programa
- Entrada/Saída via interrupção
- Acesso Direto à Memória DMA

E/S por Programa

E/S por Programa

- Processador é utilizado intensamente para a realização de uma operação de E/S
- “Perde tempo” questionando se um determinado dispositivo está pronto ou não para uma nova operação

Processo

- Processador testa o estado (status) do dispositivo (Ex: HD) – INTERROGAÇÃO ou POLLING:
 - Dispositivo Pronto ou;
 - Ocupado em outra operação (BUSY ou WAIT)
- Pode ser realizado via registrador específico
- Dispositivo Pronto = Processador comanda operação de leitura ou escrita

Mecanismo

- UCP envia o endereço do dispositivo e da sua interface e o comando de E/S desejado
- Leitura: interface interpreta o comando e busca um byte ou palavra do registrador interno do dispositivo e coloca em seu registrador
- UCP necessita posteriormente receber o dado, solicitando que a interface o coloque no barramento

Desvantagem

- Uso intenso do processador (quase uma “espera ativa”)
- Loop de Interrogação
- Velocidade: Periférico X UCP

Como a UCP comunica-se com disp. E/S

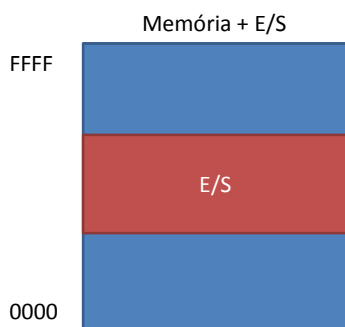
- Uma instrução de entrada (INPUT). Para transferir uma palavra de um dispositivo de E/S para a UCP
- Uma instrução de saída (STORE) para transferir uma palavra da UCP para um dispositivo de E/S

Organizando a Comunicação

- Organizar a comunicação entre:
 - Processador
 - Memória Principal
 - Interface de E/S
- Possibilidades:
 - Memória compartilhada (memory-mapped)
 - E/S isolada (isolated I/O)

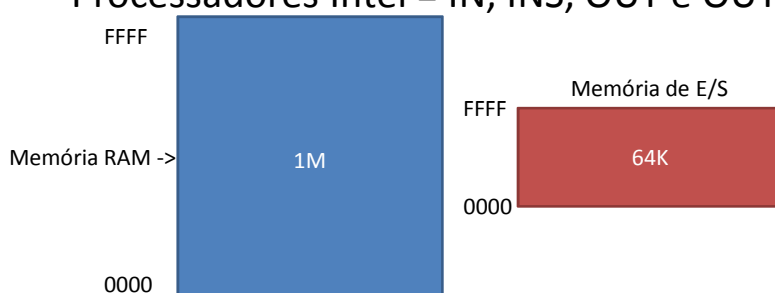
Memória Compartilhada

- Memória compartilhada por:
 - Instruções e dados de programas e;
 - Instruções e dados de E/S.



Memória Isolada E/S (isolated I/O)

- Um espaço e memória próprio
- Está fora da memória principal
- Vantagem: não utiliza a RAM
- Desvantagem: instruções específicas (Ex: Processadores Intel = IN, INS, OUT e OUTS)



Intel 8080

- Faz uso de duas instruções:
- IN x: transfere uma palavra da porta de E/S 'x' para o acumulador do 8080
- OUT x: transfere uma palavra do acumulador do 8080 para a porta de E/S 'x'

Consulta antes da comunicação

- Atenção! Somente enviar um dado não garante que o dispositivo esteja pronto para recebê-lo
- O ideal é testar as condições de recebimento e/ou envio de informações
- Geralmente temos um bit especial para esse propósito, como podemos ver nos passos do próximo slide...

Passos de E/S Programada

1. Lê a informação de estado
2. Testa o estado para determinar se o dispositivo está pronto para iniciar a transferência de dados
3. Se não está pronto, retorna ao passo 1; se está pronto efetua a transferência do dado

Exemplo 2 – Intel 8080

- Considere a transferência de uma palavra de dados de um dispositivo de E/S para o acumulador na UCP
- Considere um dispositivo ligado às portas 1 e 2
- O estado do dispositivo está continuamente na porta 1
- Os dados solicitados estão na porta 2....

Trecho de Programa de E/S

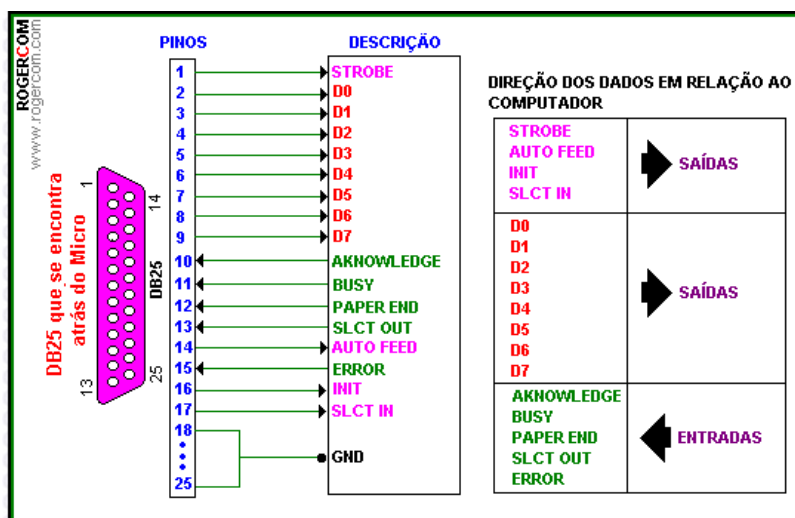
instrução	comentário
WAIT: IN 1	Lê estado do dispositivo de E/S, localizado na porta 1, para o acumulador
CMP READY	Compara a palavra READY (de forma imediata) com o acumulador; se forem iguais, liga Z=1, senão, Z=0
JNZ WAIT	Se Z ≠ 1 (dispositivo de E/S não está pronto), volta para a instrução de rótulo WAIT
IN 2	Lê palavra de dados para o acumulador a partir da porta 2

Exemplo – x86 – Porta Paralela

Nome da Porta	Endereço de memória	Endereço da Porta	Descrição
LPT1	0000:0408	378 hexadecimal	Endereço base
LPT2	0000:040A	278 hexadecimal	Endereço base

Nome	Endereços LPT1	Endereços LPT2	Descrição
Registro de Dados	378h	278h	Envia um byte para a impressora
Registro de Status	379h	279h	Ler o Status da impressora
Registro de Controle	37Ah	27Ah	Envia dados de controle para a impressora

Porta Paralela – DB25



Exemplo 1 - Código em C

```
// rogercom.com
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#define LPT1 0x378
int main(void)
{
    unsigned char Valor=128; //Em binário: 10000000
    while( Valor > 0 )
    {
        outportb(LPT1, Valor); // Envia para a Porta LPT1
        printf("\nPressione uma tecla para ascender o próximo LED...");
        getch( );
        Valor = Valor >> 1; //A cada passagem, o bit 1 é movido para a
        direita
    }
}
```

Exemplo 1 – Código em Assembly

- MOV AL, 55H
- MOV DX, 378H
- OUT DX, AL

Interrupção

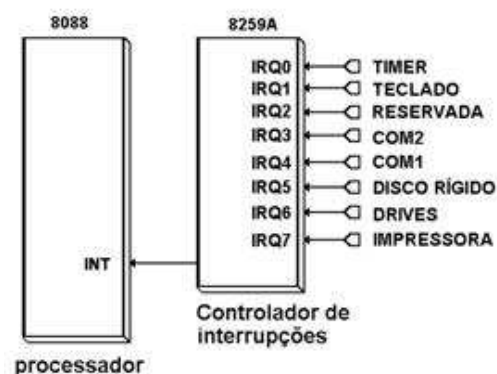
Interrupção

- Processador emite a interrupção de E/S para a interface (processador fica livre);
- Quando a interface estiver pronta para enviar os dados, ela “avisa” ao processador através de um sinal de interrupção;
- O processador inicia então o “programa” de E/S como no caso anterior

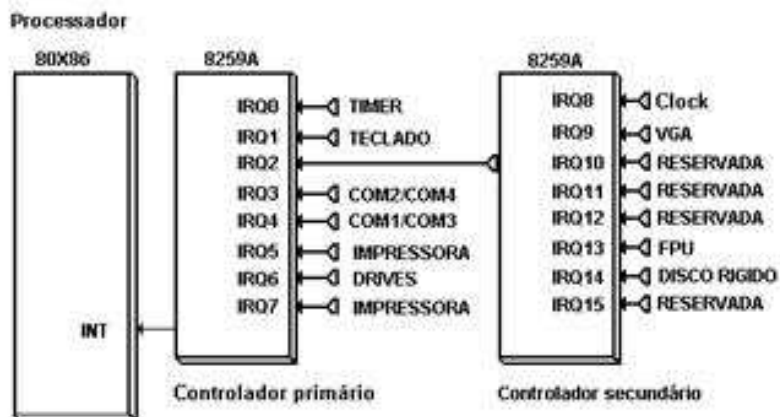
Detalhes da Interrupção

- Série de procedimentos que suspendem o funcionamento corrente do processador desvio sua atenção para outra atividade
- Quando a outra atividade é concluída, o processador retorna à execução anterior do ponto onde estava antes de ser interrompida

Controlador de INT - 8088

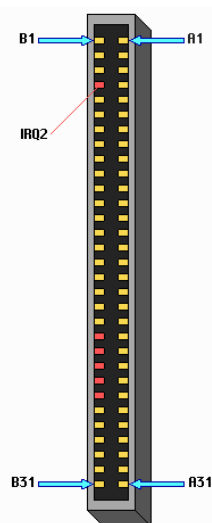


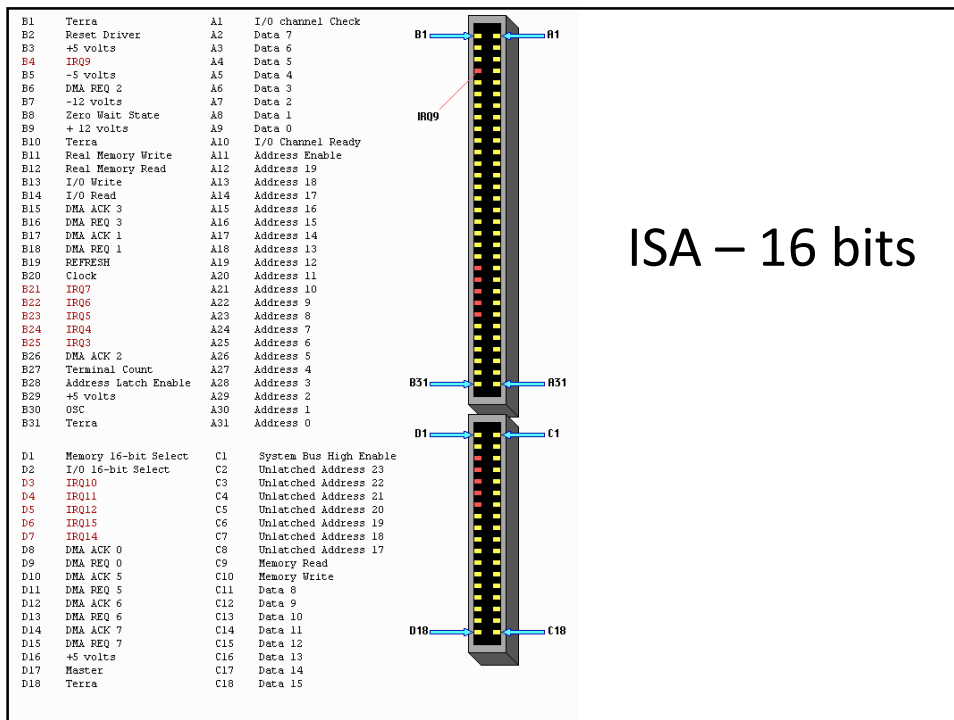
Controlador de INT - 80286



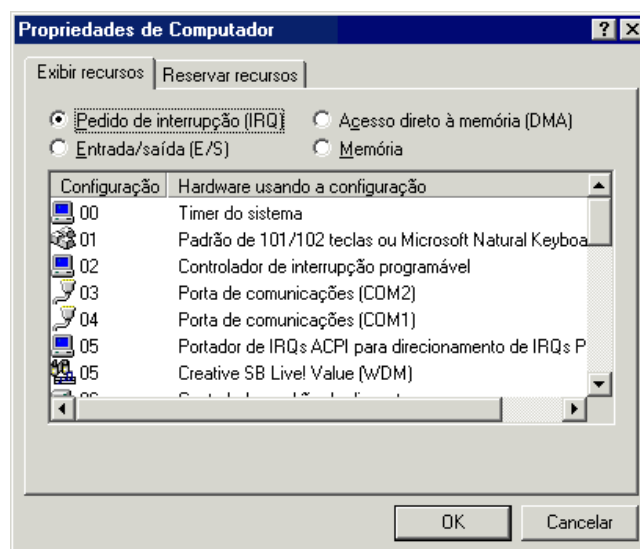
ISA – 8 bits

B1	Terra	A1	I/O channel Check
B2	Reset Driver	A2	Data 7
B3	+5 volts	A3	Data 6
B4	IRQ2	A4	Data 5
B5	-5 volts	A5	Data 4
B6	DMA REQ 2	A6	Data 3
B7	-12 volts	A7	Data 2
B8	Card select	A8	Data 1
B9	+ 12 volts	A9	Data 0
B10	Terra	A10	I/O Channel Ready
B11	Memory Write	A11	Address Enable
B12	Memory Read	A12	Address 19
B13	I/O Write	A13	Address 18
B14	I/O Read	A14	Address 17
B15	DMA ACK 3	A15	Address 16
B16	DMA REQ 3	A16	Address 15
B17	DMA ACK 1	A17	Address 14
B18	DMA REQ 1	A18	Address 13
B19	DMA ACK 0	A19	Address 12
B20	Clock	A20	Address 11
B21	IRQ7	A21	Address 10
B22	IRQ6	A22	Address 9
B23	IRQ5	A23	Address 8
B24	IRQ4	A24	Address 7
B25	IRQ3	A25	Address 6
B26	DMA ACK 2	A26	Address 5
B27	Terminal Count	A27	Address 4
B28	Address Latch Enable	A28	Address 3
B29	+5 volts	A29	Address 2
B30	OSC	A30	Address 1
B31	Terra	A31	Address 0





INT – Windows 9x/XP



Classes de Interrupção

- Interrupções Internas ou de Programas (chamadas de *traps* ou *exception*);
- Interrupções Externas.

Interrupções Internas

- Ocorrem devido a algum tipo de evento gerado pela execução de uma instrução ou até mesmo programado
- Exemplos: divisão por zero, um overflow em uma operação aritmética, um código de operação inválido.

Interrupções Externas

- Caudas por um sinal externo ao processador que o interrompe
- Em geral relacionado a interface de E/S que pretende “avisar” ao processador que um determinado periférico deseja atenção para transferir dados

INT - Ações do Processador

- Identificação do tipo de interrupção (de que se trata?)
- Qual o dispositivo que sinalizou?
- Como reagir à ocorrência?
- Dar atenção imediata? Aguardar para mais tarde? Ignorar?
- O que acontece com o programa interrompido? Quando ele retornará à sua execução?



Rotina de Tratamento de INT

- Para atender as questões anteriores
- Executa uma rotina de tratamento
- *Interruption Handler*
- Antes de ocorrer isso o processador executa o “Salvamento de Contexto” do programa corrente
- São salvos registradores, o contador de instrução (PC)

Salvamento de Contexto

- Faz uso de uma área da RAM
- Salvar registradores
- Salvar o PC ou CP (Contado de Programa)
- Modificar PC para o endereço da rotina de *Interruption Handler*

Desvantagem

- Continua gastando tempo para executar o programa de E/S para efetivar a transferência de dados
- *Programmable Peripheral Interface*
- Intel 82C55A

Exemplo 2 - Código em C

```
// rogercom.com
void PrintValor(unsigned char Valor);
{
    union REGS regs;
    regs.h.ah = 0; //Serviço de impressão
    regs.h.al = Valor;
    regs.x.dx = 0; //0 é a porta LPT1, 1,a LPT2 int86(0x17,& regs,&
regs); //Interrupção da impressora
}
//Ler o status da impressora unsigned char StatusPrint(void)
{
    union REGS regs;
    regs.h.ah = 2; //Serviço 2 ler o status regs.x.dx = 0; //0 é a LPT1
int86(0x17,& regs,& regs); //Interrupção da impressora
    return( regs.h.ah ); //Retorna um byte de status
}
```


Exemplo 1 – Código em Assembly

- MOV AH, 00H ; 00 = serviço de saída
- MOV AL, 30H ; 30 = '0'
- MOV AX, 000H ; 00 = LPT1
- INT 17h ; int da impressora

Acesso Direto á Memória - DMA

DMA

- Melhor alternativa para realizar operação E/S
- Transferência de dados entre uma determinada interface e a memória principal
- Praticamente sem intervenção do processador
- Solicita a transferência para um dispositivo denominado controlador de acesso direto à memória (*DMA controller*)

DMA Controller

- Realiza toda a transferência
- Processador fica livre para outros procedimentos
- Quando o controlador termina ele sinaliza o processador via interrupção
- Ele é responsável por controlar o barramento

DMA Controller

- Coloca um endereço no barramento de endereços para a memória principal (RAM)
- Ao mesmo tempo ele autoriza, via sinal de controle, que o periférico envie ou receba dados diretamente da memória principal
- Quando a operação de transferência de dados é concluída, o DMA retorna o controle do barramento para o processador

- Um controlador DMA é um dispositivo bastante complexo
- Possui diversos registradores internos e:
 - Registrador de dados;
 - Registrador de endereços;
 - Registrador que armazena um valor igual à quantidade de bytes ou palavras que são transferidas.

DMA – Windows 9x/XP

