

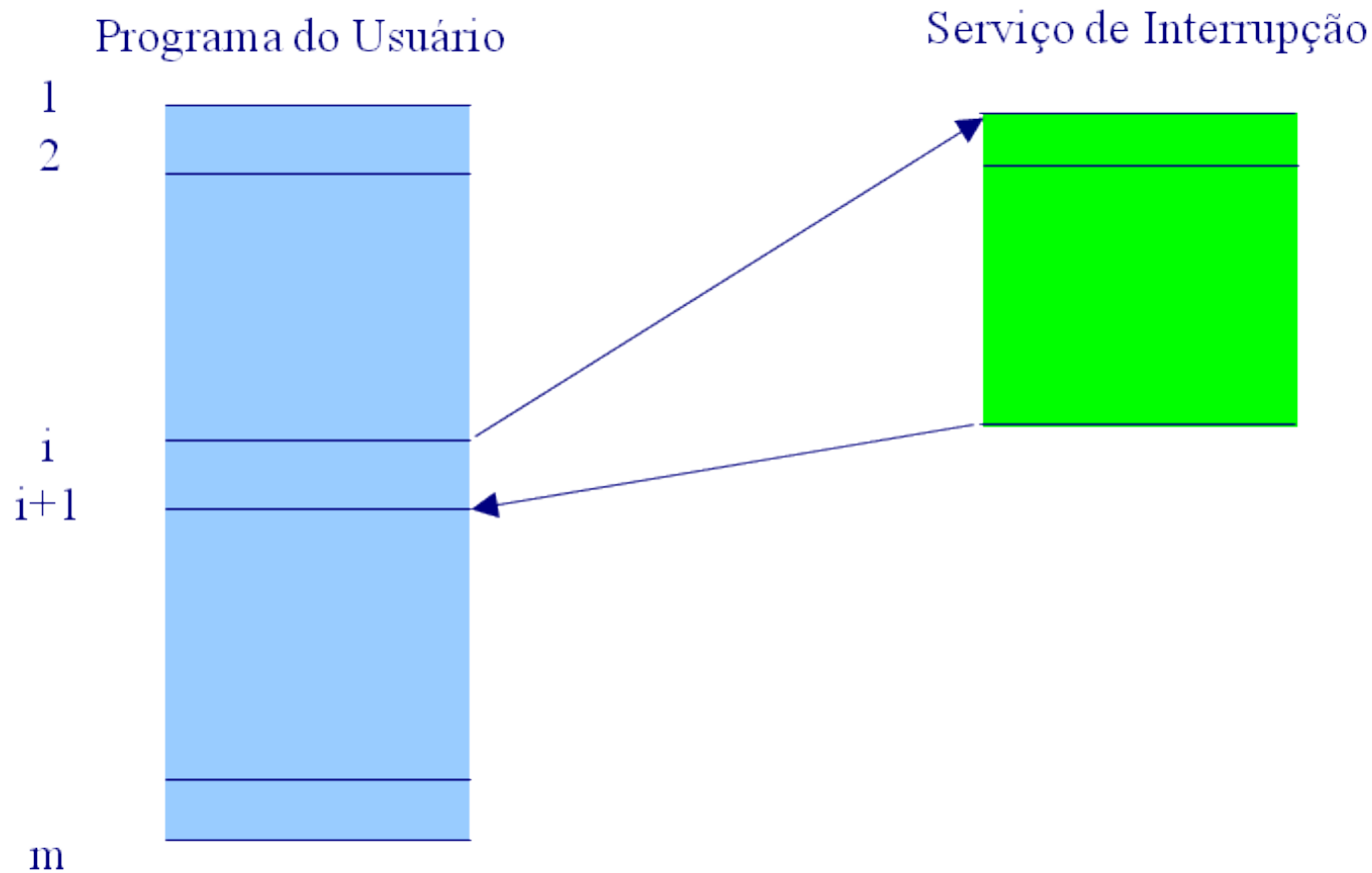
# Revisão: Modo Real, pt 2

## Interrupções

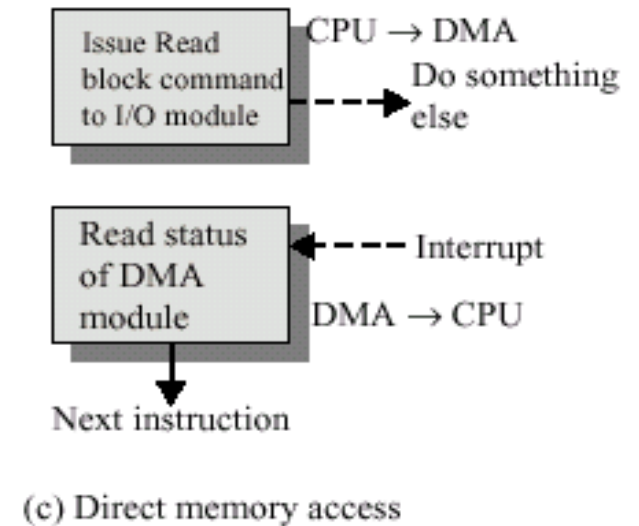
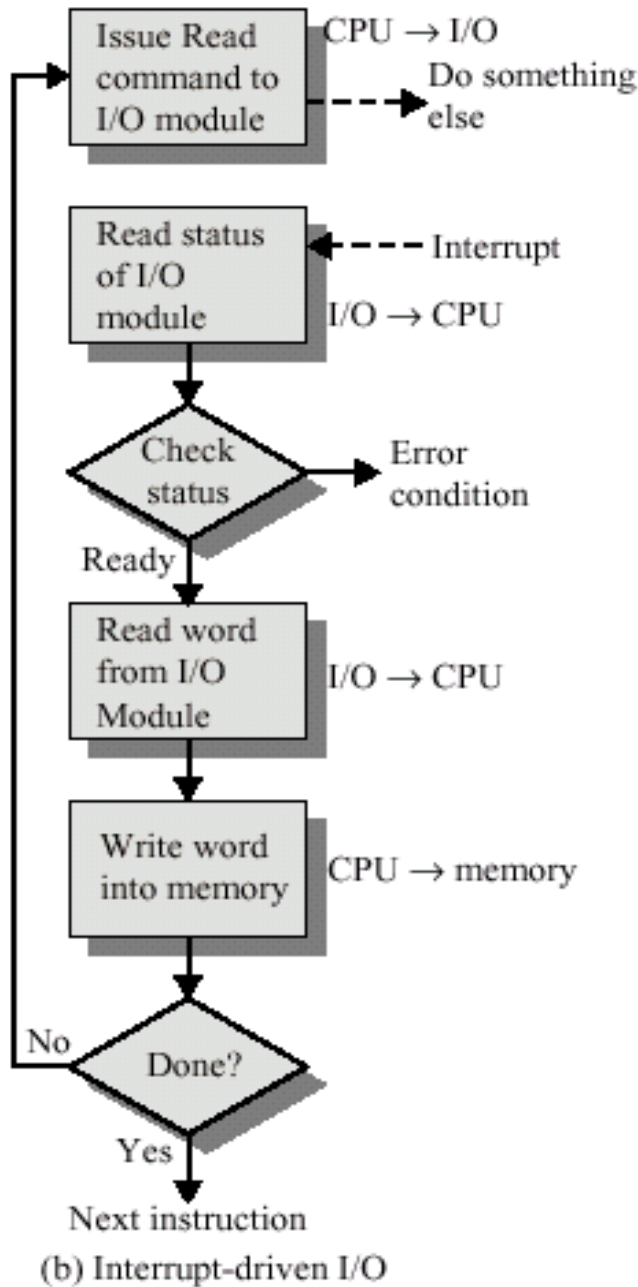
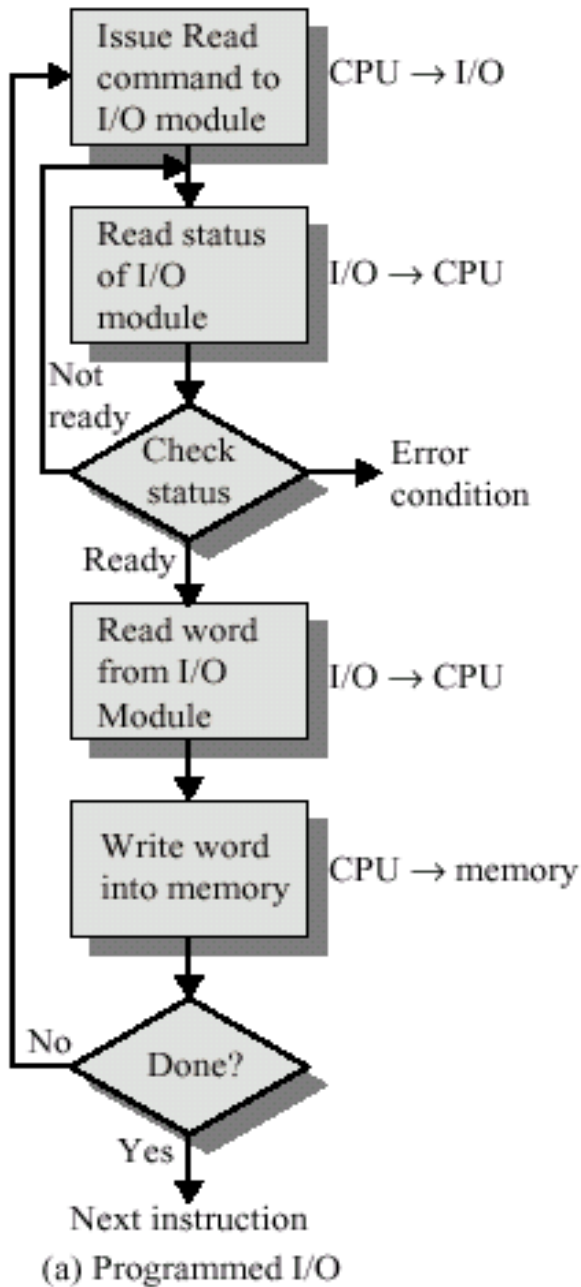
# Interrupção

Evento (interno ou externo) que faz o processador parar o programa em execução para tratar a interrupção

Ex: ler o teclado, comunicação com os dispositivos periféricos, utilizar os serviços de um S.O.



# Tratando a “hiperatividade” da CPU



# Tabela de Vetor de Interrupção

- Como o processador “encontra” o serviço de interrupção?
  - Toda interrupção gerada aponta para uma entrada em uma tabela de vetor de interrupção

# Tabela de Vetor de Interrupção

**No modo real**, coleção de endereços de 4 bytes (2 para IP seguido de 2 bytes para CS)

Apontador para rotina de serviço de interrupção

256 X 4 = 1.024 bytes  
(Faixa de endereços - 00000 até 003FFH)

Vetor	Descrição
0	Divide Error
1	Debugger Call (single-step)
2	NMI
3	Breakpoint
4	INTO
5	BOUND range exceeded
6	Invalid opcode
7	Device not available
8	Double fault
9	Reserved
10	Invalid task state segment
11	Segment not present
12	stack Exception
13	General protection
14	Page fault
15	Reserved
16	Floating-point error
17	Alignment check
18	Machine check
19-31	Reserved
32-255	Maskable Interrupts

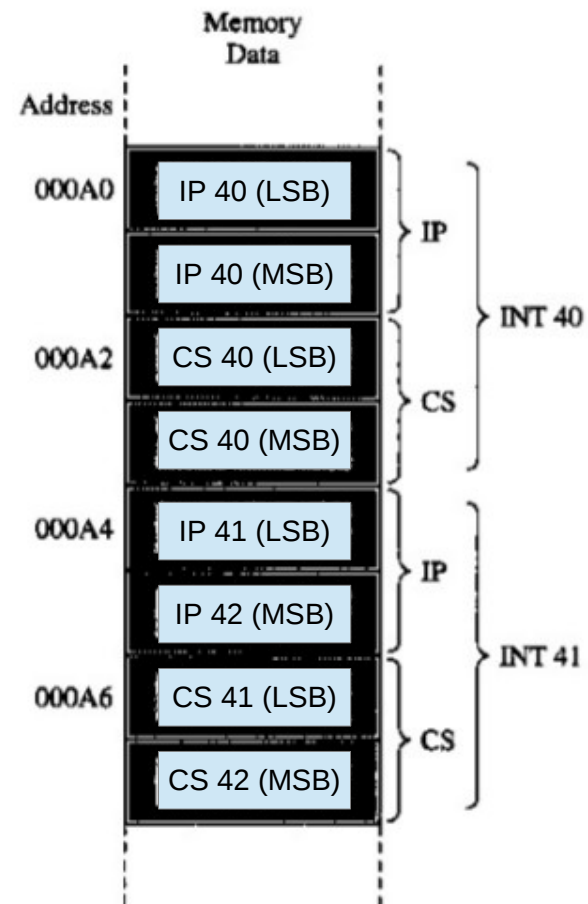
# Tabela de Vetor de Interrupção

- Usos comuns:

<b>INT (Hex)</b>	<b>IRQ</b>	<b>COMMON USES</b>
<b>00 - 01</b>	<b>Exeption Handlers</b>	-
<b>02</b>	<b>Non-Maskable IRQ</b>	<b>Non-Maskable IRQ (Parity Errors)</b>
<b>03 - 07</b>	<b>Exeption Handlers</b>	-
<b>08</b>	<b>Hardware IRQ0</b>	<b>System Timer</b>
<b>09</b>	<b>Hardware IRQ1</b>	<b>Keyboard</b>
<b>0A</b>	<b>Hardware IRQ2</b>	<b>Redirected</b>
<b>0B</b>	<b>Hardware IRQ3</b>	<b>Serial Comms. COM2/COM4</b>
<b>0C</b>	<b>Hardware IRQ4</b>	<b>Serial Comms. COM1/COM3</b>
<b>0D</b>	<b>Hardware IRQ5</b>	<b>Reserved / Sound Card</b>
<b>0E</b>	<b>Hardware IRQ6</b>	<b>Floppy Disk Controller</b>
<b>0F</b>	<b>Hardware IRQ7</b>	<b>Parallel Comms.</b>
<b>10 - 6F</b>	<b>Software Interrups</b>	-
<b>70</b>	<b>Hardware IRQ8</b>	<b>Real Time Clock</b>
<b>71</b>	<b>Hardware IRQ9</b>	<b>Redirected IRQ2</b>
<b>72</b>	<b>Hardware IRQ10</b>	<b>Reserved</b>
<b>73</b>	<b>Hardware IRQ11</b>	<b>Reserved</b>
<b>74</b>	<b>Hardware IRQ12</b>	<b>PS/2 Mouse</b>
<b>75</b>	<b>Hardware IRQ13</b>	<b>Math's Co-Processor</b>
<b>76</b>	<b>Hardware IRQ14</b>	<b>Hard Disk Drive</b>
<b>77</b>	<b>Hardware IRQ15</b>	<b>Reserved</b>
<b>78-FF</b>	<b>Software Interrups</b>	-

# Interrupção, encontrando o endereço na tabela...

- Como a tabela começa no endereço 00000h e cada entrada tem 4 bytes, basta multiplicar o número da interrupção por 4.
- Exemplo:
  - Interrupção 40, qual o endereço na tabela?
    - $40 \times 4 = 160 = 000A0h$

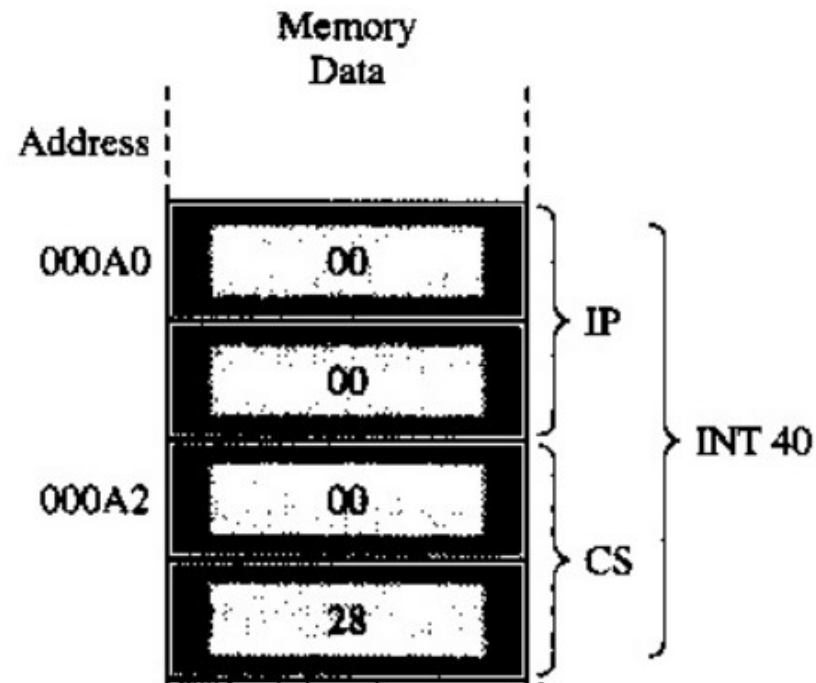


# Exemplo

- O programador quer que quando seja acionada a interrupção 40, o código na posição de memória 28000H seja executado. Como “configurar” a tabela de vetor de interrupção para isso?
  - Qual um possível CS:IP de 28000h?
    - CS = 2800h, IP = 0000h
      - Vamos então salvar esses valores na posição correta da tabela
        - Isto é, na posição de memória 000A0h
  - Estratégia: vamos apontar DS:DI para 000A0h e salvar nessa posição

# Exemplo: Solução

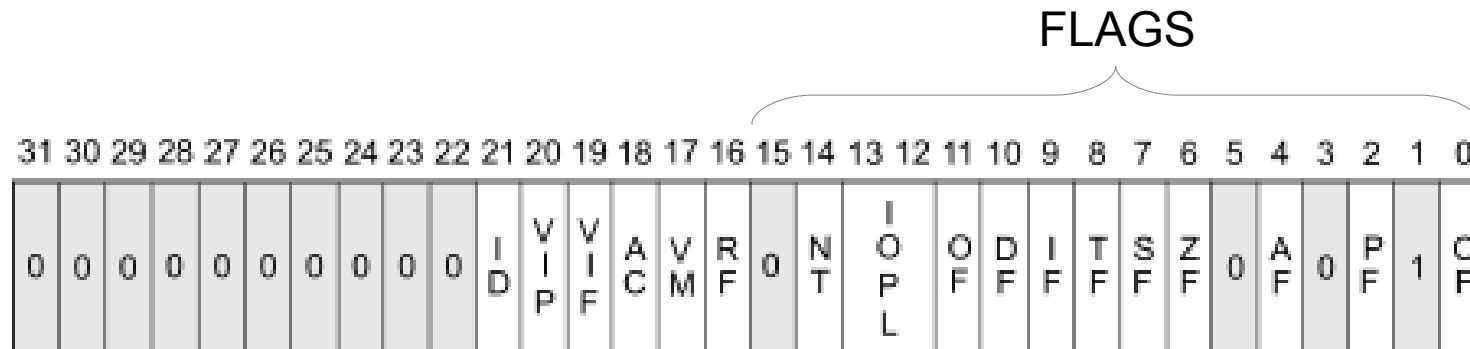
```
PUSH    DS                ;save current DS address
MOV     AX,0              ;set new DS address at 0000
MOV     DS,AX
MOV     DI,00A0H          ;offset for INT 40 vector
MOV     WORD PTR [DI],0   ;store IP address
MOV     WORD PTR [DI + 2],2800H ;store CS address
POP     DS                ;get old DS address back
```



# Interrupções: tipos

- Interrupções de hardware
  - Geradas por dispositivos conectados ao processador
  - Um x86 possui 2 pinos para interrupções de hardware
    - NMI: Nonmaskable interrupt
    - INTR: Interrupt request
  - Uma requisição de interrupção via o pino INTR pode ser ignorada, se o processador tiver sido instruído para ignorar interrupções
    - CLI : Clear interrupt-enable flag. Mascara interrupções INTR
    - STI: Set interuupt-enable flag. (Re)-Habilita interrupções INTR
  - Uma NMI não pode ser ignorada
- Interrupções de software
  - Geradas por instruções
    - Ex.: INT x, INTO
  - Geradas pelo próprio processador
    - Ex.: Divisão por zero

# Registrador (E)FLAGS



## EFLAGS

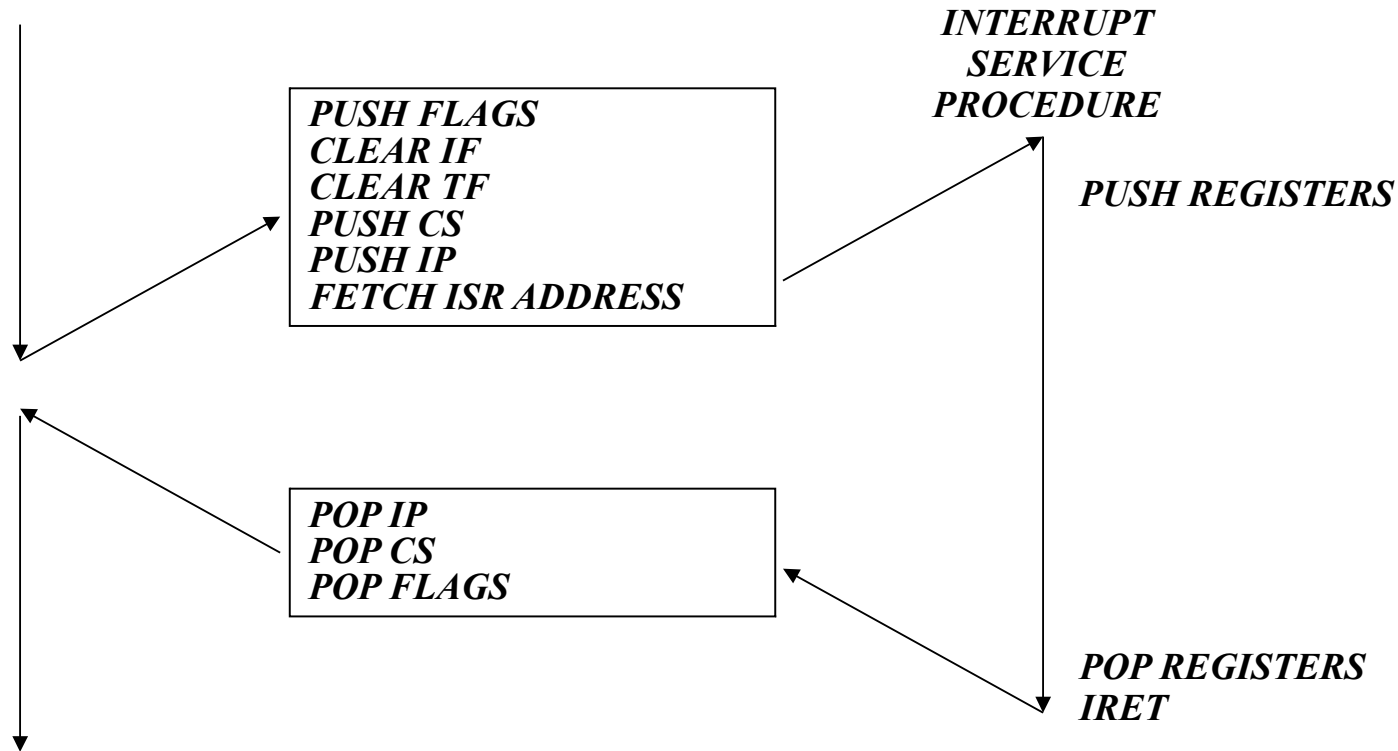
- CF - Vai-um
- PF - Paridade
- AF - Vai-um auxiliar
- ZF - Zero
- SF - Sinal
- OF - Estouro
- IF – Interrupção
- DF - Direção
- TF - Passo Simples
- IOPL - Nível de Prioridade da Tarefa
- NT - Tarefa Aninhada'

RF – Resumo  
 VM – Modo 8086  
 Virtual  
 AC – Checagem de  
 Alinhamento  
 VIF – Interrupção  
 Virtual Pendente  
 VIP – Interrupção  
 Virtual  
 ID - ID

# Sequência de Processamento

1. Pegar o número do vetor
  - INT xxx
  - INTO, NMI, divide-error
  - INTR (barramento de dados)
2. Salvar as informações da CPU
  - Salva o registrador de flags
  - IF = 0 e TF=0
  - CS e IP empilhados
3. IP e CS são pegos da tabela de vetores de interrupção

# Sequência de Processamento



# Interrupções ao mesmo tempo

- Adoção de Prioridades

Interrupt	Prioridade
Divide-Error	Maior
INT, INTO	
NMI	
INTR	
Single-Step	Menor

# Interrupções ao mesmo tempo

- Exemplo

Divisão por zero e NMI estão ativadas

1. Divisão por zero é reconhecida
2. Contexto salvo
3. NMI é reconhecida
4. Contexto salvo
5. ISR de NMI é executado
6. Contexto em 4 é restaurado
7. ISR de divisão por zero é executado
8. Contexto em 2 é restaurado
9. A execução continua na instrução seguinte aquela que gerou a divisão por zero

# Algumas interrupções especiais

## **Erro de divisão (tipo 0)**

MOV BL,0

DIV BL

A ISR pode mover 0 para AL

## **Single-Step (tipo 1)**

TF=1

Interrupção a cada instrução executada

ISR pode verificar o estado dos registradores

# Algumas interrupções especiais

## NMI (tipo 2)

Ex: Adoção em uso de erros de hardware não recuperáveis (Falha de alimentação, erro de paridade)

## Breakpoint (tipo 3)

Ajuda na depuração

O primeiro byte da instrução é trocado pelo código do breakpoint (CC)

Quando a CPU encontra tal código gera interrupção tipo 3

Na saída da ISR troca o byte do breakpoint com o primeiro byte original da instrução

Endereço

06200

F6

06201

D0

Endereço

06200

CC

06201

D0

TEMP

F6

# Algumas interrupções especiais

## **Overflow(tipo 4)**

INTO com OF = 1

Ex: overflow em operações de adição

## **INTR**

IF=1 (STI)

INTR precisa ficar 1 até a CPU reconhecer o sinal

Inicializa um ciclo de reconhecimento de interrupção (leitura do vetor no barramento de dados)

# Rotina de Tratamento de ISR

```
ISR: ;salva os registradores
      ; corpo da ISR
      ;restaurar os registradores
      IRET
```

## **ISR erro de divisão por zero**

```
DIVERR: PUSH SI
        MOV AX, 0101H
        MOV DX,0
        LEA SI, DIVMSG
        CALL FAR PTR DISPMSG
        POP SI
        IRET
```

# Rotina de Tratamento de ISR

ISR20H: CMP AH,4 ;AH precisa estar entre 0-3

JGE EXIT

CMP AH, 0

JZ ADDAB

CMP AH,1

JZ SUBAB

CMP AH,2

JZ MULAB

DIV BL

EXIT: IRET

ADDAB: ADD AL,BL  
IRET

SUBAB: SUB AL, BL  
IRET

MULAB: MUL BL  
IRET