



Priority Interrupt Controller 8259A

Walter Fetter Lages

w.fetter@ieee.org

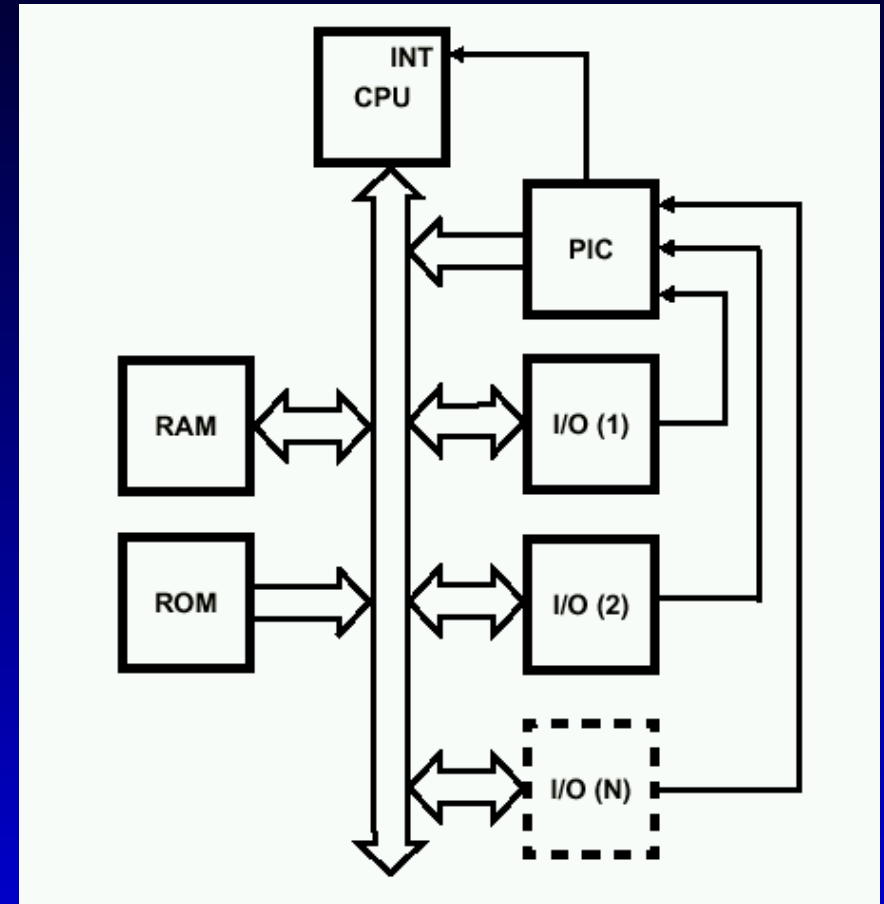
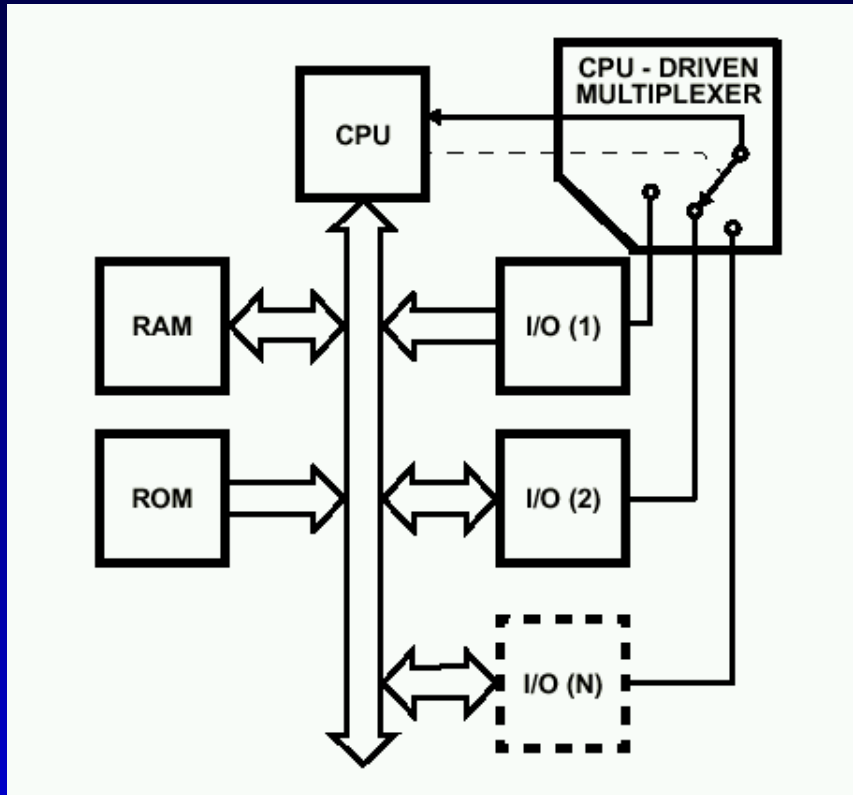
Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Engenharia Elétrica

Microprocessadores II

Polling x Interrupção





Interrupções

- Mascaráveis
 - Podem ser desabilitadas
- Não mascaráveis
 - Não podem ser desabilitadas
- Ativas por borda
 - Solicitação ativa quando ocorre transição (subida ou descida) do sinal
- Ativas por nível
 - Solicitação ativa quando o sinal está em um determinado nível (alto ou baixo)



Interrupções

- Latência
 - Tempo entre a solicitação de interrupção e o início da rotina de tratamento
- Aninhamento
 - Se o sistema permite aninhamento de interrupções, rotinas de tratamento de interrupções podem ser interrompidas
- Prioridade
 - Interrupções de maior prioridade interrompem as rotinas de tratamento das interrupções de menor prioridade



Tipos de Interrupções

- Interrupções (de *hardware*)
- Exceções
 - Geradas internamente pelo processador
- Interrupções de software
 - Concebidas para testar as rotinas de tratamento de interrupção
 - Interface com o sistema operacional



Exceções

- Faltas
 - Gerada antes da execução da instrução
 - No retorno da rotina de tratamento, a instrução que causo a falta é re-executada
- Traps
 - Gerada após a execução da instrução
- Abortos
 - Não permitem o reinício da execução
 - Erros de *hardware*
 - Inconsistência nas tabelas de sistema



Interrupções no 80x86

- Interrupção mascarável
 - Solicitada através do pino INTR
 - Se estiver habilitada, após terminar a execução da instrução atual o processador reconhece a interrupção
- Interrupção não-mascarável
 - Solicitada através do pino NMI
 - Após terminar a execução da instrução atual o processador desvia para a rotina de tratamento
- Interrupção de software
 - Gerada através da instrução INT n
 - Não podem ser desabilitadas



Atendimento de Interrupções

- A CPU ativa o pino INTA# por um ciclo de barramento para reconhecer a interrupção
- A CPU ativa o pino INTA# em outro ciclo de barramento para obter o vetor de interrupção
- Salva o registrador de flags na pilha
- Reseta IF e TF
- Carrega novos valores de CS e IP
 - O vetor de interrupção é utilizado como índice na IDT para obter o endereço da rotina de tratamento



Interrupt Descriptor Table

- Tabela de vetores de interrupção
- O vetor de interrupção é utilizado com índice desta tabela
- No 8086 a tabela está nos endereços 0-3FFh
- No 80386+ a tabela está no endereço apontado por IDTR
- IDTR é inicializado com 0 no reset
 - Compatibilidade com processadores anteriores

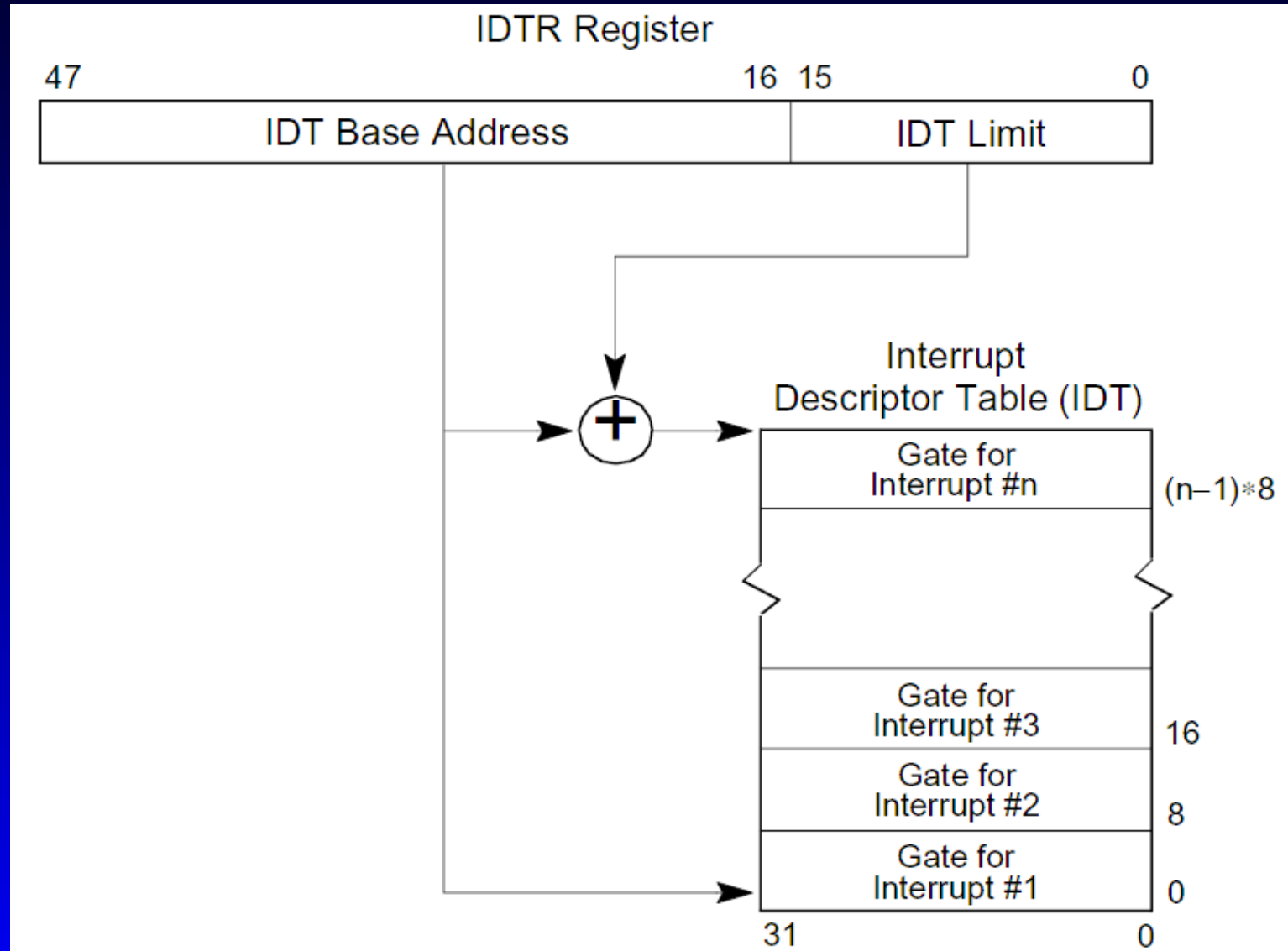


Interrupt Descriptor Table

- Modo real
 - Cada entrada possui 4 bytes com o endereço da rotina de tratamento de interrupção associada ao vetor correspondente na forma segmento:offset
- Modo protegido:
 - Em modo protegido cada entrada é um *gate descriptor*

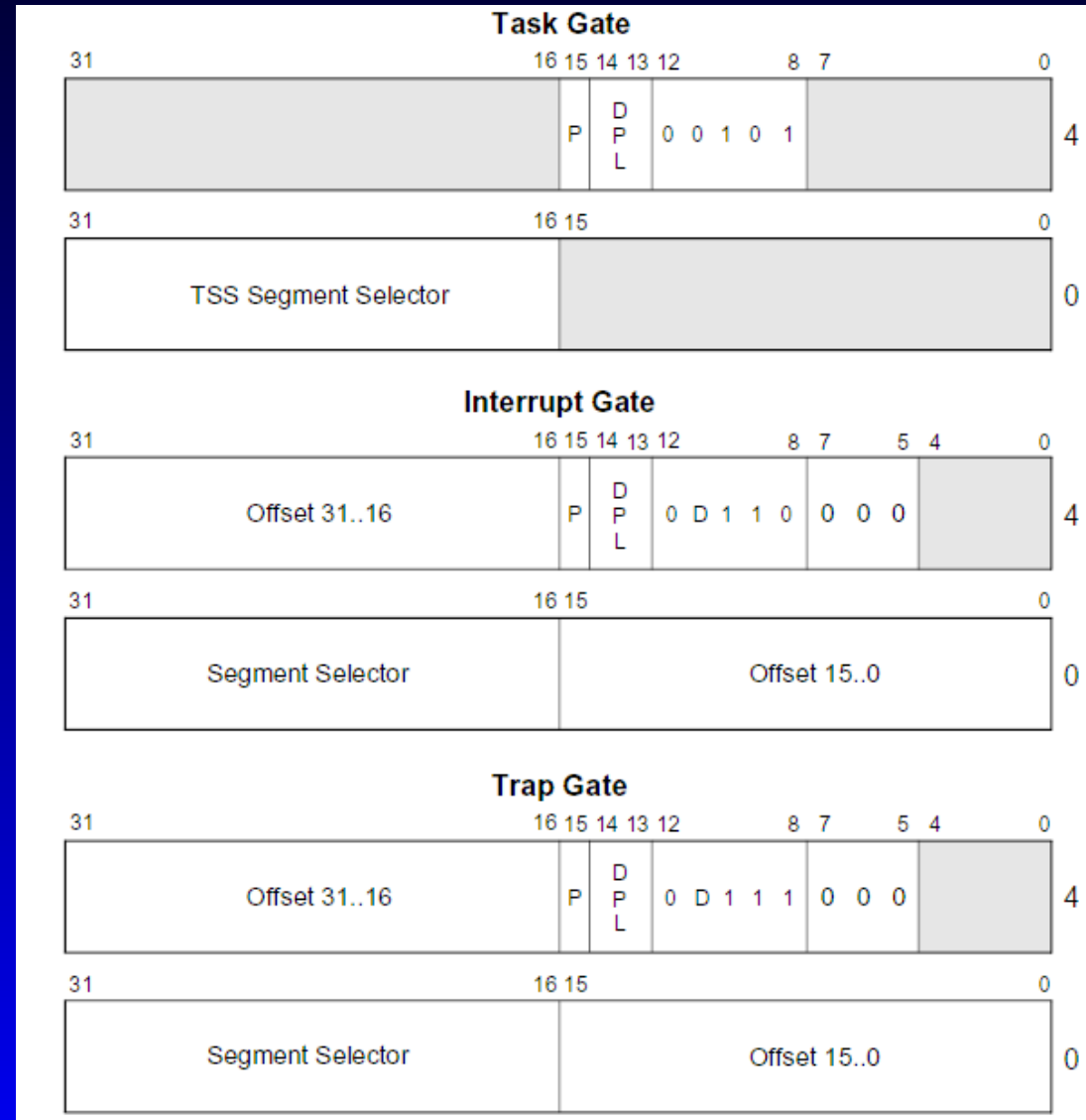


IDTR



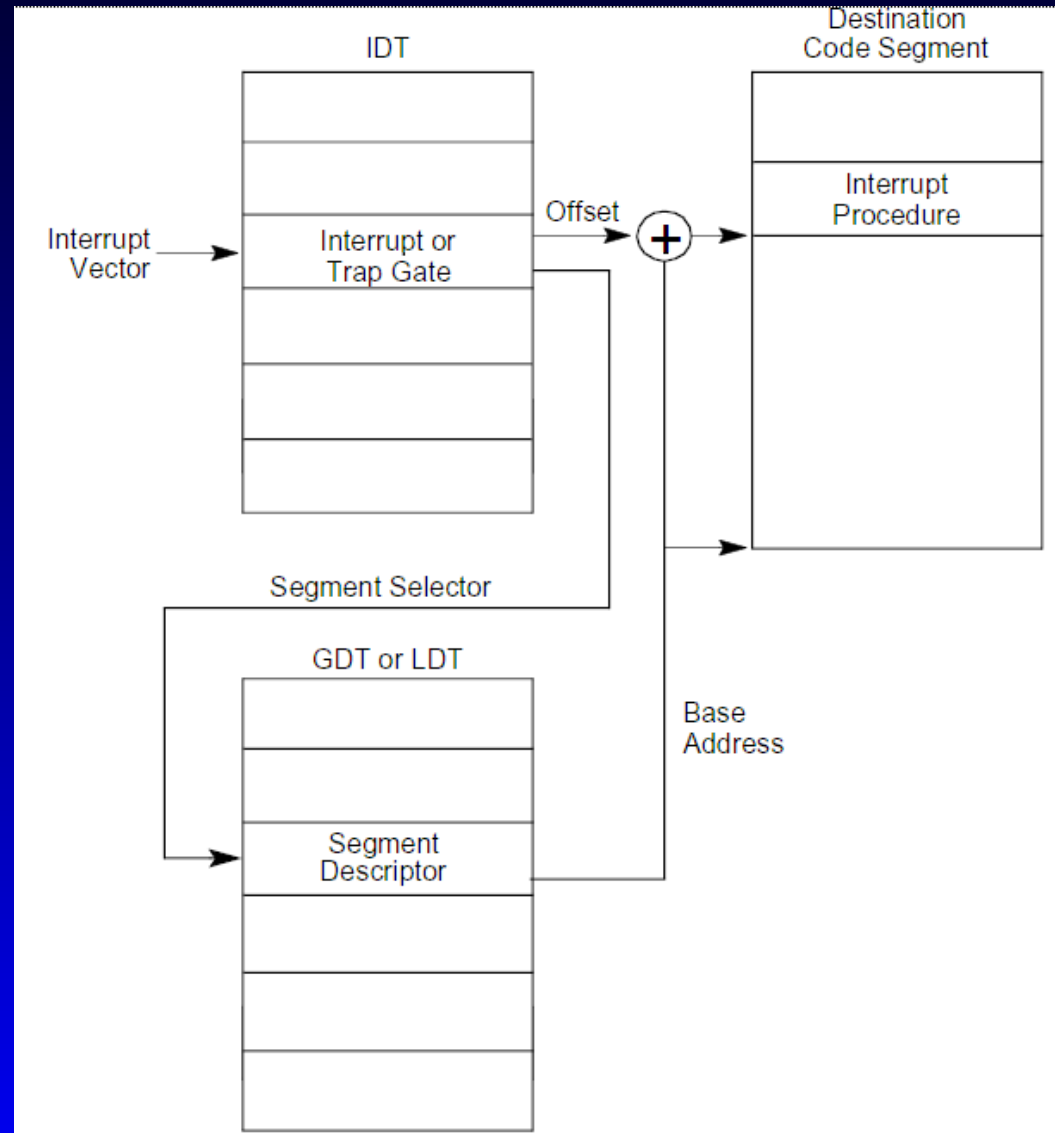


Descritores de Gates





Rotinas de Tratamento





Excessões de Modo Real

Vetor	Tipo	Código
0	Divide Error Fault	No
1	Debug Fault/Trap	No
2	NMI Interrupt Interrupt	No
3	Breakpoint Trap	No
4	Overflow Trap	No

Excessões de Modo Protegido



Vetor	Tipo	Código
5	BOUND Range Exceeded Fault	No
6	Invalid Opcode Fault	No
7	Device Not Available (No Math Coprocessor) Fault	No
8	Double Fault Abort	Yes (Zero)
9	Coprocessor Segment Overrun (reserved) Fault	No
10	Invalid TSS Fault	Yes
1	Segment Not Present Fault	Yes
12	Stack-Segment Fault Fault	Yes
13	General Protection Fault	Yes

Excessões de Modo Protegido



Vetor	Tipo	Código
14	Page Fault Fault	Yes
15	(Intel reserved. Do not use.)	No
16	Floating-Point Error (Math Fault) Fault	No
17	Alignment Check Fault	Yes
18	Machine Check Abort	No
19-31	Intel reserved	Do not use



Rotina de Tratamento

- Instalação

```
int request_irq(unsigned int irq,  
               irqreturn_t (*handler)(int irq,void *dev_id,  
               struct pt_regs *regs),  
               unsigned long flags, const char *device, void *dev_id)
```

- Remoção

```
void free_irq(unsigned int irq,void *dev_id)
```



Flags

IRQF_DISABLED: a rotina de tratamento é chamada com interrupções desabilitadas

IRQF_SAMPLE_RANDOM: a interrupção é usada para alimentar o gerador aleatório

IRQF_SHARED: permite o compartilhamento da interrupção

IRQF_PROBE_SHARED: indica que é esperado que ocorram erros de compartilhamento

IRQF_TIMER: a interrupção é uma interrupção do *timer*

IRQF_PERCPU: a interrupção é por CPU

IRQF_NOBALANCING: exclui a interrupção do balanço de interrupções

IRQF_IRQPOLL: a interrupção é usada para *polling* (apenas a primeira interrupção registrada é considerada)



Rotina de Tratamento

- Deve retornar

IRQ_NONE: indica que a interrupção não foi tratada (e deve ser tratada por outra rotina de tratamento)

IRQ_HANDLED: indica que a interrupção foi tratada

IRQ_RETVAL(x): se $x=0$, indica que a interrupção não foi tratada, se $x \neq 0$, indica que a interrupção foi tratada

```
static volatile int count;
```

```
irqreturn_t lpiirq(int irq,void *dev_id,struct pt_regs *regs)  
{  
    count++;  
  
    return IRQ_HANDLED;  
}
```



Encadeamento de Rotinas

- No caso de ter-se diversos dispositivos utilizando a mesma interrupção, as rotinas de tratamento podem ser encadeadas
- Ao invés de retornar com `IRQ_HANDLED` a rotina pode retornar `IRQ_NONE` para passa o controle para outra rotina



Acesso aos Vetores de Interrupção

- Ao invés de acessar diretamente a IDT, deve-se utilizar os serviços do sistema operacional para acessar os vetores de interrupção
- Em modo protegido a IDT pode estar em qualquer lugar na memória
- O formato das entradas da IDT é diferente dependendo do modo

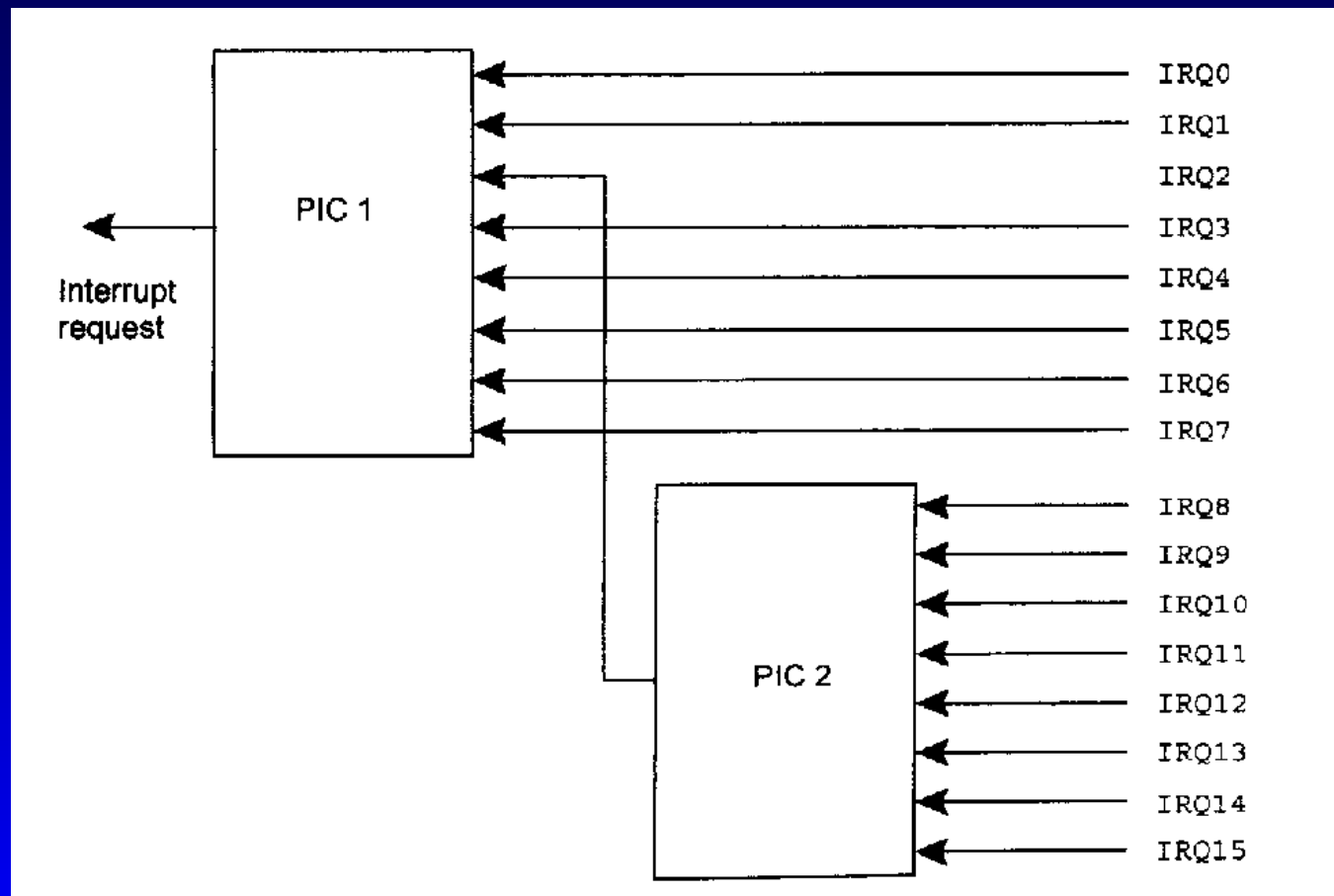


NMI no PC

- Pode ser desabilitada através de um flip-flop no bit 0 da porta 70h
- Erro de paridade na memória
- Erro no barramento ISA IOCHK#
- Erro no co-processador
- Erro de sistema no barramento PCI (SERR#)
 - Opcional

Interrupções no PC

- 2 controladores 8259A em cascata
 - PIC1 em 20h, PIC2 em a0h

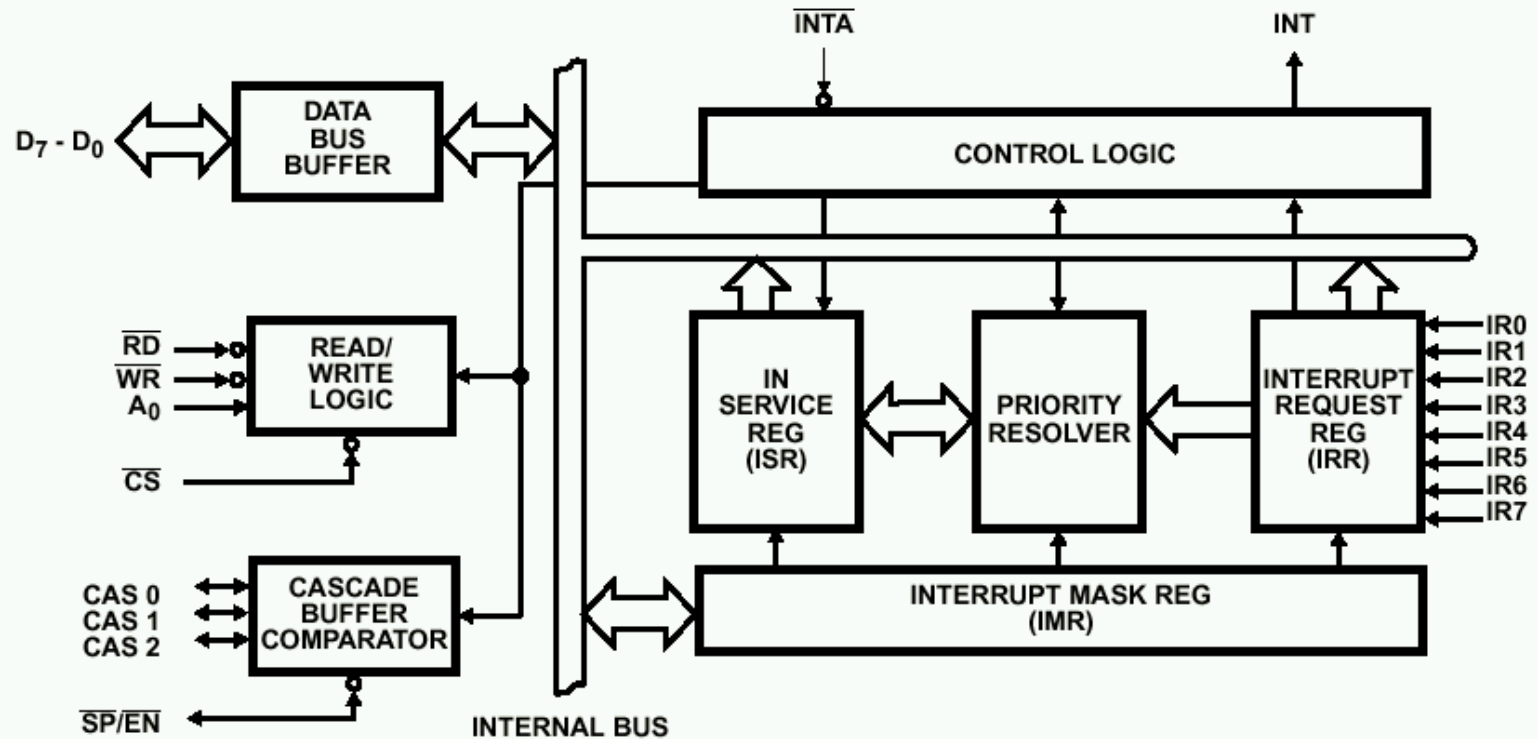




8259A

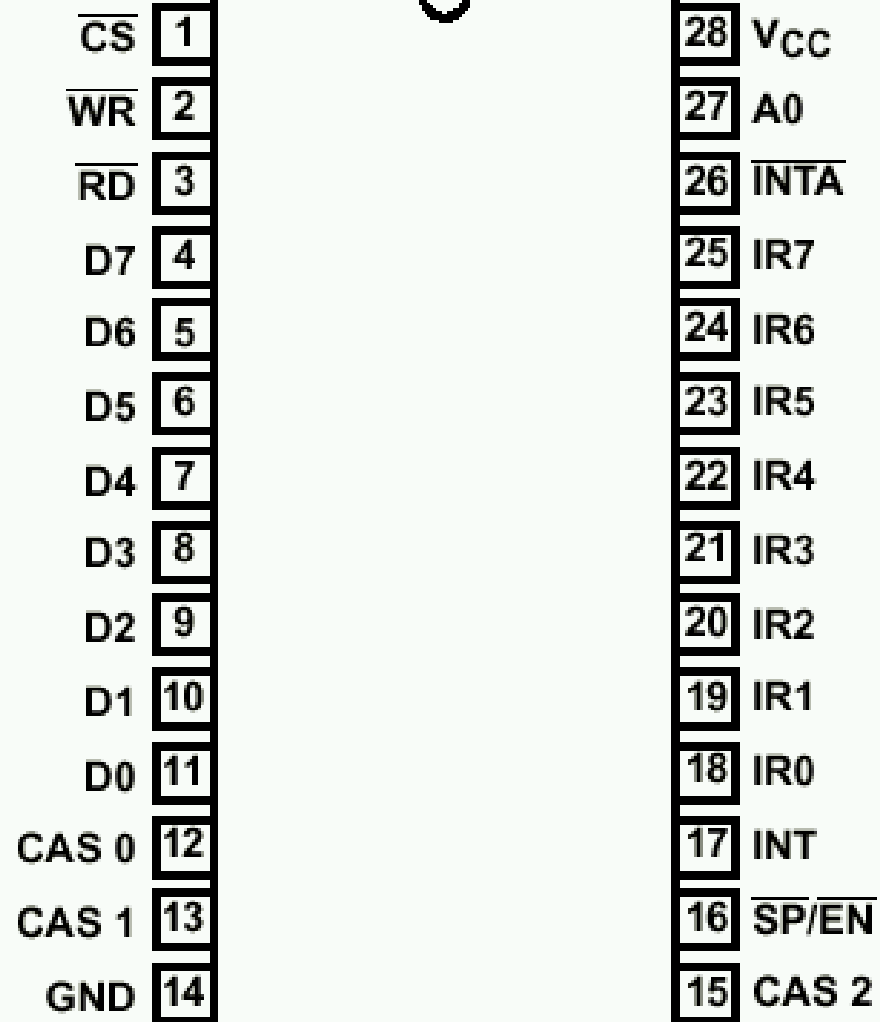
- IRQs podem ser habilitadas/desabilitadas individualmente
- Vetores de interrupção em seqüência
- O sinal IRQ deve ficar ativo até após o início do primeiro pulso INTA#
 - Caso contrário o 8259A assume uma interrupção espúria, que é mapeada para IRQ7
 - O software deve estar preparado para tratar esta situação

Diagrama de Blocos do 8259A



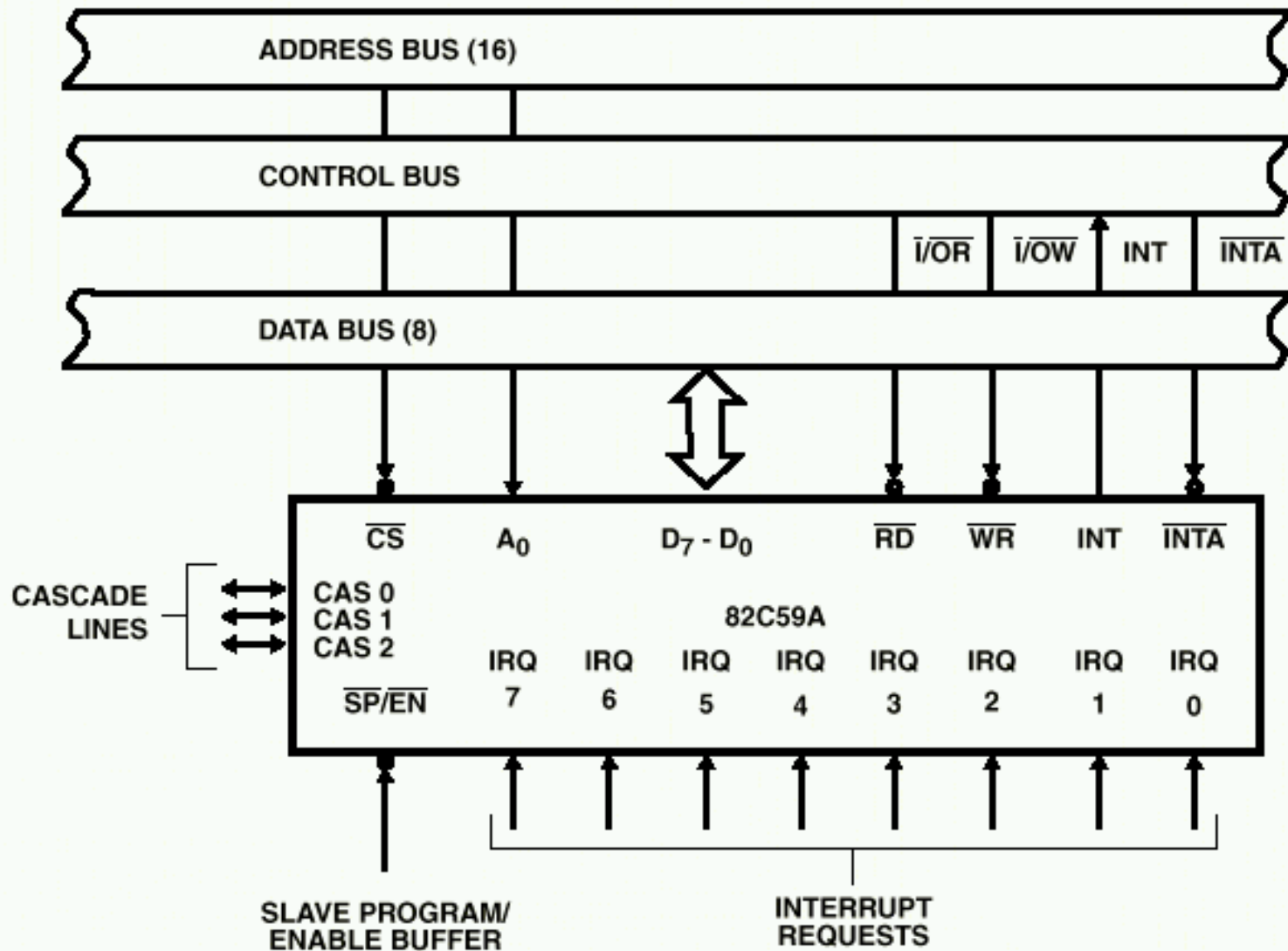


Pinagem do 8259A





Interface do 8259A





Programação do 8259A

- Palavras de comando de inicialização
 - Obrigatórias
 - ICW1 (A0=0)
 - ICW2 (A0=1)
 - ICW3 (A0=1) se operando em modo de cascata
 - Opcional
 - ICW4 (A0=1)
- Palavras de comando de operação
 - OCW1 (A0=1)
 - OCW1 (A0=0)
 - OCW2 (A0=0)



ICW1

D7	D6	D5	D4	D3	D2	D1	D0
A7	A6	A5	1	LTIM	ADI	SNGL	IC4

A7 A6 A5: bits do vetor de interrupção (modo 8085)

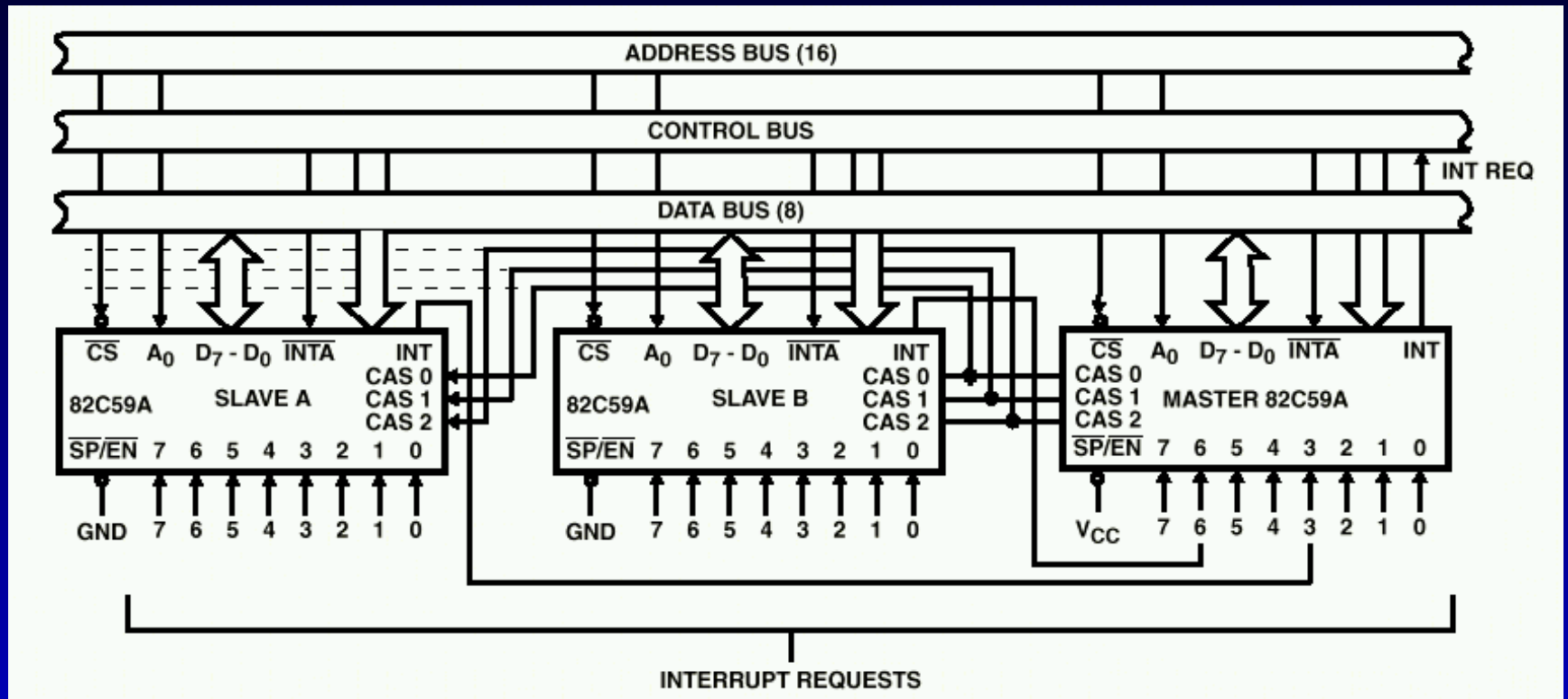
LTIM: 1=disparo por nível, 0=disparo por borda

ADI: Call Address Interval, 1=intervalo de 4,
0=intervalo de 8

SNGL: 1=modo solitário, 0=modo cascata

IC4: 1=ICW4 será utilizada, 0=ICW4 não utilizada

Cascadeamento de 8259As





ICW2

D7	D6	D5	D4	D3	D2	D1	D0
A15/T7	A14/T6	A13/T5	A12/T4	A11/T3	A10	A9	A8

A15-A8: bits do vetor de interrupção (modo 8085)

T7-T4: bits do vetor interrupção (modo 8086/8088)



ICW3 (Mestre)

D7	D6	D5	D4	D3	D2	D1	D0
S7	S6	S5	S4	S3	S2	S1	S0

- Bit em 1 indica que existe escravo conectado na entrada correspondente



ICW3 (Escravo)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	ID2	ID1	ID0

ID2 ID1 ID0: Identificação do escravo

- Identifica para o escravo em qual IRQ do mestre ele está conectado
- No reconhecimento de interrupção, quando o valor nas entradas de cascata for igual a ID2 ID1 ID0, o escravo fornece vetor de interrupção

ICW4

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	SFNM	BUF	M/S	AEOI	μ PM

SFNM: habilita modo totalmente aninhado especial

BUF: habilita modo buferizado e faz SP#/EN#
tornar-se entrada

M/S: seleciona modo mestre ou escravo quando em
modo buferizado

AEOI: habilita fim de interrupção automático

μ PM: 1=modo 8086/8088, 0=modo 8085 (default)



End of Interrupt

- Quando uma interrupção é reconhecida, o bit correspondente no ISR é setado
- O comando EOI reseta o bit no ISR
 - EOI é enviado através da OCW2
 - EOI específico
 - Reseta o bit especificado
 - EOI não específico
 - Reseta o bit da última IR processada
 - EOI automático
 - Bit resetado na borda final de INTA#



Modo Bufferizado

- Em sistemas grandes em que são utilizados buffers no barramento de dados existe o problema de habilitar os buffers
- Quando são utilizados 8259As em cascada, eventualmente buffers diferentes precisam ser habilitados para o mestre e para o escravo
- O 8259A pode utilizar o sinal SP#/EN# para habilitar os buffers
- Neste caso a programação de mestre/escravo precisa ser feita por software



Modo Aninhado

- É o modo default
- As IR são priorizadas
- Quando uma interrupção é reconhecida o pedido de maior prioridade é determinado e o vetor correspondente liberado no barramento e o bit IS_n é setado
- Enquanto o bit IS_n estiver setado, interrupções de prioridade igual ou menor são desabilitadas



Modo Aninhado Especial

- Em sistemas utilizando 8259As em cascata, enquanto uma interrupção do escravo estivesse sendo atendida, todas as demais estariam bloqueadas, pois o bit IS_n no mestre estaria setado
 - Pode ocorrer inversão de prioridade
- No modo totalmente aninhado especial, os bits do ISR que possuem escravos não bloqueiam novos pedidos de interrupção do mesmo nível



OCW1

D7	D6	D5	D4	D3	D2	D1	D0
M7	M6	M5	M4	M3	M2	M1	M0

- Máscara de interrupção
- Bit em 1 desabilita interrupção



OCW2

D7	D6	D5	D4	D3	D2	D1	D0
R	SL	EOI	0	0	L2	L1	L0

L2 L1 L0: nível de interrupção atuado pelo comando

R	SL	EOI	Comando
0	0	1	EOI não específico
0	1	1	EOI específico
1	0	1	Rodar em EOI não específico
1	0	0	Rodar em EOI automático
0	0	0	Não rodar em EOI automático
1	1	1	Rodar em EOI específico
1	1	0	Seta prioridade (menor)
0	1	0	Nenhuma operação



OCW3

D7	D6	D5	D4	D3	D2	D1	D0
0	ESMM	SMM	0	1	P	RR	R S

ESPM	SMM	Comando
0	0	Nenhuma ação
0	1	Nenhuma ação
1	0	Reseta máscara especial
1	1	Seta máscara espacial

P: Habilita modo de poll

ESPM	SMM	Comando
0	0	Nenhuma ação
0	1	Nenhuma ação
1	0	Lê IR no próximo pulso de RD#
1	1	Lê IS no próximo pulso de RD#



Modo de Máscara Especial

- Normalmente quando uma interrupção é reconhecida, as interrupções prioridade igual ou menor ficam bloqueadas até que seja enviado o EOI
- No modo de máscara especial apenas as interrupções de igual prioridade ficam bloqueadas



Modo de Polling

- Neste modo não é utilizado o pino INT, ou o processador está com as interrupções desabilitadas
- Ao receber um comando de Poll o 8259A trata o próximo ciclo de RD# como um reconhecimento de interrupção, setando o bit ISn apropriado e libera o nível de prioridade no barramento
- Interrupções são congeladas entre o WR# e o RD#



Leitura de Status do 8259A

- Interrupt Request Register (IRR)
 - Pode ser lido via OCW3
- In-Service Register (ISR)
 - Pode ser lido via OCW3
- Interrupt Mask Register (IMR)
 - Pode ser lido via OCW1
 - Leitura direta com A0=1

Inicialização do 8259A no PC

- Interrupções ativas por borda
- Prioridade fixa

0, 1, 2, 3, 4, 5, 6, 7

8, 9, 10, 11, 12, 13, 14, 15

- Base dos vetores de interrupção
 - $PIC1 = 08h \implies IRQ0 \equiv INT\ 8h$
 - $PIC2 = 70h \implies IRQ8 \equiv INT\ 70h$
 - Coincidência de vetores do PIC1 com excessões de modo protegido
 - Sistema Operacional reprograma base dos vetores



Interrupções no XT

- O XT tem apenas um 8259A
- No slot ISA, a IRQ9 ocupa o lugar da IRQ2 que existia no slot do XT
 - Redirecionamento por software da rotina de tratamento da IRQ9 para IRQ2
 - A rotina de tratamento de IRQ9 executa um INT 0Ah



Interrupções ISA

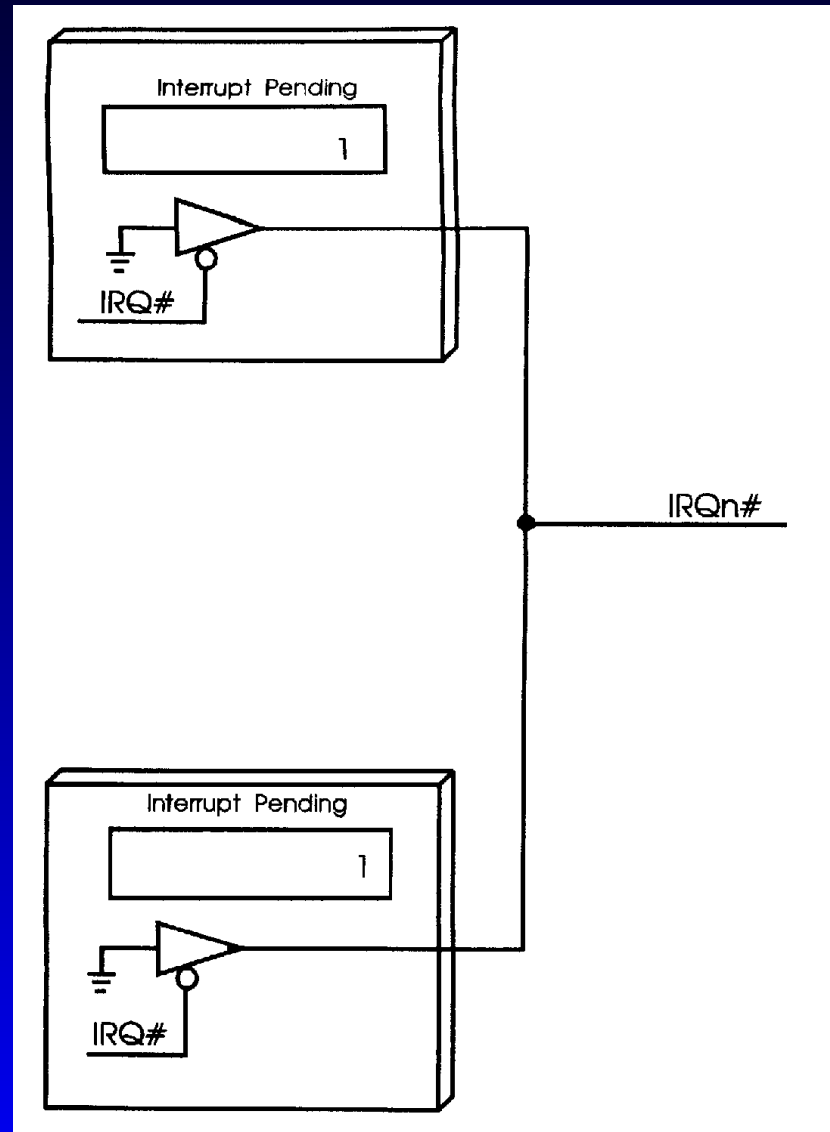
- IRQ0 timer
- IRQ1 teclado
- IRQ2 cascata
- IRQ3 COM2/COM4
- IRQ4 COM1/COM3
- IRQ5 LPT2/HD-XT
- IRQ6 floppy
- IRQ7 LPT1
- IRQ8 RTC
- IRQ9 livre/VGA
- IRQ10 livre
- IRQ11 livre
- IRQ12 livre/PS2
- IRQ13 FPU
- IRQ14 IDE
- IRQ15 livre/IDE2



Compartilhamento de IRQs

- Pela especificação do barramento ISA, as IRQs devem ser geradas por um circuito em coletor aberto
- IRQs deveriam poder ser compartilhadas
 - Poucos fabricantes implementam corretamente o circuito de geração de IRQ
 - Compartilhamento apenas seria possível se fossem ativas por nível (EISA)
 - Poucos drivers estão preparados para possibilitar o compartilhamento de IRQs

Compartilhamento de IRQs

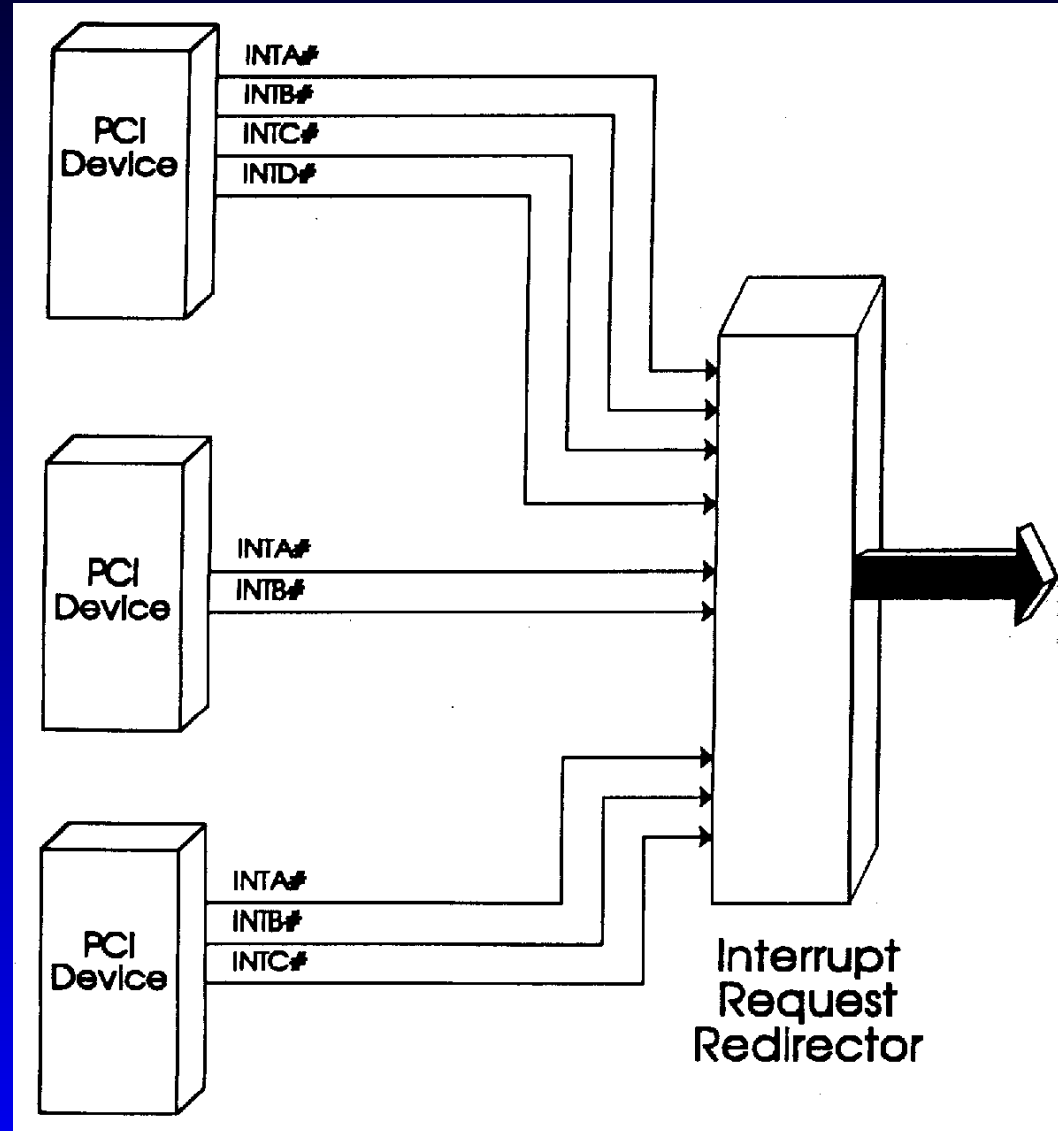




Interrupções PCI

- 4 linhas de requisição de interrupção
- INTA#, INTB#, INTC#, INTD#
- Ativas por nível, ativas em baixo, compartilháveis
- Dispostivos PCI com uma única função devem utilizar INTA#
- Cada função deve utilizar apenas uma linha de requisição de interrupção

Interrupções PCI





Roteamento de Interrupções

- As interrupções PCI devem ser roteadas para o controlador de interrupções
- Cada função PCI que utiliza interrupção deve implementar dois registradores de configuração
 - Interrupt Pin Register (R/O)
 - Interrupção PCI
 - Interrupt Line Register (R/W)
 - Linha de IRQ a ser utilizada



APIC

- APIC=Advanced Programmable Interrupt Controller
- Desenvolvido para ser utilizado em sistemas com mais de um processador
 - Pode ser utilizado em sistemas uniprocessador
 - Menor latência do que o controlador com 8259A
 - Pode ser utilizado conjuntamente com o 8259A
 - Cada controlador controla algumas interrupções
- Suporta a distribuição das interrupções entre os processadores



Sistema APIC

- Composto por duas partes que se comunicam através do barramento APIC
 - Local APIC
 - Localizado em cada processador
 - Determina se o processador deve processar a interrupção difundida no barramento APIC
 - Gera interrupções entre os processadores
 - Possui um temporizador
 - I/O APIC
 - Suporta linhas de interrupção para os periféricos
 - Possui tabela de redirecionamento de interrupções

Tabela de Redirecionamento



- Armazena as características de cada linha de interrupção
 - Ativa por borda ou por nível
 - Vetor de interrupção
 - Prioridade
 - Processador de destino
 - Seleção de processador estática ou dinâmica
 - Estática = interrupção associada ao processador
 - Dinâmica = processador selecionado dinamicamente

Barramento APIC

