#### **ELE-8575**

#### Interrupções

# Slides baseados no material do curso do Prof. Chen Lian Kuan

#### Interrupções

- > Tem-se três categorias básicas de interrupções:
  - 1. Interrupções de software: geradas pela instrução INT interrupt-vector;
  - Interrupções de hardware: geradas pela ativação do pino físico NMI (Non-Maskable Interrupt) ou ativação do pino físico INTR (que pode ser mascarável ou não, dependendo-se da flag IF)
  - 3. Interrupções automáticas: geradas por determinadas condições de erro (por exemplo, divisão por 0), ou no final de cada instrução se a flag TRAP estiver definida no registo da FLAGS.
- Como veremos, todas as interrupções são processadas da mesma forma:

```
(push FLAGS, clear IF, TF, push CS, push IP, load interrupt vector into CS:IP).
```

#### Interrupções

- A interrupção de hardware INTR pode ser mascarável ou não, dependendo-se do estado da flag de interrupção no registro FLAGS.
  - Se IF = '1' o μP aceita pedidos de interrupção que chegam pelo pino físico INT (a interrupção não foi mascarada);
  - Se IF = '0' o μP não aceita pedidos de interrupção que chegam pelo pino físico INTR (a interrupção foi mascarada).
- As interrupções de hardware permitem que dispositivos externos (geralmente associados a uma porta de I/O) sejam atendidos pelo microprocessador apenas quando necessitam de atenção o microprocessador pode executar outras tarefas quando o dispositivo não necessita da atenção dedicada do microprocessador.

### Instruções associadas

Mnemonic	Meaning	Format	Operation	Flags Affected
CLI	Clear interrupt flag	CLI	$0 \rightarrow (IF)$	IF
STI	Set interrupt flag	STI	$1 \rightarrow (IF)$	IF
INT n	Type n software interrupt	INT n	$(Flags) \rightarrow ((SP - 2)$	TF, IF
			$0 \rightarrow TF$ , IF	
			$(CS) \rightarrow ((SP) - 4)$	
			(2+4·n) → (CS)	
			$(IP) \rightarrow ((SP) - 6)$	
			$(4\cdot n) \rightarrow (IP)$	All
IRET	Interrupt return	IRET	$((SP)) \rightarrow (IP)$	7 (11
			$((SP) + 2) \rightarrow (CS)$	
			((SP) + 4) → (Flags)	
			$(SP) + 6 \rightarrow (SP)$	TF, IF
INTO	Interrupt on overflow	INTO	INT 4 steps	,
	Halt		Wait for an external	None
			Interrupt <u>or re</u> set to occur	
WAIT	Wait	WAIT	Wait for TEST input to	None
			go active \	

→ Usually connected to BUSY of 8087

### Alguns tipos Pré-definidos de Interrupções

#### Type 0: Divide error interrupt

A interrupção número 0 é gerada quando é feita uma tentativa de divisão por zero. A interrupção é gerada automaticamente.

#### Type 1: Single-Step Interrupts

Quando o bit 8 do registro FLAGS (trap flag ou TF) é feito TF='1', é gerada a interrupção número 1 após a execução de uma instrução. Quando o microprocessador entra na rotina de serviço de interrupção, a flag TF é automaticamente apagada (caso contrário, não seria capaz de ir além da primeira instrução da rotina de serviço de interrupção).

#### Type 2: NMI (Non-Maskable Interrupt)

Interrupção de hardware NMI.

### Alguns tipos Pré-definidos de Interrupções

#### Type 3: Breakpoint (One-byte) Interrupt

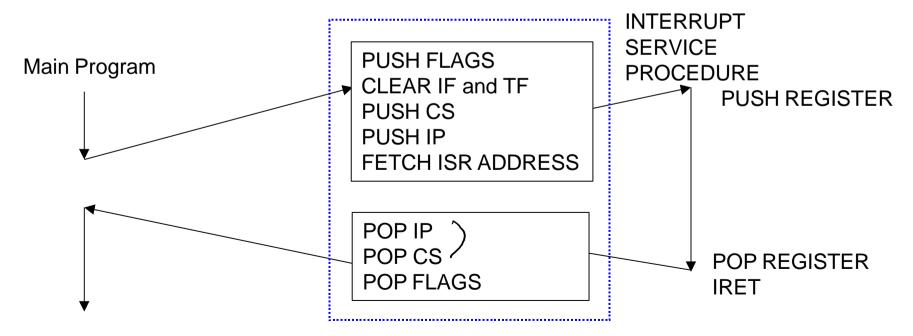
A interrupção número 3 tem um *opcode* CCh e pode ser inserido em qualquer lugar num programa para gerar um ponto de interrupção. A instrução assembly é INT3.

#### Type 4: Overflow

Uma operação realizada na ULA gerou situação de overflow.

## Sequência de Resposta a um pedido de Interrupção

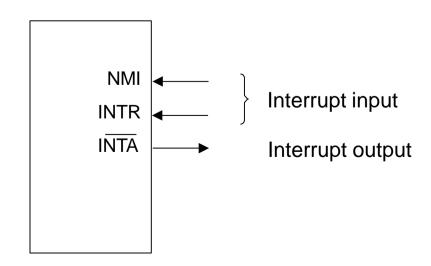
- 1. PUSH Flag Register
- 2. Clear IF and TF
- 3. PUSH CS
- 4. PUSH IP
- 5. Buscar o conteúdo do vector de Interrupção e colocar em ambos IP e CS para iniciar o Procedimento de Serviço de Interrupção (ISP)



### Interrupções de Hardware (NMI e INTR)

Hardware interrupts (interrupções de hardware) são muito eficientes na manipulação de dispositivos periféricos, particularmente operações de E/S, uma vez que o processador pode realizar outras tarefas e só presta serviços ao periférico quando necessário.

NMI é normalmente reservado para o tipo de interrupção mais urgente (por exemplo, falha iminente de energia de um periférico), enquanto que o INTR é utilizado para interrupções do sistema.



### Interrupções de Hardware (NMI e INTR)

- ➤ Um sinal de disparo de interrupção no pino NMI gera a interrupção número 2 (o vetor de interrupção está localizado no local de memória 0008, o novo endereço (P) de offset é armazenado em 00008 e 00009; o novo endereço de segmento (C) é armazenado em 0000Ah e 0000Bh). A interrupção NMI é ativada na borda de subida, ou seja, numa transição de 0 para 1).
- Quando mais de um periférico é ligado ao pino da MNI, a rotina de serviço de interrupção pode ter de verificar qual o periférico que gerou a MNI, através da sondagem (pooling) das possíveis fontes da MNI.
- ➤ O pino INTR is level sensitive (sensível ao nível) deve ser mantido no N.L.1 até ser reconhecido pelo INTA. Com a verificação de IF, o INTR é automaticamente desativado quando o microprocessador já está tratando da INTR. INTR é reativado no final da rotina de serviço de interrupção (interrupt service routine ou ISR).
- O número da interrupção gerado por um INTR é lido a partir do barramento de dados.

#### **Prioridade**

Quando ocorrem diferentes tipos de interrupção (ou seja, software, MNI, INTR ou interrupções automáticas) ao mesmo tempo, a que tem a maior prioridade é tratada em primeiro lugar. Os microprocessadores Intel utilizam a seguinte ordem de prioridade:

Interrupt Type	Priority
divide error interrupt, INT n, INTO NMI INTR TRAP flag (single step)	highest_ t lowest

#### **Prioridade**

Se vários INTR forem gerados a partir de diferentes periféricos simultaneamente, é necessário decidir a sua prioridade e enviar o sinal INTA apenas para o periférico de maior prioridade. (Se mais do que um periférico receber um INTA ao mesmo tempo, ambos tentarão colocar o número de interrupção no bus de dados simultaneamente).

Dois métodos diferentes podem ser usados para estabelecer a prioridade dos pedidos de interrupção de diferentes periféricos:

- 1. Polling (com estrutura do tipo daisy chaining)
- 2. Interrupt priority management hardware (hardware de gestão de prioridade de interrupção)

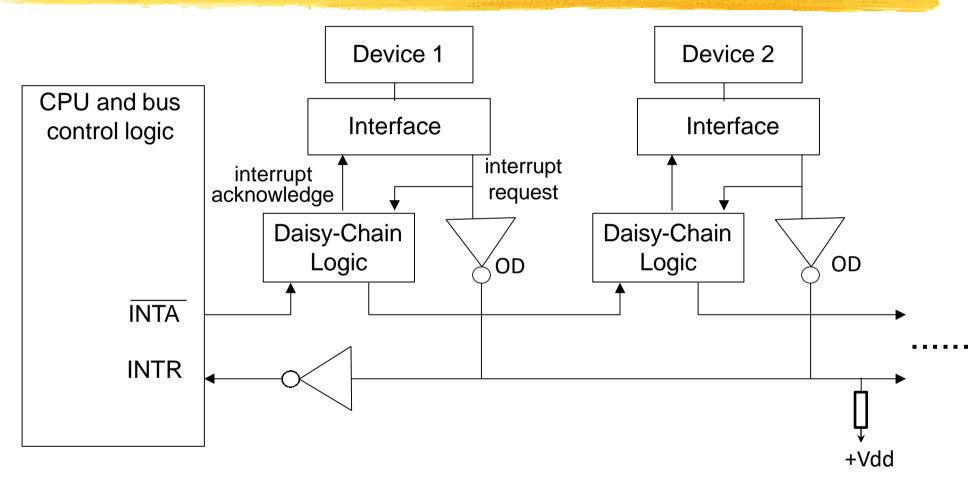
### Atribuição de prioridades por Polling e Daisy-Chaining

Polling implica perguntar a cada periférico, numa ordem pré-determinada, se necessita da atenção do microprocessador. O primeiro periférico que responder "sim" é servido pela rotina apropriada.

Daisy-chaining é um método de implementação do esquema de sondagem (pooling) por hardware. O sinal INTA passa ao longo de uma série de periféricos de um periférico para o outro apenas se o periférico em questão não estiver solicitando uma interrupção.

Assim, o primeiro periférico da estrutura daisy-chain tem a maior prioridade e o último periférico a menor prioridade.

## Atribuição de prioridades por Polling e Daisy-Chaining



Daisy chaining pode ser combinado com a sondagem por software para determinar que rotina é necessária para o periférico. (OD = *Open Drain* para CMOS e OC = *Open Collector* para TTL).

#### Controlador Programável de Interrupções

- ➤ O método daisy chain sofre a desvantagem de ser limitado no número de dispositivos que poderiam ser encadeados devido aos atrasos na passagem da INTA através da cadeia. O microprocessador espera que o número de interrupção seja colocado no barramento de dados dentro de um determinado período de tempo após o envio do INTA. Um atraso demasiado longo levará a erros.
- O CI 8259A Controlador Programável de Interrupções (ou Programmable interrupt controller) é normalmente utilizado em sistemas práticos para determinar a prioridade das interrupções.
- ➤ O 8259A atribui a prioridade de até 8 vetores de interrupção a serem ligadas ao pino INTR. O 8259A pode ser ligado em cascata (um master 8259A e oito 8259A secundários) para fornecer 64 linhas de interrupção.

## O 8259A - Programmable Interrupt Controller (PIC)

8259A é um circuito integrado de 28 pinos que foi concebido especificamente para os microprocessadores 8088/8086.

10 9 8 7 6 5 4	D0 D1 D2 D3 D4 D5 D6	IR0 IR1 IR2 IR3 IR4 IR5 IR6	18 19 20 21 22 23 24 25
27 1 3 2 16 17 26	A0 CS RD WR SP/EN INT INTA	CAS0 CAS1 CAS2	12 13 15

## O 8259A: Programmable Interrupt Controller (PIC)

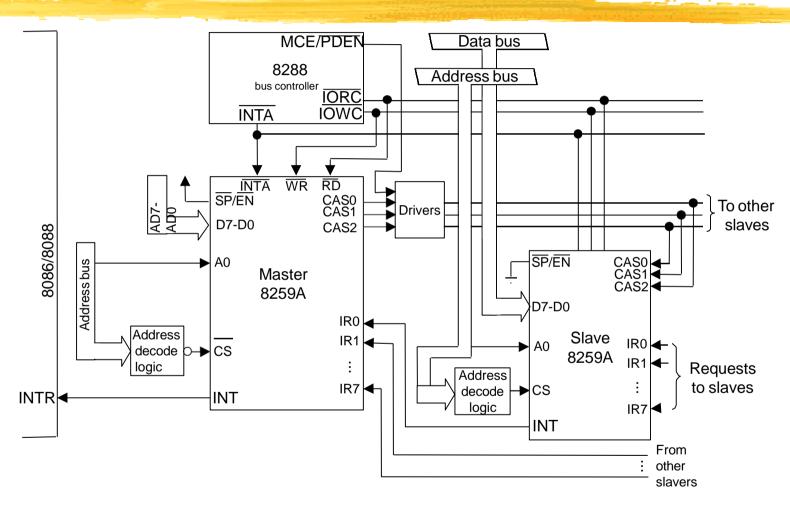
Os 28 pinos do 8259A são definidos como:

- 1. Data pins D0-D7 conectados ao bus de dados para permitir programação
- 2. IR0-IR7 8 entradas de interrupção
- 3. CAS0-CAS2 conexão em cascata (usada em sistemas com múltiplos 8259A)
- 4. SP/EN saída que habilita um buffer de dados (quando no modo buffered) ou entrada que sinaliza se o 8259A é principal ou secundário.
- 5. A0 entrada que é usada para selecionar diferentes registradores do 8259A

Usualmente IR0 tem a mais alta prioridade e IR7 a menor prioridade.

Com o 8259A é possível definir diversas políticas de interrupção. Nesse curso, vamos utilizar a forma padrão.

#### 8259A em Cascata



Cada uma das entradas de pedido de interrupção do 8259A (mestre) pode aceitar a saída INT de outro dispositivo (secundário) 8259A, aumentando assim o número de linhas de pedido de interrupção para além de 8.

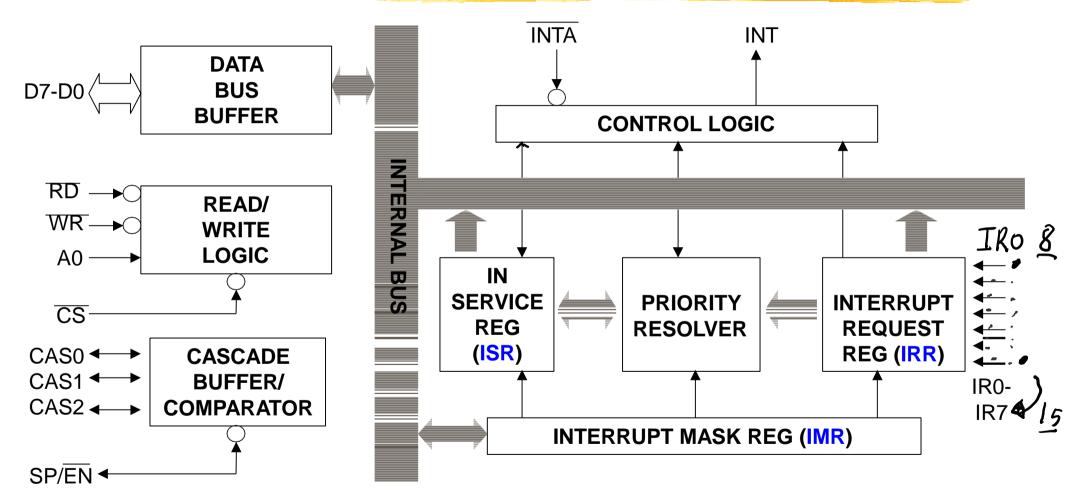
### Pinagem do 8259A

Symbol	Pin No.	Туре	Name and Function
V <sub>CC</sub>	28	I	SUPPLY: +5V Supply
GND	14	I	GROUND
CS	1	I	CHIP SELECT: A LOW on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the 8259A. INTA functions are independent of $\overline{CS}$ .
WR	2	I	WRITE: A LOW on this pin when CS is low enables the 8259A to accept command words from the CPU.
RD	3	I	READ: A LOW on this pin when CS is low enables the 8259A to release data onto the data bus for the CPU.
D <sub>7</sub> -D <sub>0</sub>	4 -11	I/O	BIDIRECTIONAL DATA BUS: Control, status and interrupt- vector information is transferred via this bus.
CAS <sub>0</sub> -CAS <sub>2</sub>	12,13,15	I/O	CASCADE LINES: The CAS lines form a private 8259A bus to control a multiple 8259A structure. These pins are outputs for a master 8259A and inputs for a slave 8259A.

### Pinagem do 8259A

SP/EN	16	I/O	SLAVE PROGRAM/ENABLE BUFFER: This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP - 1) or slave (SP - 0).
INT	17	0	INTERRUPT: This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR <sub>0</sub> -IR <sub>7</sub>	18-25	I	INTERRUPT REQUESTS: Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged ( <i>Edge Triggered Mode</i> ), or just by a high level on an IR input ( <i>Level Triggered Mode</i> ).
ĪNTA	26	I	INTERRUPT ACKNOWLEDGE: This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU
A <sub>0</sub>	27	I	AO ADDRESS LINE: This pin acts in conjunction with the CS, WR, and RD pins. It is used by the 8259A to decipher various Command Words the CPU writes and status the CPU wishes to read.

#### **Arquitetura Interna do 8259A**



Internal Block Diagram

#### Arquitetura Interna do 8259A

Data bus buffer: buffer bidirecional de três estados através do qual a MPU pode acessar os registos internos do 8259A.

R/W Logic: aceita sinais RD, WR, A0, e CS para controlar a direção, o tempo e a fonte/destino da transferência de dados.

IMR: usado para mascarar a entrada de pedidos de interrupção individuais (IR0-IR7).

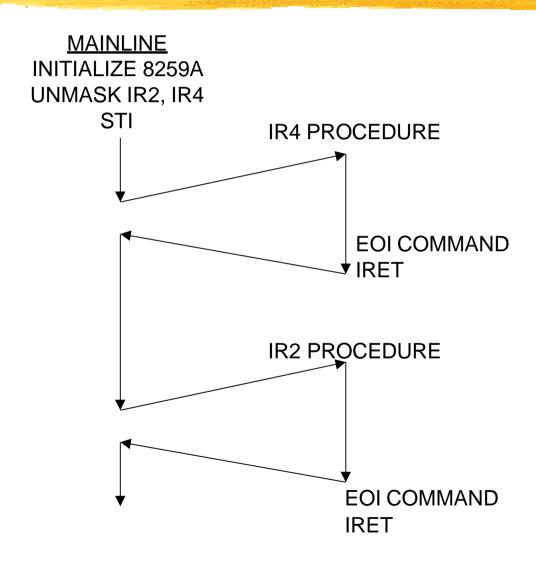
IRR: armazena o estado atual da entrada do pedido de interrupção.

Priority resolver: determina a prioridade da interrupção das entradas de interrupção ativas.

ISR: armazena a interrupção que está a sendo tratada.

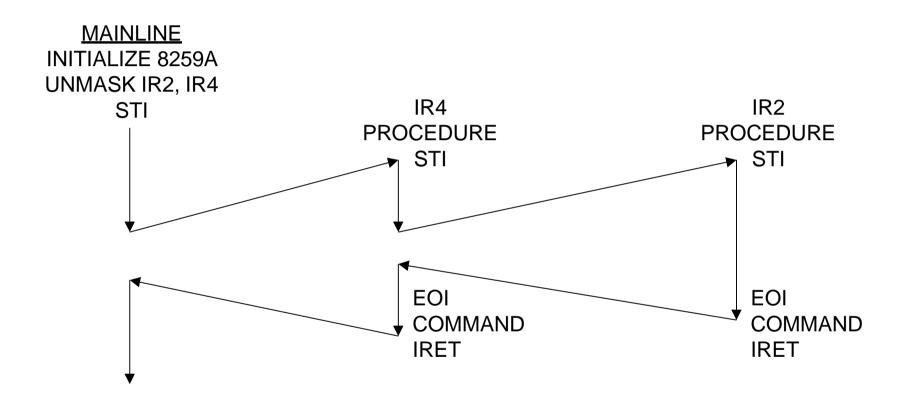
Cascade buffer/ comparator: fornece a interface entre o 8259A mestre e o 8259A secundário Cada 8259A secundário tem um código de identificação nesse subsistema.

### 8259A e 8088 Fluxo de programa Para IR4 Seguido de IR2



(b) Resposta com INTR não habilitada no procedimento IR4 (IF=0)

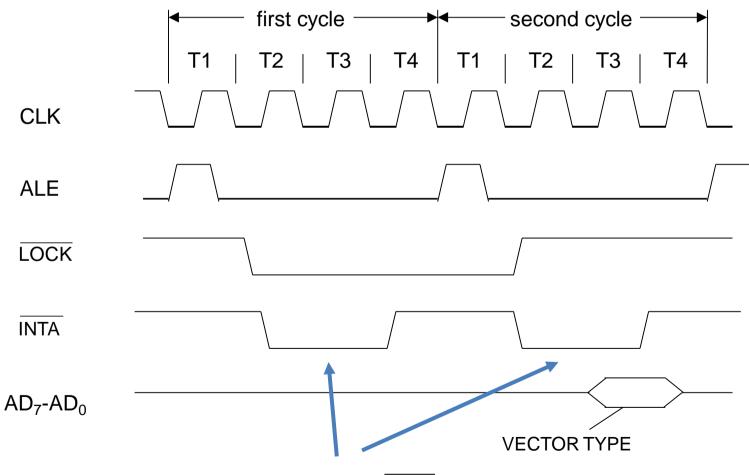
### 8259A e 8088 Fluxo de programa Para IR4 Seguido de IR2



(a) Resposta com INTR ativada no procedimento IR4 (IF=1)

#### Interrupt Acknowledge Bus Cycle

Assuma IF='1' (ou seja, aceitando INTR).

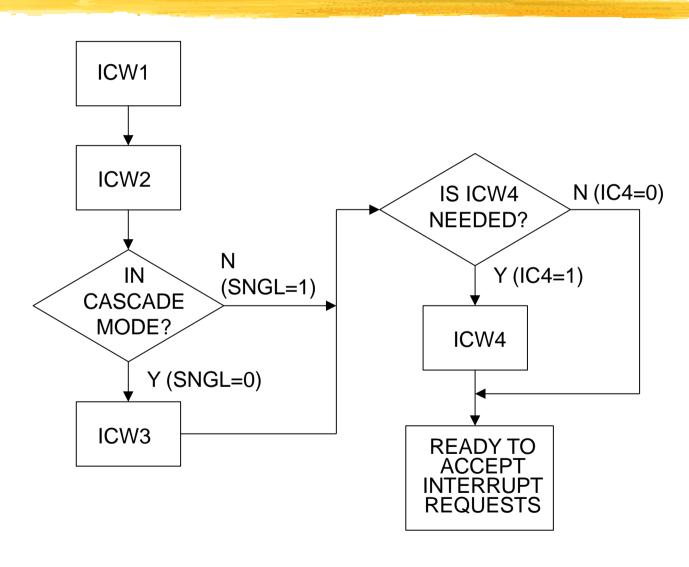


Observe a resposta com 2 sinais INTA = '0'

#### Programando o 8259A

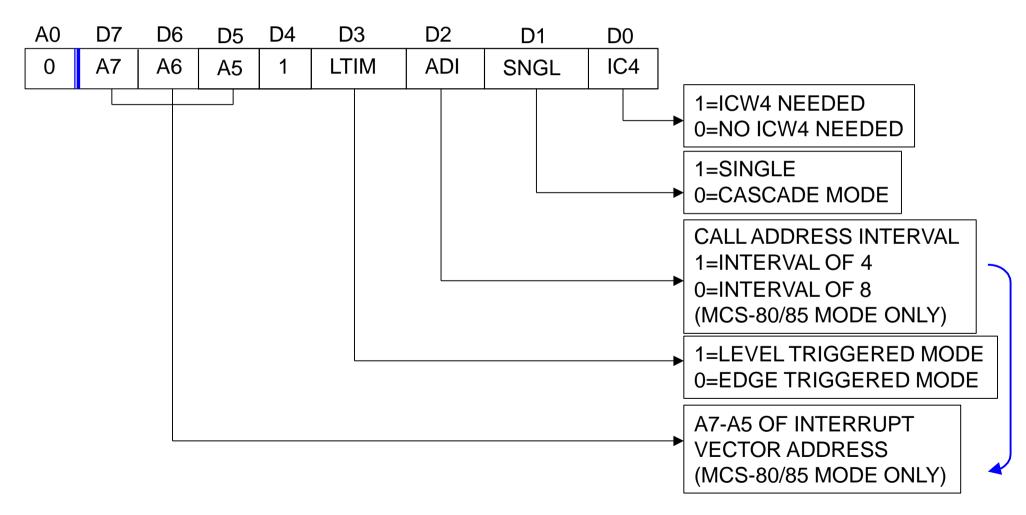
- Os CI's 8259A podem ser programados escrevendo bytes apropriados nos seus 7 registros internos (4 "palavras de comando de inicialização" registro ICW e 3 "palavras de comando operacional" registro OCW).
- Os comandos ICW são utilizados para carregar os registros de controle interno do 8259A.
- Os comandos OCW permitem ao 8088 iniciar variações nos modos de funcionamento básicos definidos pelo comando ICW.
- Os comandos ICW e OCW podem ser enviados ao 8259A utilizando a instrução OUT (para I/O mapeada) ou MOV (para memória mapeada).
- > A sequência de inicialização da ICW e OCW é mostrada a seguir.

## 8259A: Ordem de Envio de Palavra de Comando de Inicialização



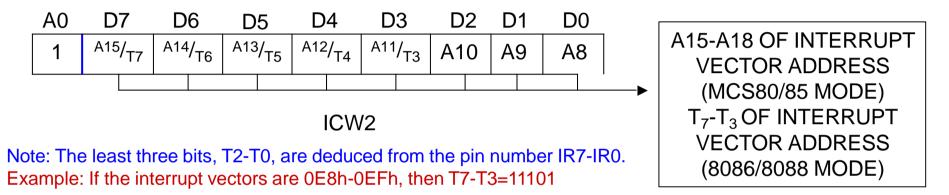
## 8259A: Palavra de Comando de inicialização ICW1

ICW1: usada para programar a operação básica do 8259a.

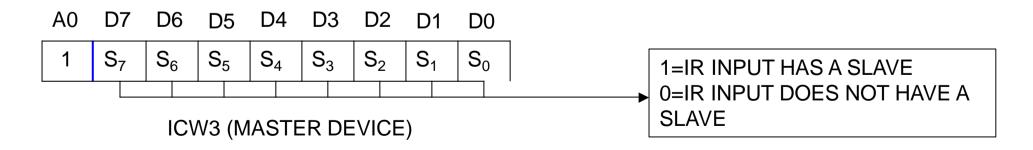


## 8259A: Palavra de Comando de inicialização ICW2 & ICW3 (master)

ICW2: used to select vector number used with the interrupt request inputs.

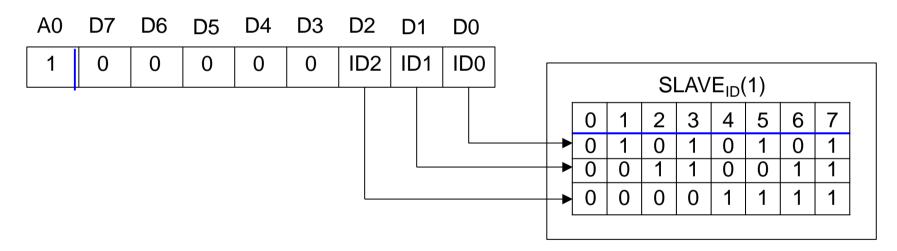


ICW3: used only when ICW1 specifies that the system is operated in cascade mode; used to indicate where the slave is connected to the master.



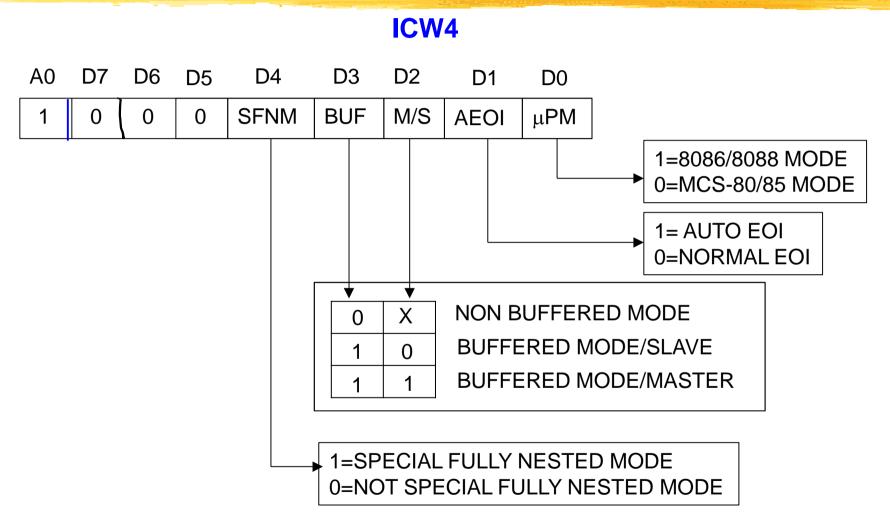
## 8259A: Palavra de Comando de inicialização ICW3 (secundário)

**ICW3 (SLAVE DEVICE)** 



NOTE 1: SLAVE ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT

## 8259A: Palavra de Comando de inicialização ICW4

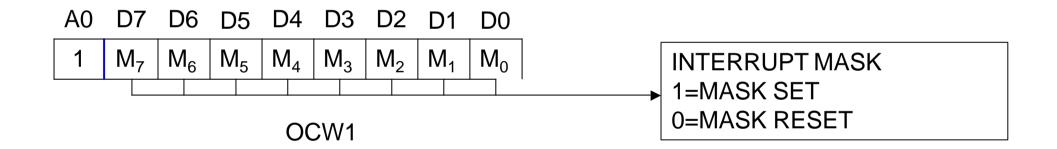


Special Fully Nested Mode: allows master PIC to accept request on a (master) IR input that is already in service

## 8259A: Palavras de Comando de modo de Operação OCW1

OCW1: is used to set and read the interrupt mask register

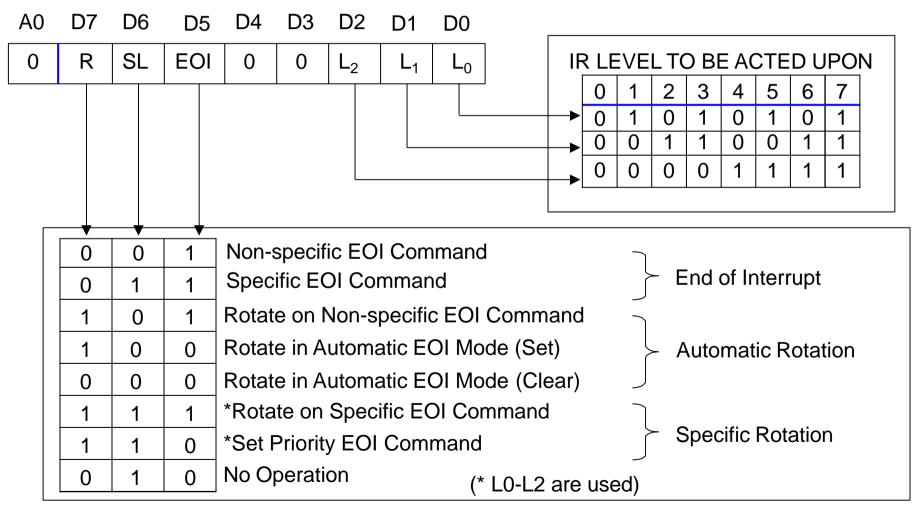
(Must be programmed after programming ICW)



For example, if IR0 and IR2 will be used → OCW1= 11111010

## 8259A: Palavras de Comando de modo de Operação OCW2

OCW2: to select how the 8259A responds to an interrupt.



### Nonspecific Priority Rotation

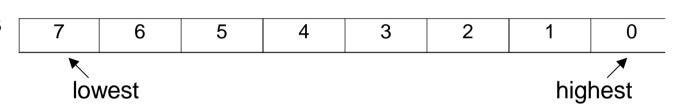
#### **Before Rotation**

(Assume IR4 has the highest priority)

(ISR)
IS status

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	1	0	0	0	0

Priority status

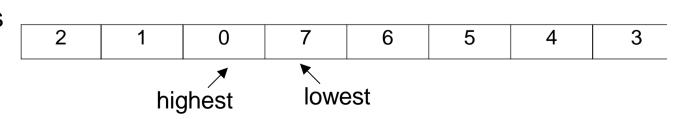


#### **After Rotation**

IS status

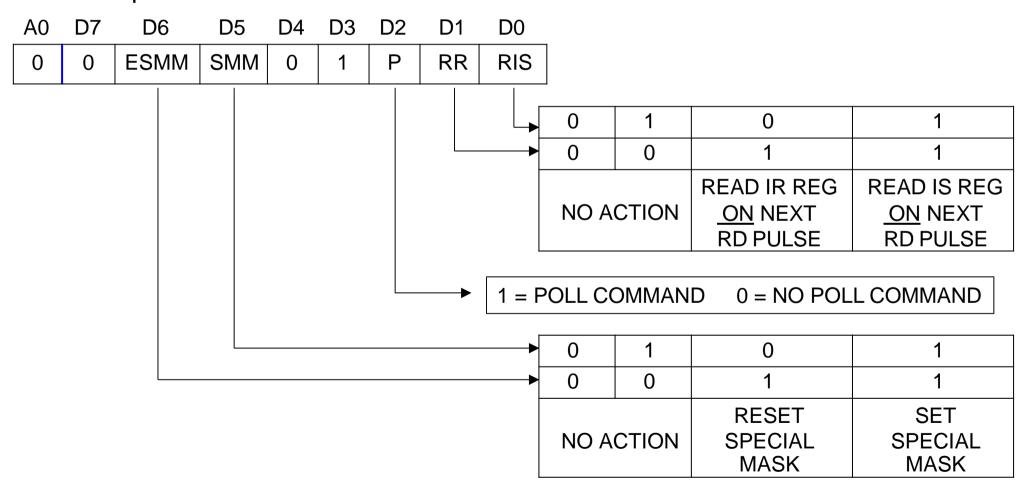
IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	0	0	0	0	0

Priority status



## 8259A: Palavras de Comando de modo de Operação O<del>C</del>W3

OCW3: to select the register to be read, the operation of the special mask register, and the poll command.



#### Examplos de Programação OCW

**Examplo:** Qual deve ser o código OCW1 se as entradas de interrupção IR0 até IR3 devem ser desativadas e IR4 até IR7 ativadas?

$$OCW1 = 000011111B = 0FH$$

**Examplo:** Que OCW2 deve ser emitido ao 8259A se o esquema de prioridade rodar em comando EOI não específico deve ser selecionado?

OCW2=<u>1010</u>0000B=A0H

#### **Teclado IBM-PC**

- O teclado se comunica com o computador através de uma interface serial síncrona própria.
- Cada tecla, quando acionada, gera um código que sinaliza quando foi pressionada (*make* da tecla) e um segundo código quando é liberada (*break* da tecla)
- Os códigos gerados não são ASCII.
- O teclado gera uma interrupção de hardware (9h) em cada uma das situações acima
- Associada à interrupção, o teclado envia serialmente o correspondente código (make ou break) para o computador.
- No computador, um subsistema digital recebe o código serial, remonta-o na sua forma paralela e o disponibiliza em uma porta de I/O (60H).
- Parte da gestão da comunicação é feita através do bit 7 da porta 61H.

#### **Teclado IBM-PC**

