

Interrupciones Externas

Programación de Sistemas Operativos

David Alejandro González Márquez

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Controlador de Interrupciones

- Una PC compatible posee mínimamente los siguientes dispositivos:
 - 8259 - Controlador de Interrupciones
 - 8254 - Timer Tick
 - 8042 - Controlador del Teclado

Controlador de Interrupciones

- Una PC compatible posee mínimamente los siguientes dispositivos:
 - 8259 - Controlador de Interrupciones
 - 8254 - Timer Tick
 - 8042 - Controlador del Teclado
- Tanto **Timer Tick** como el **Controlador del Teclado**, están conectados al Controlador de Interrupciones a IRQ0 y IRQ1 respectivamente.

Controlador de Interrupciones

- Una PC compatible posee mínimamente los siguientes dispositivos:
 - 8259 - Controlador de Interrupciones
 - 8254 - Timer Tick
 - 8042 - Controlador del Teclado
- Tanto **Timer Tick** como el **Controlador del Teclado**, están conectados al Controlador de Interrupciones a IRQ0 y IRQ1 respectivamente.
- En realidad, se poseen dos controladores de interrupciones conectados en cascada, mapeados a las interrupciones a partir del índice 08h

Configuración de PIC 8259

- Las interrupciones por hardware del procesador, **se pisan** con los índices seteados para los controladores de interrupciones.

Configuración de PIC 8259

- Las interrupciones por hardware del procesador, **se pisan** con los índices seteados para los controladores de interrupciones.
- Se deben **remapear** los controladores a un espacio de interrupciones designado a dispositivos de entrada/salida.

Configuración de PIC 8259

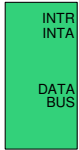
- Las interrupciones por hardware del procesador, **se pisan** con los índices seteados para los controladores de interrupciones.
- Se deben **remapear** los controladores a un espacio de interrupciones designado a dispositivos de entrada/salida.
- Para acceder a los PIC se usan las instrucciones “in” y “out”.

Configuración de PIC 8259

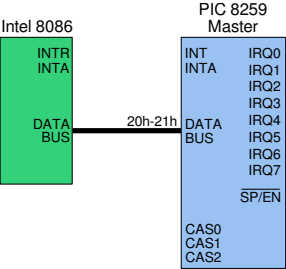
- Las interrupciones por hardware del procesador, **se pisan** con los índices seteados para los controladores de interrupciones.
- Se deben **remapear** los controladores a un espacio de interrupciones designado a dispositivos de entrada/salida.
- Para acceder a los PIC se usan las instrucciones “in” y “out”.
- Las direcciones 20h y 21h corresponden al PIC1, y las direcciones A0h y A1h al PIC2

Configuración de PIC 8259

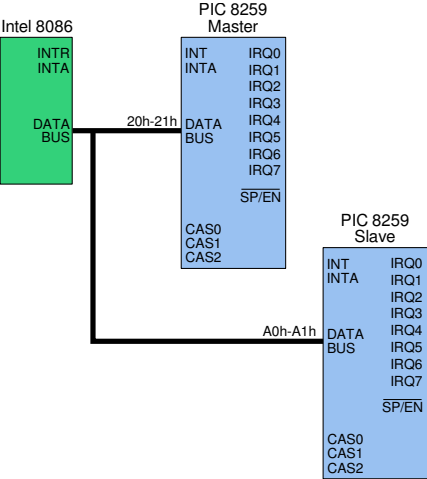
Intel 8086



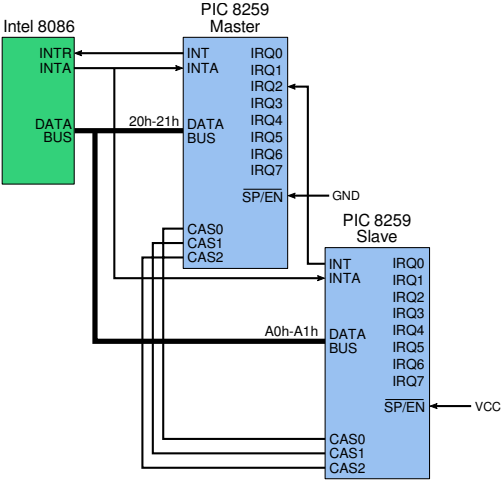
Configuración de PIC 8259



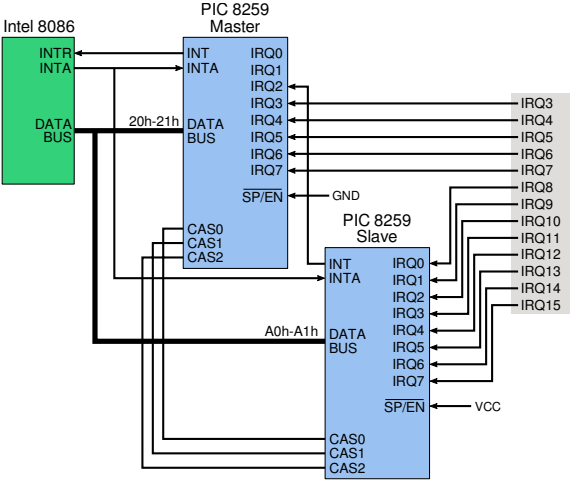
Configuración de PIC 8259



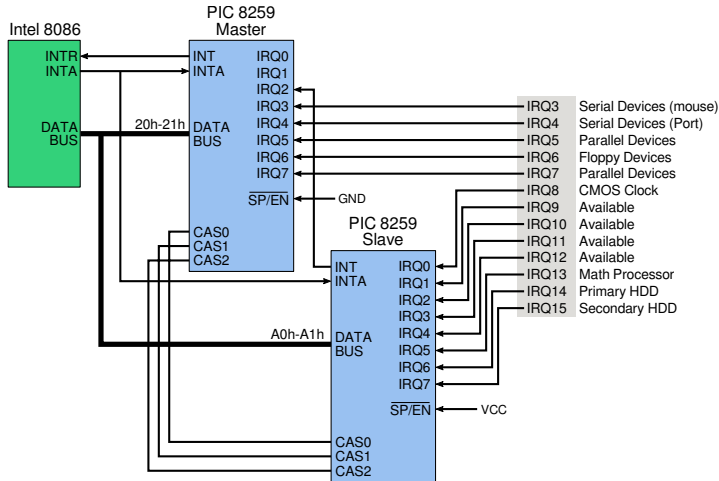
Configuración de PIC 8259



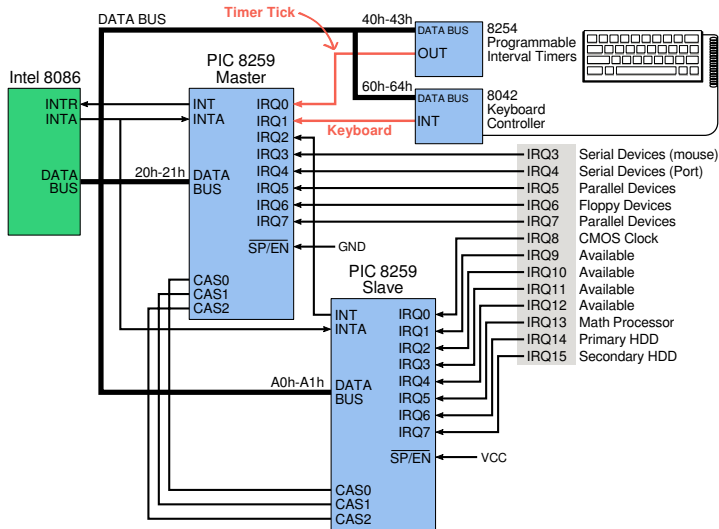
Configuración de PIC 8259



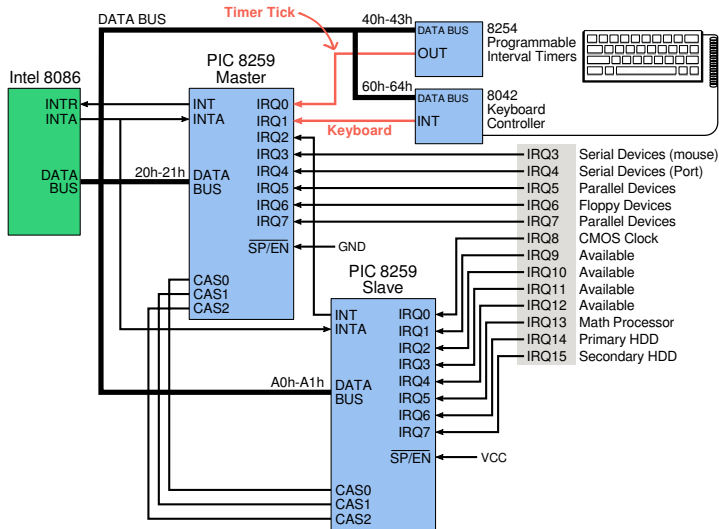
Configuración de PIC 8259



Configuración de PIC 8259



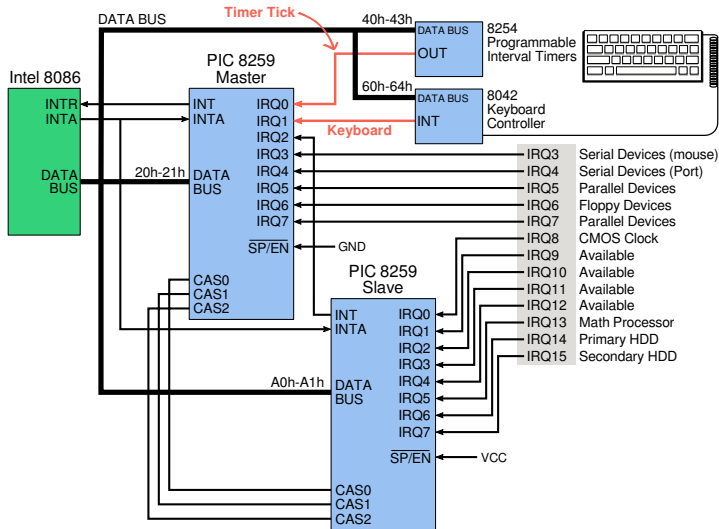
Configuración de PIC 8259



Protected Mode Interrupt Table

INT 0
...
INT 31
INT 32
INT 33
...
INT 47
INT 48
...
INT 255

Configuración de PIC 8259



Protected Mode Interrupt Table

INT 0	Excepciones del procesador	
...		
INT 31		
INT 32	IRQ0	Timer Tick
INT 33	IRQ1	Keyboard
	IRQ2	Reserve
	IRQ3	Serial Devices (mouse)
	IRQ4	Serial Devices (Port)
	IRQ5	Parallel Devices
	IRQ6	Floppy Devices
	IRQ7	Parallel Devices
	IRQ8	CMOS Clock
	IRQ9	Available
	IRQ10	Available
	IRQ11	Available
	IRQ12	Available
	IRQ13	Math Processor
	IRQ14	Primary HDD
	IRQ15	Secondary HDD
INT 47	Interrupciones de Software	
INT 48		
...		
INT 255		

Configuración de PIC 8259 - Palabras de Control

La configuración se realiza enviando palabras en el siguiente orden, por los puertos indicados.

Address	Read/Write	Function
20h y A0h	Write Write Write Read Read	Initialization Command Word 1 (ICW1) Operation Command Word 2 (OCW2) Operation Command Word 3 (OCW3) Interrupt Request Register (IRR) In-Service Register (ISR)
21h y A1h	Write Write Write Read/Write	Initialization Command Word 2 (ICW2) Initialization Command Word 3 (ICW3) Initialization Command Word 4 (ICW4) Interrupt Mask Register (IMR)

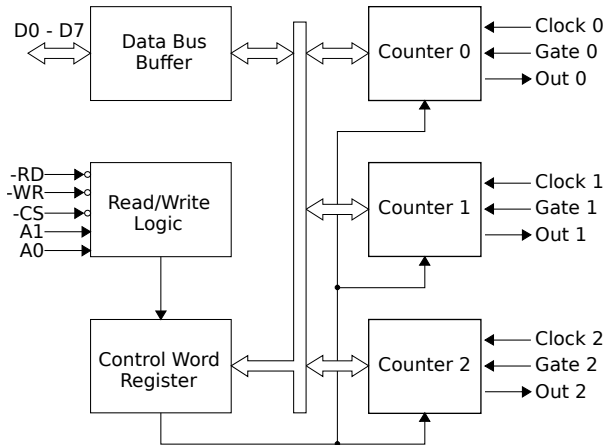
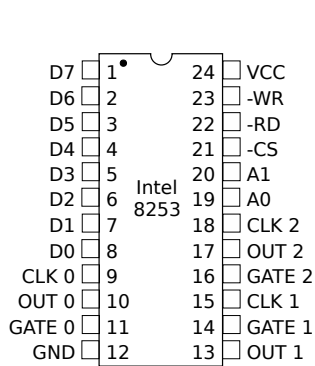
Configuración de PIC 8259 - Rutina de remapeo

```
; Inicializacion PIC1
mov al, 11h ;ICW1: IRQs activas por flanco, Modo cascada, ICW4 Si.
out 20h, al
mov al, 8 ;ICW2: INT base para el PIC1 Tipo 8.
out 21h, al
mov al, 04h ;ICW3: PIC1 Master, tiene un Slave conectado a IRQ2
out 21h, al
mov al, 01h ;ICW4: Modo No Buffered, Fin de Interrupcion Normal
out 21h, al ; Deshabilitamos las Interrupciones del PIC1
mov al, FFh ;OCW1: Set o Clear el IMR
out 21h, al
```

```
; Inicializacion PIC2
mov al, 11h ;ICW1: IRQs activas por flanco, Modo cascada, ICW4 Si.
out A0h, al
mov al, 70h ;ICW2: INT base para el PIC1 Tipo 070h.
out A1h, al
mov al, 02h ;ICW3: PIC2 Slave, IRQ2 es la linea que envia al Master
out A1h, al
mov al, 01h ;ICW4: Modo No Buffered, Fin de Interrupcion Normal
out A1h, al
```

Programmable Interval Timer (PIT) “El clock”

El integrado 8253/8254 tiene 3 contadores independientes. El registro de control (Control Word Register) permite programar cada uno de los tres contadores independientemente. Su tarea es **generar interrupciones** a intervalos regulares de tiempo (*temporizador*).



Keyboard controller

El integrado 8042 o controlador PS/2 permite controlar tanto el teclado como el mouse.

Keyboard controller

El integrado 8042 o controlador PS/2 permite controlar tanto el teclado como el mouse.

Uso

- Leemos teclas a través del puerto **0x60 (in al, 0x60)**
- Obtenemos un **scan code**

Keyboard controller

El integrado 8042 o controlador PS/2 permite controlar tanto el teclado como el mouse.

Uso

- Leemos teclas a través del puerto **0x60 (in al, 0x60)**
- Obtenemos un **scan code**

Scan code: Único para cada tecla

El teclado genera dos interrupciones:

Scancode=0xxxxxxx - **make**: Cuando se está presionando una tecla

Scancode=1xxxxxxx - **break**: Cuando se la está soltando una tecla

Keyboard controller

El integrado 8042 o controlador PS/2 permite controlar tanto el teclado como el mouse.

Uso

- Leemos teclas a través del puerto **0x60 (in al, 0x60)**
- Obtenemos un **scan code**

Scan code: Único para cada tecla

El teclado genera dos interrupciones:

Scancode=0xxxxxxx - **make**: Cuando se está presionando una tecla

Scancode=1xxxxxxx - **break**: Cuando se la está soltando una tecla

Ejemplo:

Presionar **a** → scancode **0x1E**

Keyboard controller

El integrado 8042 o controlador PS/2 permite controlar tanto el teclado como el mouse.

Uso

- Leemos teclas a través del puerto **0x60 (in al, 0x60)**
- Obtenemos un **scan code**

Scan code: Único para cada tecla

El teclado genera dos interrupciones:

Scancode=0xxxxxxx - **make**: Cuando se está presionando una tecla

Scancode=1xxxxxxx - **break**: Cuando se la está soltando una tecla

Ejemplo:

Presionar **a** → scancode **0x1E**

Presionar **b** → scancode **0x30**

Keyboard controller

El integrado 8042 o controlador PS/2 permite controlar tanto el teclado como el mouse.

Uso

- Leemos teclas a través del puerto **0x60 (in al, 0x60)**
- Obtenemos un **scan code**

Scan code: Único para cada tecla

El teclado genera dos interrupciones:

Scancode=0xxxxxxx - **make**: Cuando se está presionando una tecla

Scancode=1xxxxxxx - **break**: Cuando se la está soltando una tecla

Ejemplo:

Presionar **a** → scancode **0x1E**

Presionar **b** → scancode **0x30**

Soltar **a** → scancode **0x9E** ($0x1E + 0x80$)

Rutinas del PIC

El archivo `pic.h` contiene las rutinas para controlar el PIC.

- **pic_reset**: Remapea los PICs a indices despues de las excepciones del procesador.

Rutinas del PIC

El archivo `pic.h` contiene las rutinas para controlar el PIC.

- **pic_reset**: Remapea los PICs a índices después de las excepciones del procesador.
- **pic_enable**: Habilita los PICs para que generen interrupciones.
- **pic_disable**: Deshabilita los PICs para impedir que generen interrupciones.

Rutinas del PIC

El archivo `pic.h` contiene las rutinas para controlar el PIC.

- **pic_reset**: Remapea los PICs a índices después de las excepciones del procesador.
- **pic_enable**: Habilita los PICs para que generen interrupciones.
- **pic_disable**: Deshabilita los PICs para impedir que generen interrupciones.
- **pic_finish1**: Indica al PIC1 que fue atendida una interrupción.
- **pic_finish2**: Indica al PIC2 que fue atendida una interrupción.

Rutinas del PIC

El archivo `pic.h` contiene las rutinas para controlar el PIC.

- **pic_reset**: Remapea los PICs a índices después de las excepciones del procesador.
- **pic_enable**: Habilita los PICs para que generen interrupciones.
- **pic_disable**: Deshabilita los PICs para impedir que generen interrupciones.
- **pic_finish1**: Indica al PIC1 que fue atendida una interrupción.
- **pic_finish2**: Indica al PIC2 que fue atendida una interrupción.

Código para activar interrupciones externas

```
call pic_reset ; remapear PIC
call pic_enable ; habilitar PIC
sti           ; habilitar interrupciones
```

Una vez activado el PIC debemos indicarle al procesador que debe responder interrupciones externas. La instrucción **sti** setea el flag de interrupciones.

Rutinas de interrupciones

Una vez activadas, nos comenzarán a llegar interrupciones del reloj (**32**) y del teclado (**33**).

Rutinas de interrupciones

Una vez activadas, nos comenzarán a llegar interrupciones del reloj **(32)** y del teclado **(33)**.

Código para atender una interrupción externa

```
_isr:
    pushad
    ...
    call pic_finish1
    ...
    popad
    reti
```

Cuando atendemos una interrupción generada por el PIC.

Debemos notificarle que ya la atendimos. Para eso usamos rutina **pic_finish1**

Rutinas de interrupciones de ejemplo

Código ejemplo para atender el reloj

```
_isr32:
    pushad
    call pic_finish1 ; Indica que la interrupcion fue atendida
    call nextClock   ; Imprimir el reloj del sistema
    popad
    iret
```

Código ejemplo para atender el teclado

```
_isr33:
    pushad
    in al, 0x60      ; Captura una tecla
    push eax
    call printScanCode ; Rutina para imprimir el ScanCode
    add esp, 4
    call pic_finish1  ; Indica que la interrupcion fue atendida
    popad
    iret
```

Bibliografía: Fuentes y material adicional

- Convenciones de llamados a función en x86:
https://en.wikipedia.org/wiki/X86_calling_conventions
- Notas sobre System V ABI:
https://wiki.osdev.org/System_V_ABI
- Documentación de NASM:
<https://nasm.us/doc/>
- Artículo sobre el flag -pie:
<https://eklitzke.org/position-independent-executables>
- Documentación de System V ABI:
https://uclibc.org/docs/psABI-x86_64.pdf
- Manuales de Intel:
<https://software.intel.com/en-us/articles/intel-sdm>

¡Gracias!

Recuerden leer los comentarios al final de este video por aclaraciones o fe de erratas.