

### 3.- Unidad de entrada-salida

#### Introducción

- 1.- Dispositivos externos
- 2.- Controlador EIS
- 3.- EIS controlada por programa
- 4.- EIS " " interrupciones
- 5.- Acceso directo a memoria
- 6.- Procesador EIS

#### Introducción

Unidad EIS  $\Rightarrow$  proporciona un método de comunicación entre sistema central y usuario.

Motivos de NO conexión directa CPU-Periféricos |  
- Gran variedad de periféricos y modos operación  
- Velocidad lenta de los periféricos  
- Formatos datos diferentes entre CPU y periféricos

Funciones EIS |  
- Comunicarse con CPU y Mem. por los buses  
- " " con los periféricos

Técnicas interacción de EIS y sistema |  
- EIS controlada por programa  
- " por IRQs  
- Acceso directo a memoria (DMA)

## 1.- Dispositivos externos (periféricos)

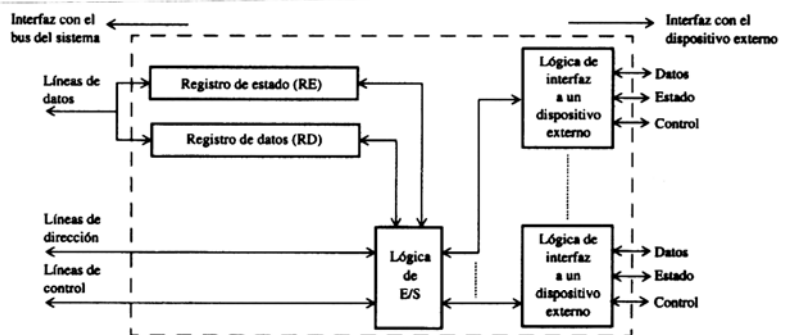
- Categorías
- Adaptados al usuario
  - " a la máquina
  - De comunicación
  - De interacción con el medio externo

## 2.- Controlador E/S

- Funciones
- Control y temporización
  - Comunicación con la CPU
  - " con el dispositivo externo
  - Almacen temporal de datos
  - Detección de errores

- Requerimientos de comunicación con la CPU
- Decodificación de la orden
  - Datos
  - Información sobre el estado
  - Reconocimiento de la información

## Estructura del controlador de E/S



# Estructura del sistema de E/S

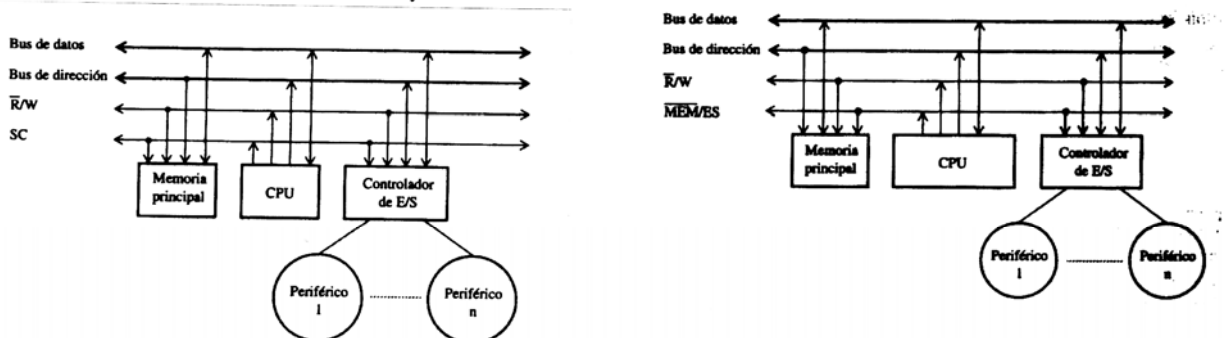
- Modos de interconexión
- Buses independientes  $\Rightarrow$  E/S aislada
  - Bus común pero líneas control separadas
  - Bus único  $\Rightarrow$  E/S localizada en memoria
- E/S  $\leftrightarrow$  CPU

## 3.- E/S controlada por programa

La CPU está pendiente del estado de la E/S ejecutando un programa de sondeo.

- Ordenes de E/S
- Ordenes de control  $\rightarrow$  decirle que hacer
  - " de comprobación  $\rightarrow$  ver su estado
  - " de lectura
  - " de escritura

- Instrucciones de E/S
- E/S localizada en mem  $\Rightarrow$  mismas instrucciones que acceso a mem.
  - E/S aislada  $\Rightarrow$  Instrucc específicas } IN OUT



#### 4.- E/S por interrupciones

- Inconvenientes  
E/S por programa
- Pérdida tiempo en bucle espera
  - Algunos programas no permiten estar en bucle
  - Si varios periféricos  $\Rightarrow$  muchos problemas

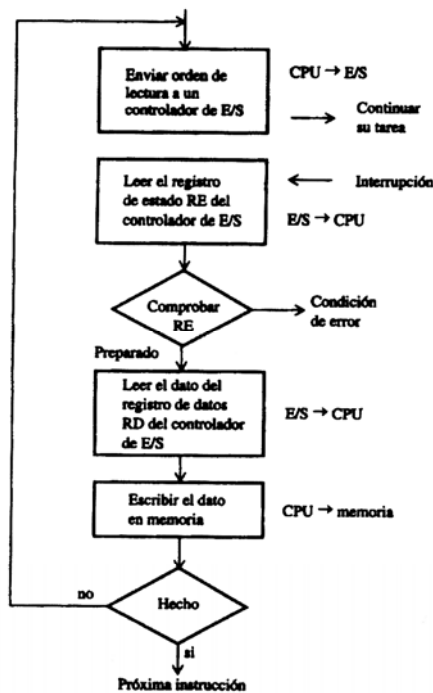
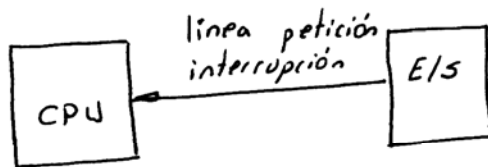
Interrupción  $\Rightarrow$  eliminar bucles de espera.



Cuando periférico preparado para intercambiar información fuerza una interrupción en la tarea de la CPU para atender a la E/S, una vez transferido, CPU continúa con la tarea.



Entre CPU y E/S (Periférico)  $\Rightarrow$  línea de petición de interrupción PI



Cuando periférico activa PI, la mantiene activada hasta que se atiende su petición



PI activada durante respuesta a la interrupción



Durante respuesta a interrupción inhabilitada

Métodos {

- La propia CPU ignora la PI cuando responde al programa con la 1ª instrucción
- El propio sistema tiene un mecanismo de enmascaramiento de interrupciones

### Tratamiento de una interrupción:

- 1º) Activar sistema de interrupciones
- 2º) Periférico activa PI
- 3º) CPU termina instrucción en curso, salva el PC y otros registros usados por el programa
- 4º) CPU inhibe interrupciones (más caras) y carga en el PC el vector interrupción (dirección de localización del programa de respuesta).
- 5º) Se informa a periférico de fin de respuesta (línea PI) para que desactive PI
- 6º) Se reactiva sistema de interrupciones.
- 7º) Se recupera el PC y los registros ⇒ se continúa con la ejecución del programa.

Clasificación  
de las  
interrupciones  
JRQs

- Origen
  - Externa (Periféricos)
  - Interna (La propia CPU)
- N° líneas
  - 1 línea
  - Múltiples líneas
- Control de la CPU sobre la JRQ
  - Enmascarables
  - No enmascarables
- "Identificación" de Fuente de la interrupción
  - Múltiples líneas
  - Encuesta
  - Vectorizadas
- Gestión de la prioridad
  - Por software
  - Por hardware
- Niveles de interrupción
  - Nivel único
  - Multinivel

Origen

- Externas
  - Generada por periférico
  - Fallo alimentación
  - Señal de reloj
  - Acceso a mem. no permitido
  - Sistemas multiprocesadores
- Internas
  - Desbordamiento de la ALU
  - Código operación desconocido
  - Interrupción de rastreo
  - Interrupción de programa.

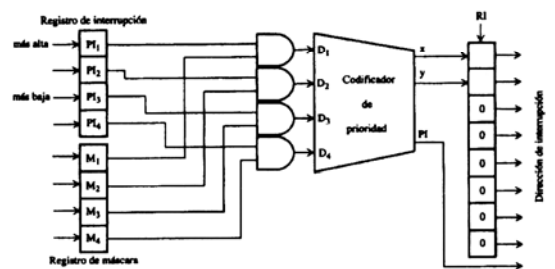
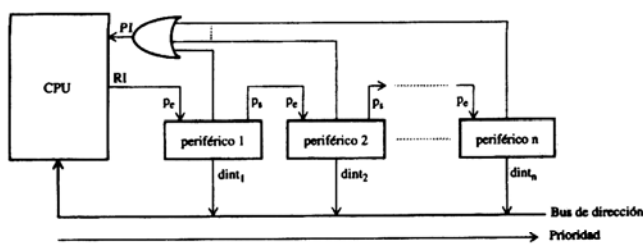
Nº de líneas de interrupción } - Varias líneas  
 - Única línea para varios periféricos

Control de la CPU sobre la interrupción } - Enmascaramiento individual  $\Rightarrow$  Reg. máscara de IRAs  
 - Enmascar. por nivel

Identificación de la fuente de la interrupción y gestión de la prioridad } - Identif. mediante encuesta  $\Rightarrow$  Única línea y cuando hay IRQ se encuesta a los periféricos  
 - Mediante encadenamiento  $\Rightarrow$  Encuesta por hard.  $\hookrightarrow$  Interrup. vectorizadas  
 - Mediante hard. paralelo  $\Rightarrow$  Usa Reg. interrupción + Reg. máscara

Encadenadas

Hardware

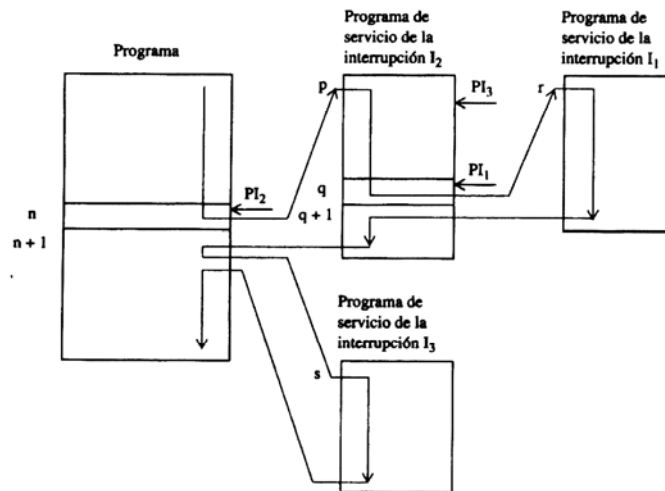


Cuando una IRQ se produce se avisa a la CPU; ésta devuelve señal por RI y el primer perif. con IRQ bloquea la señal y envía vector interrupción

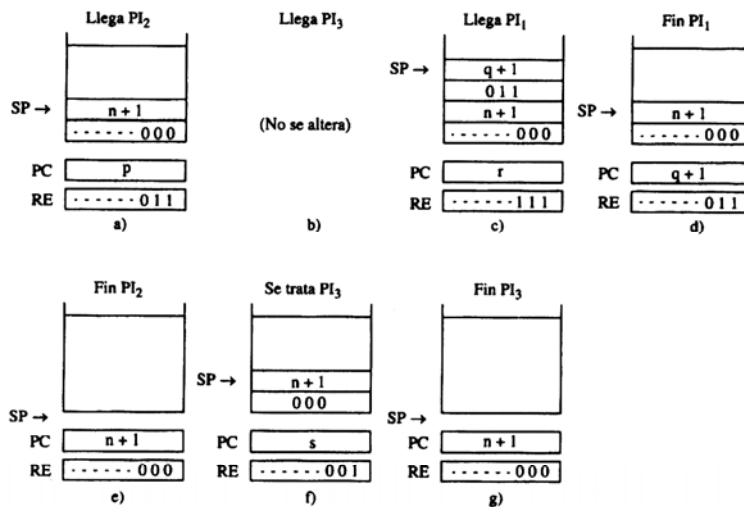
# Niveles de interrupción

Hasta ahora supuesto que si atendiendo a una interrupción aparece otra no se atiende hasta terminar la 1ª. (JRA)  
 ↪ UNICO NIVEL ↪

Anidamiento ⇒ atendiendo a una IRQ si aparece otra de mayor prioridad, se deja de ejecutar la 1ª, se atiende a la 2ª y cuando termina ésta sigue con la 1ª  
 ↪ MULTINIVEL ↪



Evolución de la pila y de los registros de la CPU

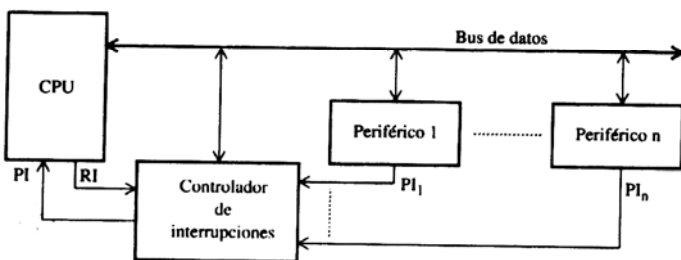
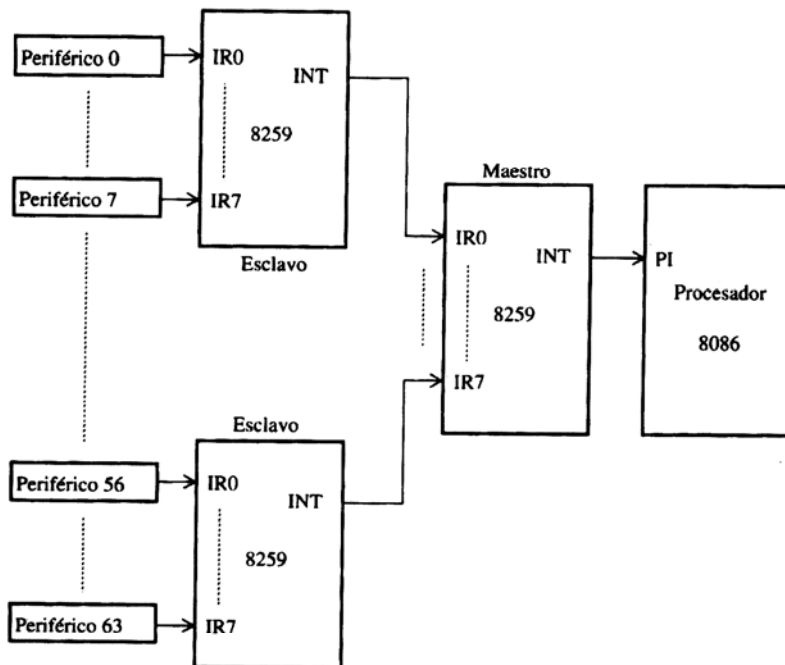




## Controlador de interrupciones

↳ Ampliar el n° de líneas y encargarse de la gestión.

- Funciones
- Identificar la fuente de la interrupción
  - Establecer las prioridades de cada periférico
  - Activar/desactivar IRQs
  - Enviar información a la CPU sobre la petición de interrupción y el periférico a atender.

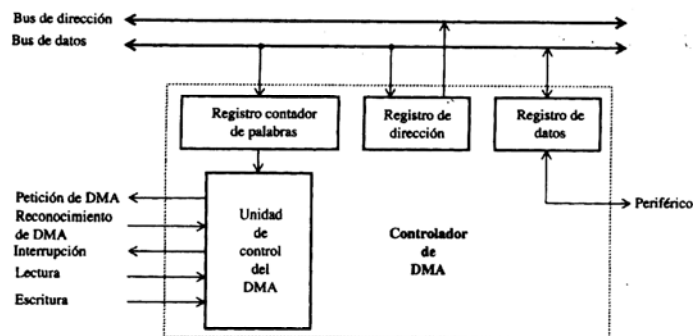


- Modos
- Onidadas
  - En rotación
  - Enmascarables

## 5.- Acceso Directo a Memoria (DMA)

- Desventajas  
EIS por  
JRQ
- La transferencia de datos limitada por la velocidad de la CPU
  - La CPU gestiona la EIS  $\Rightarrow$  pérdida rendimiento

Grandes transferencias de datos  $\rightarrow$  Controlador DMA



Controlador DMA transfiere directamente (sin pasar por CPU) palabra a palabra el bloque completo de datos entre periférico y memoria. Participando la CPU solo al principio y final de la transmisión

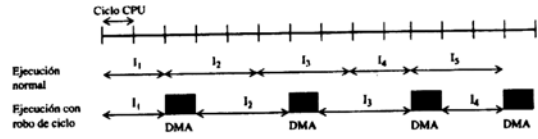
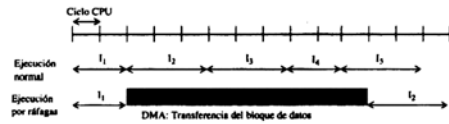
- Información  
enviada a la  
DMA para  
empezar la  
transmisión
- Si operación es lectura o escritura
  - Dirección del periférico
  - Posición inicial de la memoria
  - N<sup>o</sup> de palabras a leer o escribir

## Modos transferencia

- Por ráfagas  
↳  
Para a la CPU

- Robo de ciclos

- Transparente

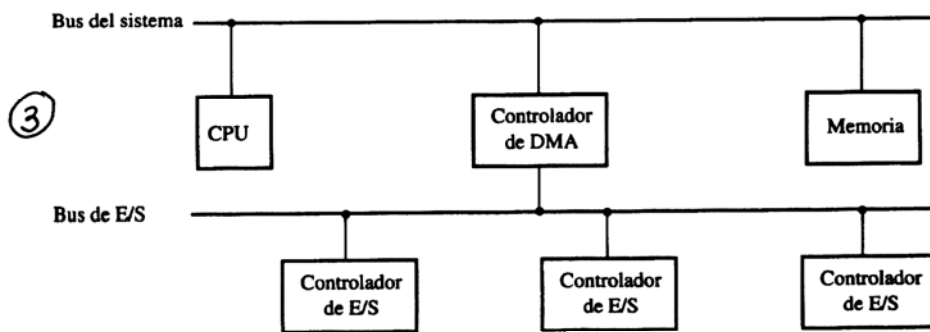
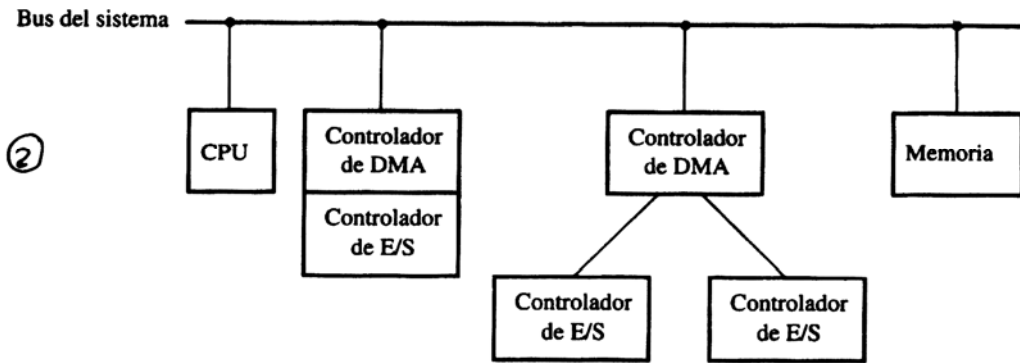
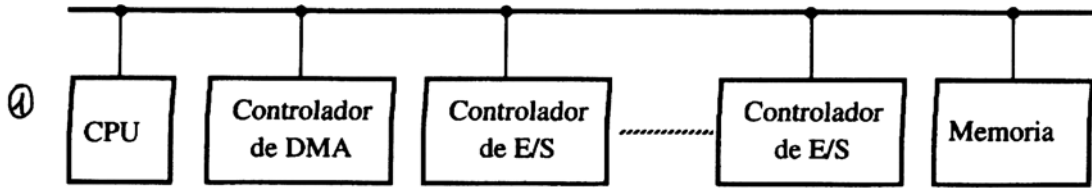


## Pasos en una transferencia:

- 1º La CPU carga en la DMA
  - Direc. memoria
  - " periférico
  - Nº palabras
- 2º Cuando DMA preparada, activa la línea de petición de DMA y espera que CPU libere el bus
- 3º DMA transfiere datos
- 4º Si contador palabras  $\neq 0$  espera próxima liberación
- 5º Si contador palabras = 0, DMA abandona la transmisión

# Configuración de la DMA

- Conexión DMA
- Bus único con DMA conectado ①
  - " " " " integrados ②
  - Controlador DMA conectado a bus E/S ③

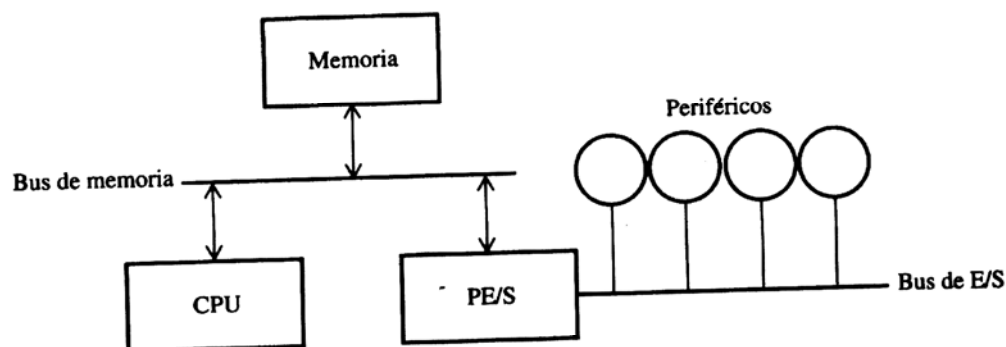


## 6.- Procesador de E/S (PEIS)

### Historia control periféricas:

- 1º CPU controla periférico  $\Rightarrow$  sistemas simples
- 2º Incorpora controlador E/S  $\Rightarrow$  mejora rendimiento  
 $\hookrightarrow$  ctrl por programa.
- 3º " " E/S pero controlado por interrupciones.
- 4º DINA
- 5º Se potencia controlador E/S convirtiéndolo en un procesador
- 6º Se le incorpora memoria  $\Rightarrow$  se convierte en microcontrolador

En el PEIS el microcontrolador ejecuta las instrucciones, lo que supone que libera más a la CPU



Clasificación de los PEIS

- Canal selector
- Canal multiplexor

