

# Introducción a las E/S del sistema

- Interfaz entre dispositivos periféricos y el  $\mu$ procesador.
  - Diferencia en velocidad de funcionamiento (tasa de transferencia).
  - Diferencia en las unidades de información.
  - Diferencia en los modos de funcionamiento.
- Dispositivos mas comunes:
  - Adaptador para comunicación paralelo PPI o PIA.
  - Adaptador para comunicación serie USART.
  - Controlador de interrupciones PIC.
  - Controlador de tiempos TIMER.
  - Controlador de acceso directo a memoria DMA.

## Manipulación de periféricos

- Procedimientos para la manipulación de los periféricos:
  - *Por software* : Proceso de *consultas sucesivas*. Bajo rendimiento de la CPU. Proceso síncrono. Consulta por *Polling*.
  - *Mediante interrupciones* : El dispositivo provoca una interrupción a la CPU. Posibilidad de asignar *prioridades* y *enmascaramiento*. Esta gestión la realiza el controlador de interrupciones (PIC).
  - *Acceso Directo a Memoria* : Utilizando la señal HOLD. Transferencia directa entre memoria y periférico (para grandes bloques de información). El  $\mu$ procesador se desconecta de los buses.
    - Ejemplo: Disco duro < > Memoria Principal

# Esquema General de un dispositivo de E/S

## ■ Adaptador de E/S paralelo:

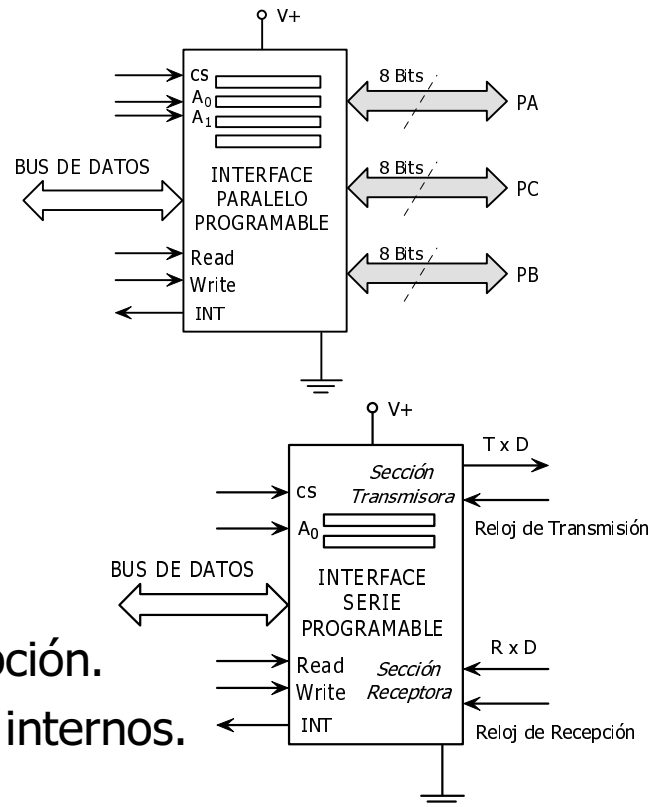
- - Resuelve la adaptación del sistema a periféricos que trabajan con información en paralelo.
- - Dispositivo programable.

## ■ Adaptador de E/S serie:

- - Serialización del dato a transmitir.
- - Paralelización del dato recibido.
- - Gestión de errores en la transmisión del dato.

## ■ Disponen de peticiones de interrupción.

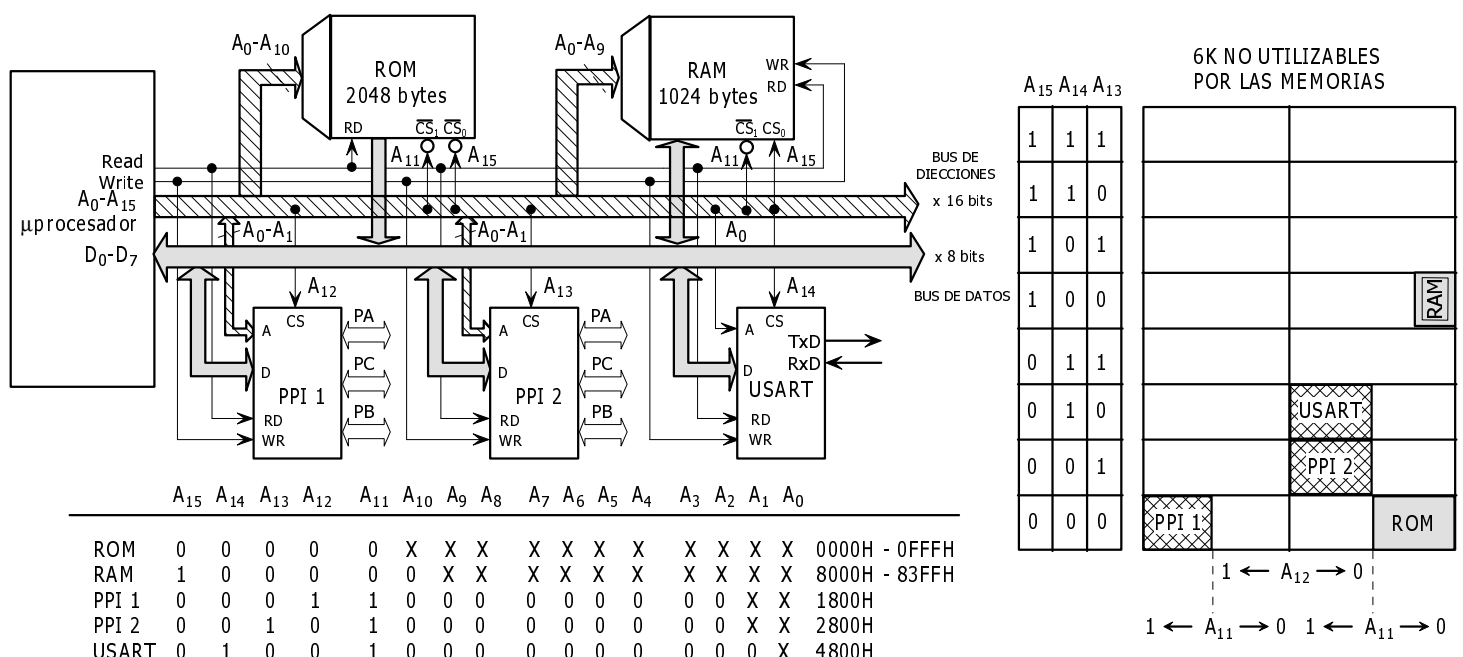
## ■ Entradas de selección de registros internos.



## I. Ejemplo de interconexión

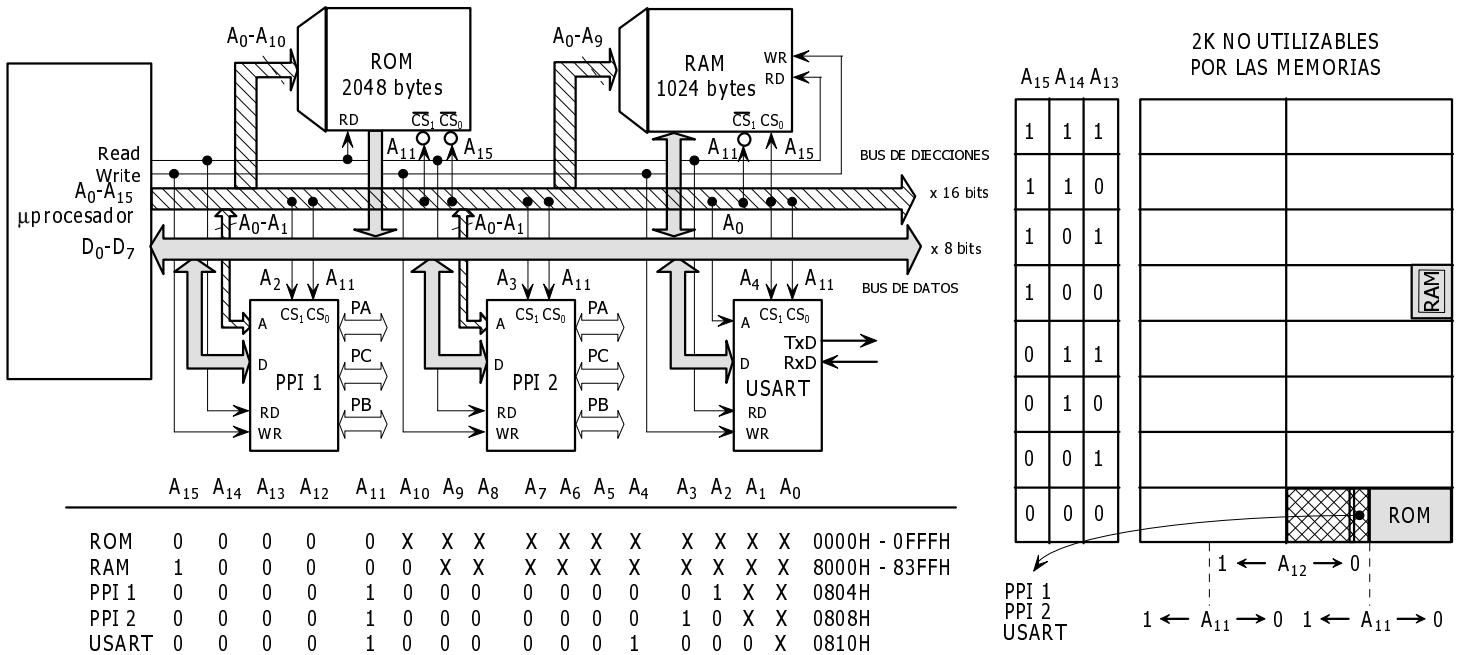
### ■ Interconexión con el $\mu$ procesador: selección lineal.

- Ejemplo: Sistema con 2K de ROM, 1K de RAM, 2 PPI, 1 USART.



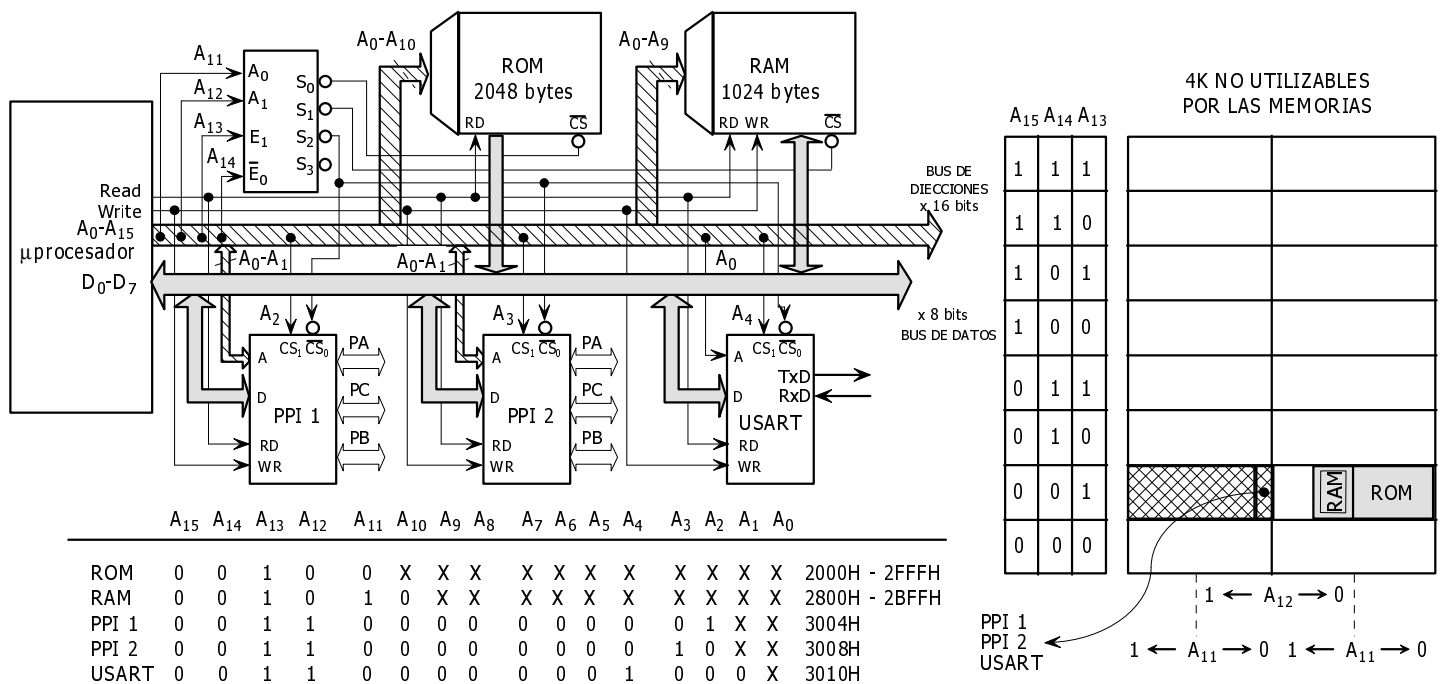
## II. Ejemplo de interconexión

### ■ Optimización del mapa de memoria (selección lineal).



## III. Ejemplo de interconexión

### ■ Optimización del mapa de memoria (selección por decodificación).



# Introducción a las E/S del sistema

## ■ Estructura de E/S por instrucciones de memoria:

- Las transferencias de E/S periféricas se efectúan por instrucciones de memoria. Las señales *Read* y *Write* sirven tanto para las memorias como para los interfaces de E/S.

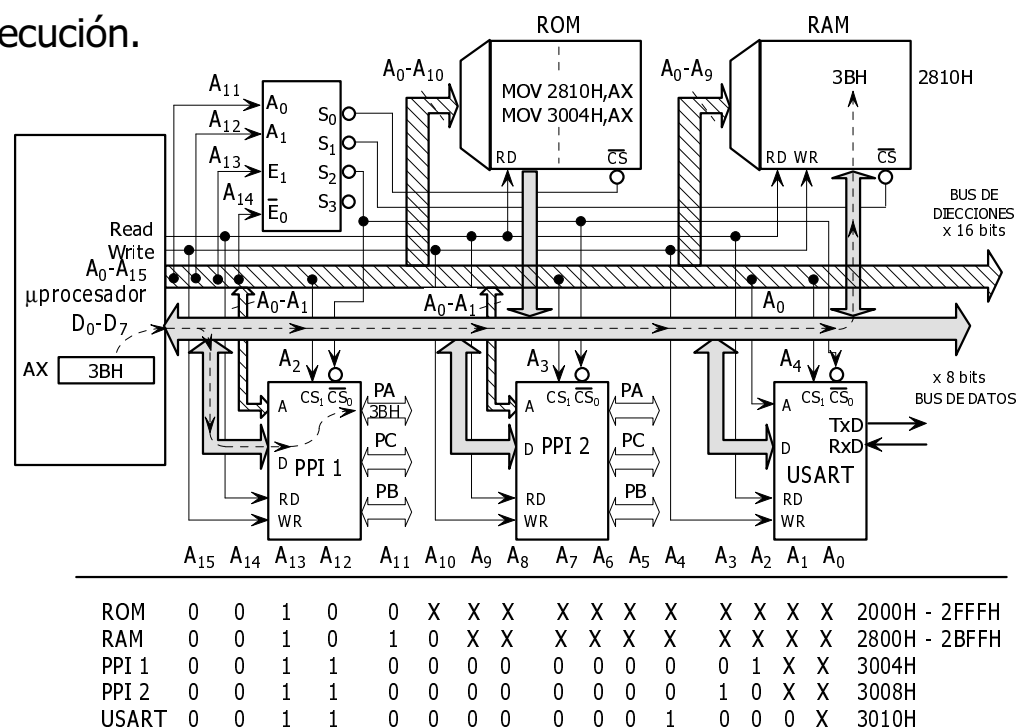
## ■ Estructura de E/S por instrucciones de E/S:

- Las transferencias son efectuadas por instrucciones específicas de los periféricos.
- En esta estructura, el mapeado de una memoria y de un interface de E/S, pueden coincidir (señales de *Read* y *Write* diferentes para las memorias y para los interfaces de E/S).
- El espacio de memoria está totalmente disponible para la memoria. En este caso, la cantidad máxima posible de memoria, coincide con el espacio de memoria del sistema.

# Introducción a las E/S del sistema

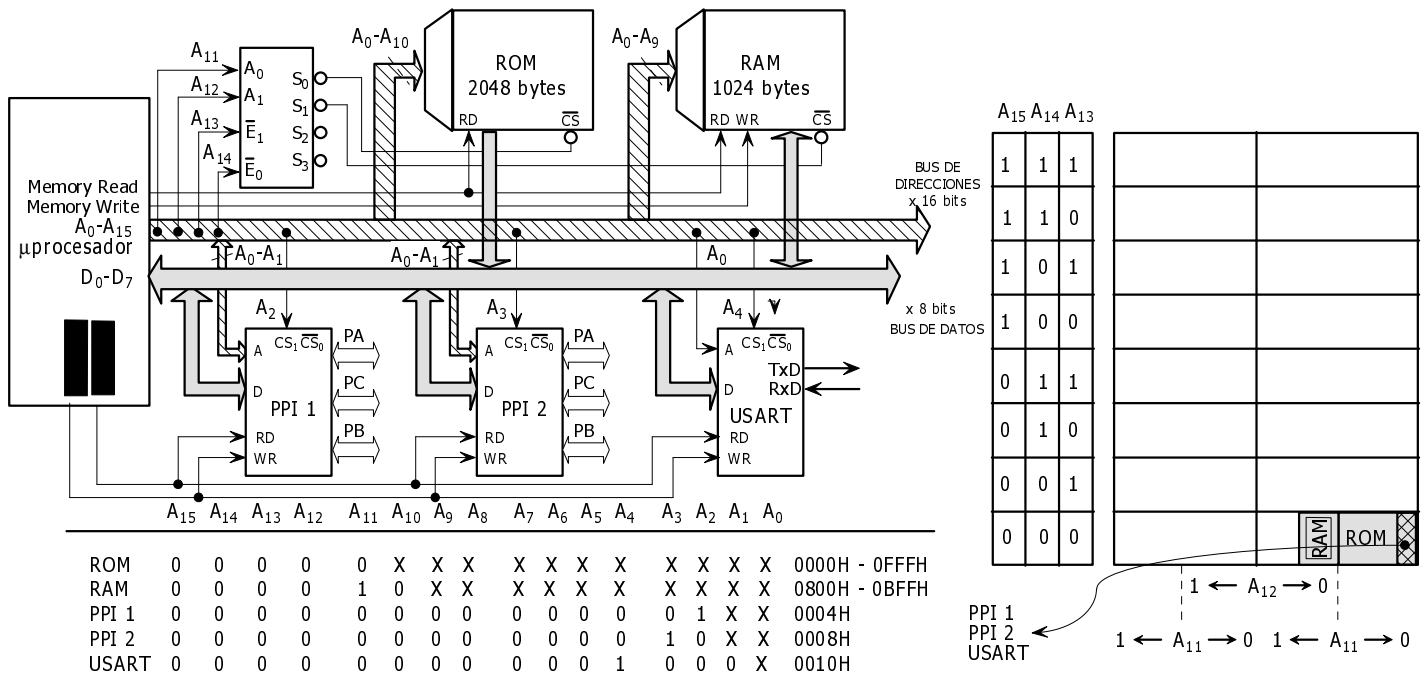
## ■ Estructura de E/S por instrucciones de memoria:

- Ejemplo de ejecución.



# Introducción a las E/S del sistema

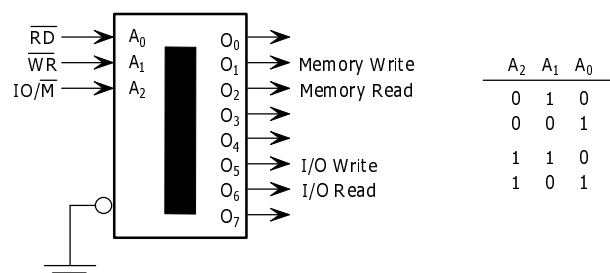
## ■ Estructura de E/S por instrucciones de E/S:



# Introducción a las E/S del sistema

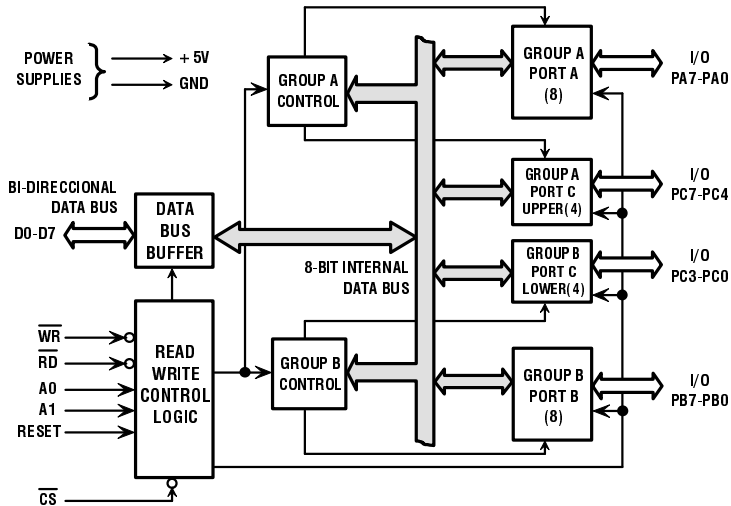
## ■ Generación de las señales Memory Read, Memory Write, I/O Read e I/O Write:

- Las genera directamente el  $\mu$ procesador (ejemplo,  $\mu$ p. Intel 8080).
- El  $\mu$ procesador genera solamente Read y Write (tanto para Memoria como para E/S):
  - l En este caso el  $\mu$ procesador genera una señal ( $IO/\overline{M}$  en el caso de Intel) indicando si el acceso es a Memoria o a E/S.
  - l A.- Uso de la señal  $IO/\overline{M}$  para la selección de las memorias e interfaces de E/S.
  - l B.- Uso de la señal  $IO/\overline{M}$  para genera las señales de Read y Write para memoria y E/S.



# Interfaz Paralelo Programable 82C55A

## ■ Estructura del PPI:



- Dispone de 3 puertos de E/S paralelos denominados A, B y C, con salidas latcheadas.
- El puerto C está dividido en dos grupos de 4, que pueden trabajar de forma combinada con las puertas A y B en determinados modos.
- Internamente tiene 3 registros de datos y un registro de control para configurar el funcionamiento del dispositivo.
- Admite tres modos diferentes de funcionamiento, denominados Modo 0, Modo 1 y Modo 2.

## 82C55A: Tabla de verdad

- Sentido de las transferencias de información entre el bus de datos del sistema y los registros internos del PPI:

| A <sub>1</sub> | A <sub>0</sub> | $\overline{RD}$ | $\overline{WR}$ | $\overline{CS}$ | FUNCIÓN                      |
|----------------|----------------|-----------------|-----------------|-----------------|------------------------------|
| 0              | 0              | 0               | 1               | 0               | Puerta A -> Bus de Datos     |
| 0              | 1              | 0               | 1               | 0               | Puerta B -> Bus de Datos     |
| 1              | 0              | 0               | 1               | 0               | Puerta C -> Bus de Datos     |
| 0              | 0              | 1               | 0               | 0               | Bus de Datos -> Puerta A     |
| 0              | 1              | 1               | 0               | 0               | Bus de Datos -> Puerta B     |
| 1              | 0              | 1               | 0               | 0               | Bus de Datos -> Puerta C     |
| 1              | 1              | 0               | 1               | 0               | ILEGAL                       |
| 1              | 1              | 1               | 0               | 0               | Bus de Datos -> Reg. Control |
| X              | X              | X               | X               | 1               | Dispositivo en Triestado     |

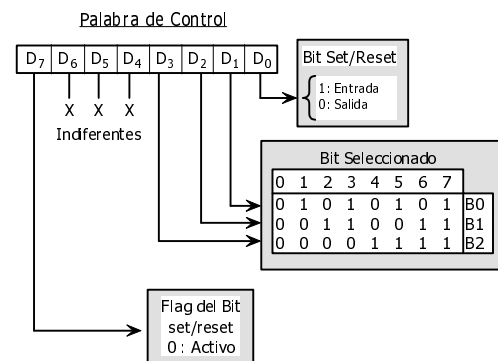
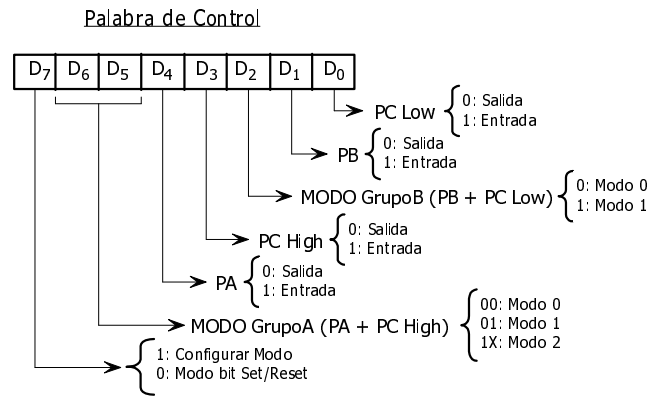
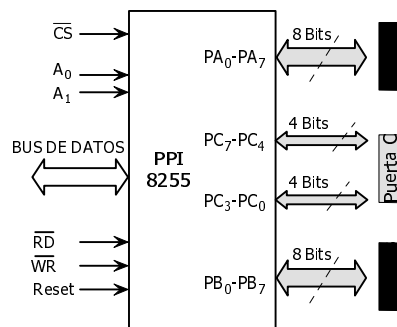
- Las líneas A<sub>0</sub> y A<sub>1</sub> determinan la dirección de los puertos.

- Ejemplo: Dirección de base -> 3000H
  - PA: 3000; PB: 3001; PC: 3002; Reg. Control: 3003H

# PPI 82C55A: Programación

## ■ Programación del dispositivo:

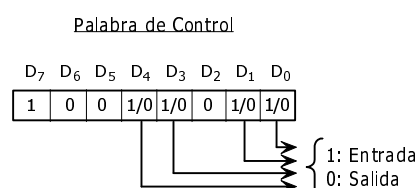
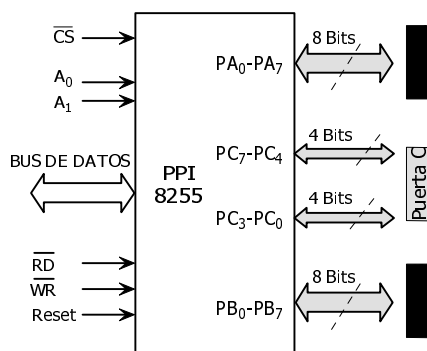
- Registro de Control: Configura la dirección y modo de funcionamiento de cada una de las puertas.
- Modo bit Set/Reset: Permite sacar 1 ó 0 por cualquiera de las líneas del puerto C.



## PPI: Funcionamiento en Modo 0

### ■ Modos de funcionamiento:

- Modo 0: Corresponde a la forma más básica de funcionamiento.
  - Cada puerta se configura como entrada o salida en el registro de control.
  - Salidas latcheadas y entradas no latcheadas.
  - No hay señales de diálogo con los periféricos.

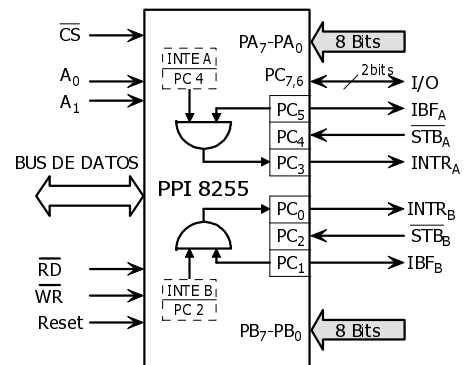


# PPI: Funcionamiento en Modo 1 (Entrada)

- Modo 1: Transferencia con *Handshaking*. Las Puertas A y B son controladas por las líneas de la puerta C (3 líneas por puerta).

## Modo de Entrada:

- Puerta B: Controlada por los bits PC<sub>0</sub> a PC<sub>3</sub>.
- Puerta A: Controlada por los bits PC<sub>3</sub> a PC<sub>5</sub>.
- PC<sub>6</sub> y PC<sub>7</sub>, pueden configurarse como líneas convencionales de E/S.
- Señales del protocolo de entrada: IBF, STB e INTR.
- Habilitación de las interrupciones a través de los biestables internos INTEA (PC<sub>4</sub>) e INTEB (PC<sub>2</sub>).



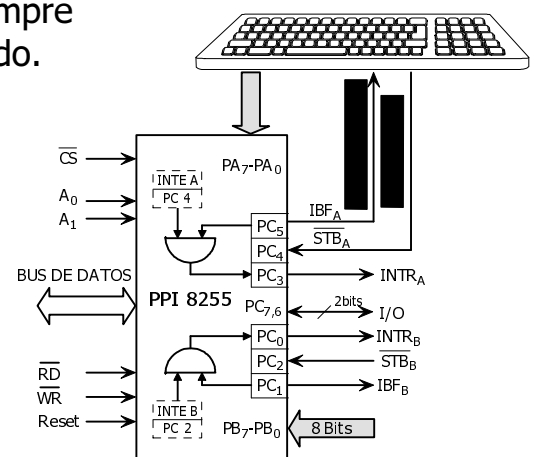
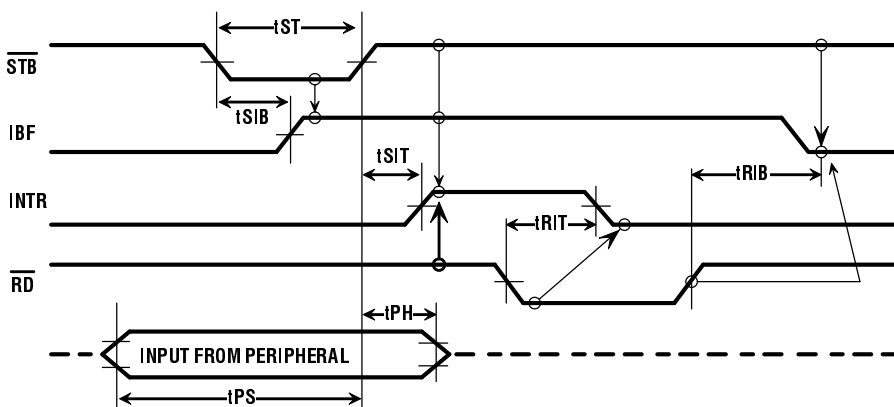
Palabra de Control

| D <sub>7</sub> | D <sub>6</sub> | D <sub>5</sub> | D <sub>4</sub> | D <sub>3</sub> | D <sub>2</sub> | D <sub>1</sub> | D <sub>0</sub> |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1              | 0              | 1              | 1              | 1/0            | 1              | 1              | ⊗              |

{ 1: Entrada  
 0: Salida

## PPI: Protocolo de entrada en Modo 1

- Modo 1: Protocolo de comunicación con *handshaking* para la entrada.
  - STB: Un nivel bajo en esta entrada da la señal de memorizar el dato que se encuentra en la entrada.
  - IBF: Un nivel alto, indica que el dato ha sido memorizado en los latches de entrada.
  - INTR: Se activa al pasar STB a nivel alto, siempre que IBF esté a nivel alto e INTE esté habilitado.



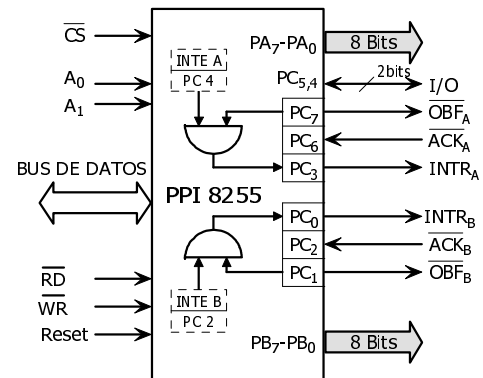


# PPI: Funcionamiento en Modo 1 (Salida)

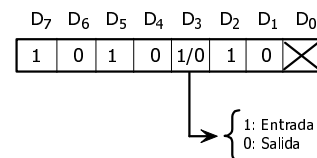
- Modo 1: Transferencia con *Handshaking*. Las Puertas A y B son controladas por las líneas de la puerta C (3 líneas por puerta).

## Modo de Salida:

- Puerta B: Controlada por los bits PC<sub>0</sub> a PC<sub>3</sub>.
- Puerta A: Controlada por los bits PC<sub>3</sub>, PC<sub>6</sub> y PC<sub>7</sub>.
- PC<sub>4</sub> y PC<sub>5</sub>, pueden configurarse como líneas convencionales de E/S.
- Señales del protocolo de salida: OBF, ACK e INTR.
- Habilitación de las interrupciones a través de los biestables internos INTEA (PC<sub>6</sub>) e INTEB (PC<sub>2</sub>).

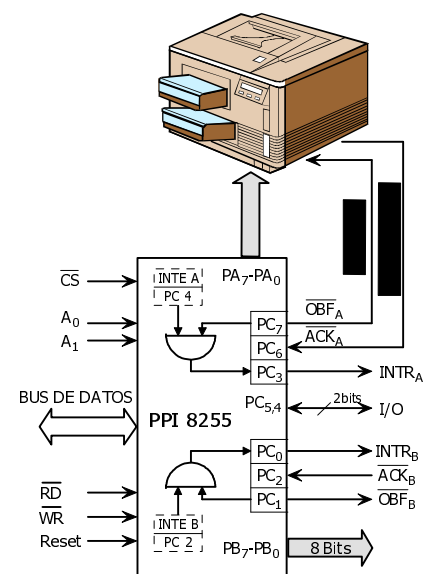
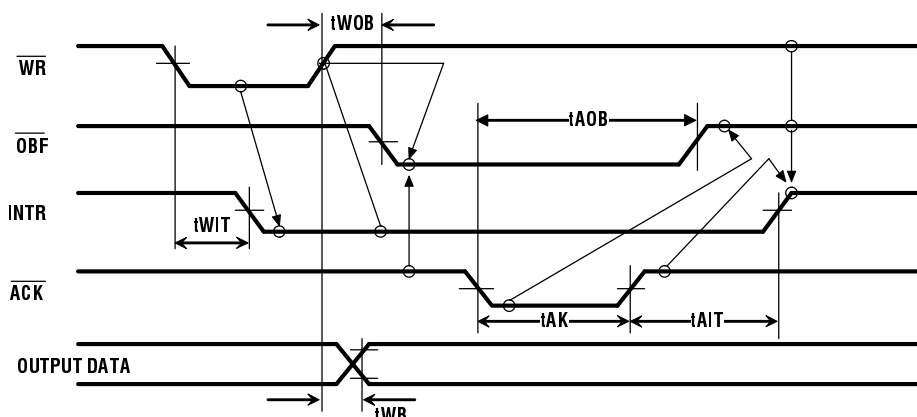


Palabra de Control



## PPI: Protocolo de salida en Modo 1

- Modo 1: Protocolo de comunicación con *handshaking* para la salida.
  - $\overline{\text{OBF}}$ : Esta señal se activa a nivel bajo cuando queda lleno el buffer de salida de la puerta.
  - $\overline{\text{ACK}}$ : Un nivel bajo, señala el reconocimiento de la salida cuando el periférico lee la puerta.
  - $\text{INTR}$ : Se activa a nivel alto con la señal de reconocimiento, siempre que  $\overline{\text{OBF}}$  esté a nivel alto e INTE esté habilitado.

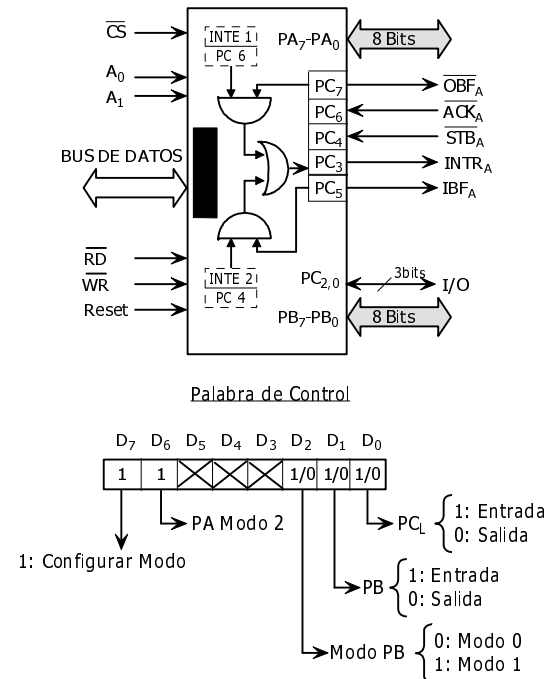


# PPI: Funcionamiento en Modo 2

## Modo 2: Transferencia bidireccional con *Handshaking*.

### Características:

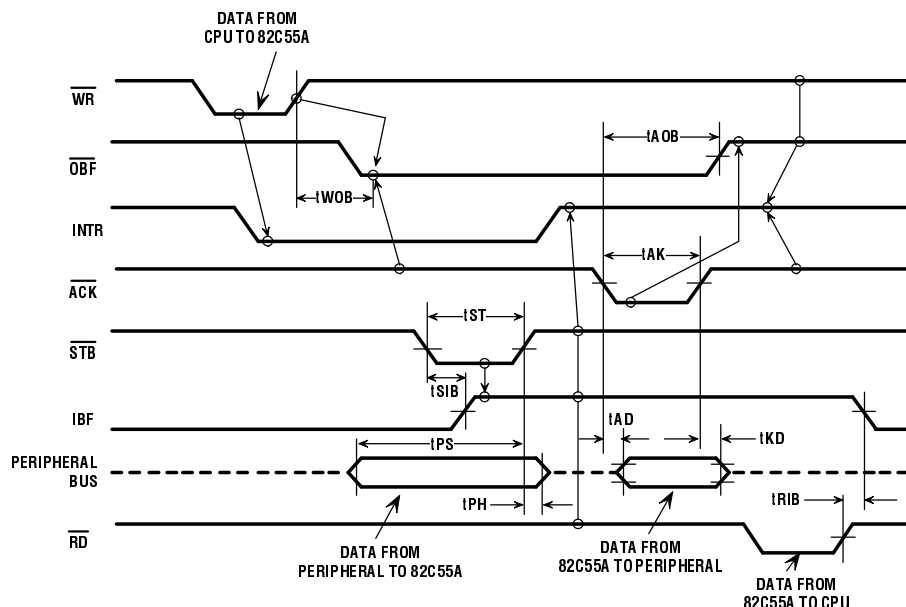
- Solamente puede programarse para la Puerta A.
- Puerta A: Controlada por los bits PC<sub>3</sub> a PC<sub>7</sub>.
- Puerta B: Puede funcionar en Modo 0 ó Modo 1 (Controlada por PC<sub>0</sub> a PC<sub>2</sub>).
- Señales del protocolo: IBF, STB, OBF, ACK e INTR.
- Habilitación de la interrupción a través de los biestables internos INTE 1 (PC<sub>6</sub>) e INTE 2 (PC<sub>4</sub>).



# PPI: Protocolo en Modo 2

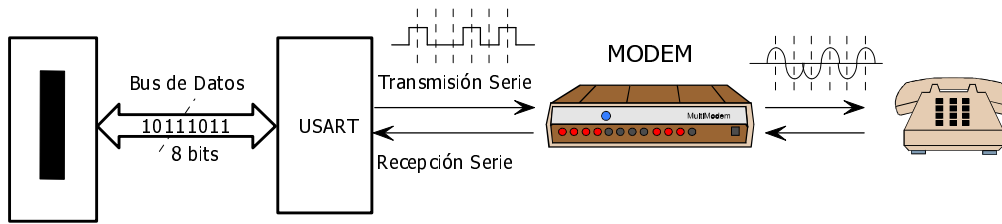
## Modo 2: Protocolo de comunicación bidireccional con *handshaking*.

- INTR: La petición de interrupción se activa tanto por el protocolo de entrada como por el protocolo de salida. Estas peticiones pueden deshabilitarse por separado (biestable INTE 1 e INTE 2).



# Introducción a la E/S serie

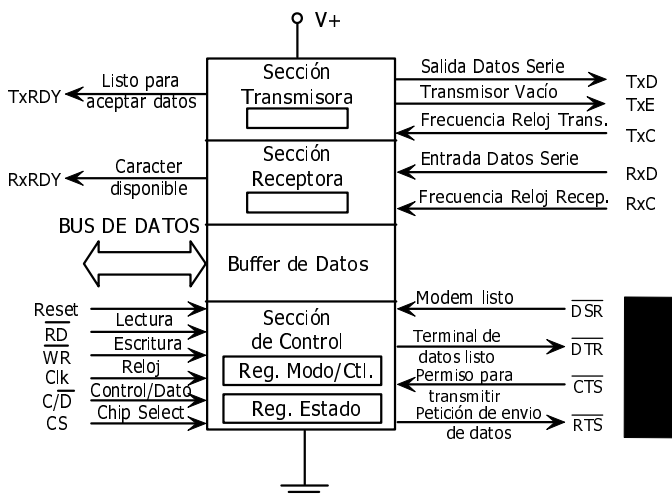
## ■ Generalidades sobre la transmisión serie:



- Necesidad de un dispositivo adaptador entre los periféricos serie y el  $\mu$ procesador: el USART.
  - ├ Necesidad de señales de reloj para la transmisión y recepción de datos.
  - ├ Comunicaciones a largas distancias. Utilización del MODEM. Señales analógicas.
    - Modulación en Amplitud, frecuencia, o fase.
  - ├ Modos de transmisión síncrono y asíncrono.

## Diagrama de bloques de una USART

## ■ Esquema general:



### ■ Sección Transmisora:

- Envía los datos en serie por TxD.
- TxC recibe una señal cuya frecuencia sincroniza el reloj de transmisión.
- Comunica al  $\mu$ procesador la posibilidad de enviar otro carácter por TxRDY.

### ■ Sección Receptora:

- Recibe datos serie por RxD.
- Sincroniza la frecuencia del reloj receptor a través de RxC.
- Comunica al  $\mu$ procesador la presencia de un carácter usando TxRDY.

### ■ Sección de Control:

- Gestiona la comunicación con el  $\mu$ procesador y el MODEM.

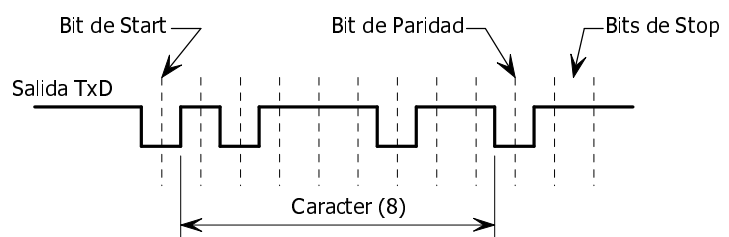
# Funcionamiento de una USART

- 1.- Configuración del modo de funcionamiento:
  - Frecuencia, en baudios, para la transmisión serie.
  - Anchura del carácter a transmitir.
  - Número de bits de STOP.
  - Selección del tipo de operación: síncrona o asíncrona.
  - Existencia del bit de paridad y tipo.
- 2. La señal TxRDY pasa a nivel alto indicando disponibilidad para la transmisión. La señal TxD está a nivel alto en ausencia de información.
  - La transmisión es posible siempre que TxEN esté a nivel alto y se active la señal CTS (permiso para transmitir).
- 3. La señal RxRDY pasa a nivel alto indicando que se ha recibido un dato.

## Tipos de Transmisiones

### ■ Transmisión Asíncrona:

- | Señal de transmisión a nivel alto en ausencia de información.
- | Utilización de bits de Start y Stop (comienzo y fin de la transmisión).
- | El carácter es codificado en 5, 6 ó más bits.
- | Bit de paridad: Control de la transmisión.
  - Ejemplo de transmisión: Dato 10111011



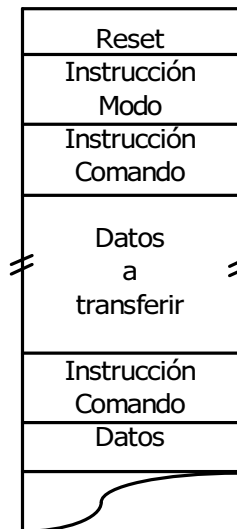
### ■ Transmisión Síncrona:

- | No incluye los bits de Start y Stop.
- | Transmisión por bloques de N caracteres.
- | Uso de caracteres de sincronización.

# Programación de la USART 8251 de Intel

## I Configuración de la USART por programa:

- I Se realiza escribiendo dos palabras de control sobre el registro de control, denominadas *instrucción de modo* e *instrucción de comando*.
- I La instrucción de Modo va siempre después de un Reset del dispositivo.
  - Reset Externo
  - Reset Interno (a través de la instrucción de Comando).
- I Después de una instrucción de Modo, le sigue una instrucción de Comando.
- I La instrucción de Comando puede insertarse en cualquier momento.

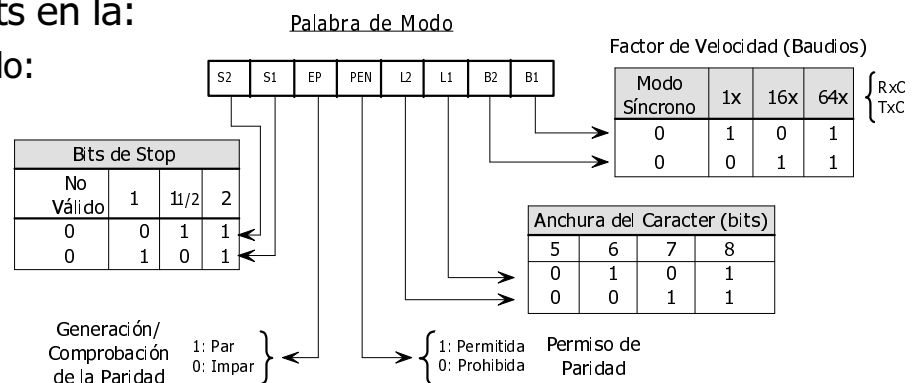


*Disposición de la secuencia de instrucciones que inicializa el USART.*

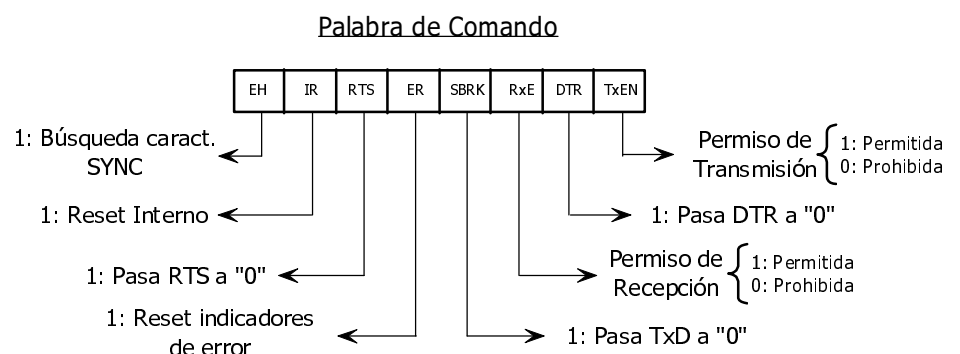
## Instrucciones de Modo y Comando

### I Disposición de los bits en la:

#### I Instrucción de Modo:



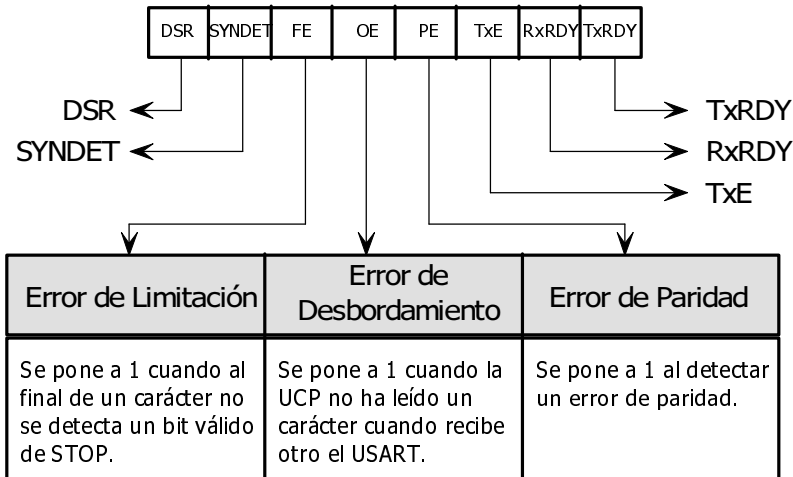
#### I Instrucción de Comando:



# Palabra de estado del 8251 de Intel

- Se obtiene mediante la lectura en el Registro de Control:
  - ┆ Señala las condiciones de errores en la transferencia de información.
  - ┆ Permite la comprobación de la existencia de un dato a leer y la posibilidad de transmitir, señales TxRDY y RxRDY, a través de una exploración por programa.

Palabra de Estado



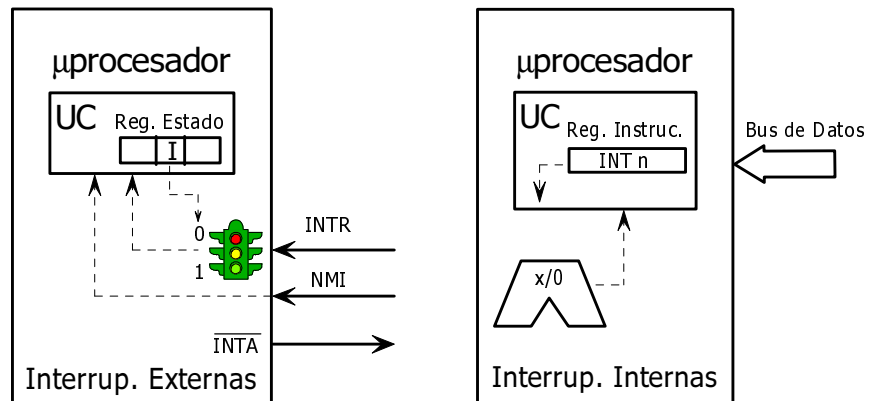
## Introducción a las Interrupciones

- Manejo de las interrupciones en el sistema: Generalidades.
  - ┆ Una interrupción es una llamada a una subrutina.
  - ┆ La petición de interrupción es asíncrona.
  - ┆ Ante un interrupción el sistema:
    - ┆ Termina de ejecutar la instrucción en curso.
    - ┆ Salva en la pila la dirección de retorno y el registro de estado.
    - ┆ Carga en el PC la dirección de comienzo de una rutina, denominada *Rutina de Servicio de la Interrupción* y la ejecuta (cada interrupción está asociada a una RSI).
    - ┆ Ejecuta la RSI, y termina con un IRET (Retorno de Interrupción), que además de recuperar de la pila la dirección de retorno, recupera el registro de estado.
  - ┆ Cada interrupción está asociada a un número, denominado *Vector de Interrupción*.

# Clasificación de las Interrupciones

## ■ Clasificación de las interrupciones:

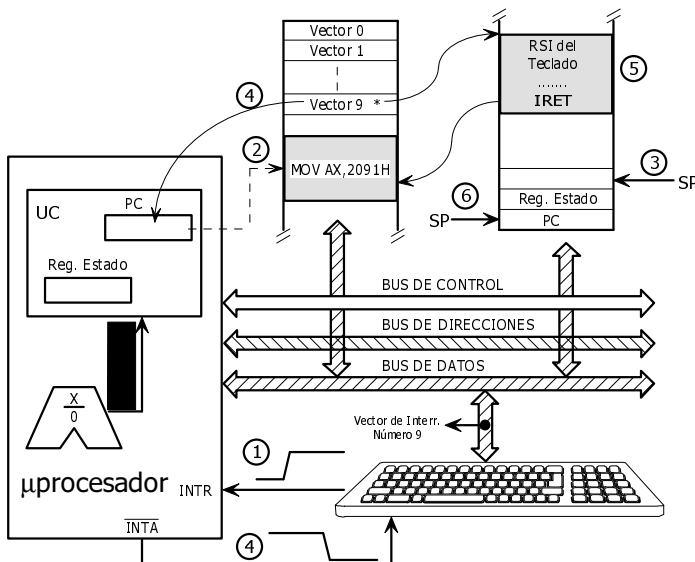
- **Internas:** Se producen dentro del  $\mu$ procesador y pueden ser:
  - | *Automáticas:* Desbordamiento por división, Overflow, Ejecución paso a paso.
  - | *Provocadas por software:* Se producen al ejecutarse una instrucción del tipo "INT n".
- **Externas:** Se piden desde fuera de la CPU y se clasifican en:
  - | *Enmascarables.*
  - | *No Enmascarables.*



## Vectorización de las interrupciones

## ■ Las interrupciones pueden a su vez clasificarse en:

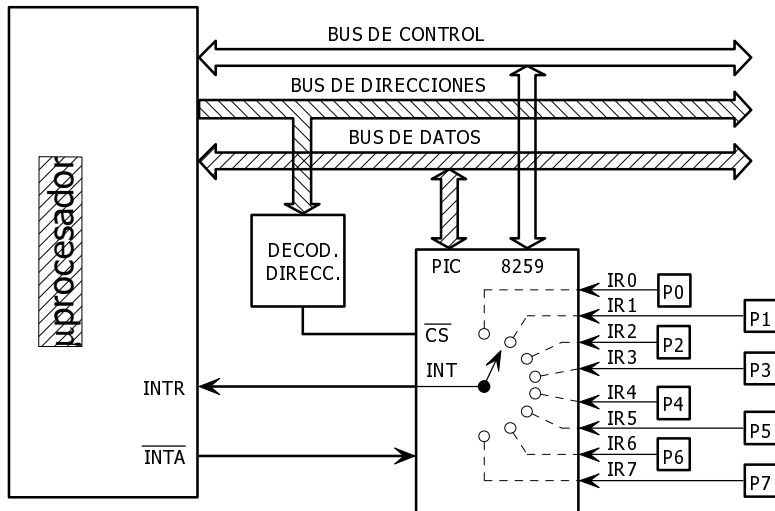
- | *Vectorizadas:* Están asignadas por el sistema a un vector de interrupción. Todas las interrupciones internas están vectorizadas.
- | *No Vectorizadas:* Mandan el vector de interrupción al pedir la interrupción.



- 1.- Petición de Interrupción.
- 2.- Terminar la instrucción en curso.
- 3.- Guardar en la pila la dirección de retorno y el registro de estado.
- 4.- Reconocimiento de la interrupción. El contenido del vector de interrupción se carga en el contador de programa.
- 5.- Se ejecuta la Rutina de Servicio de la Interrupción.
- 6.- Recuperar de la pila la dirección de retorno (instrucción IRET).

# Introducción al PIC (8259) de Intel

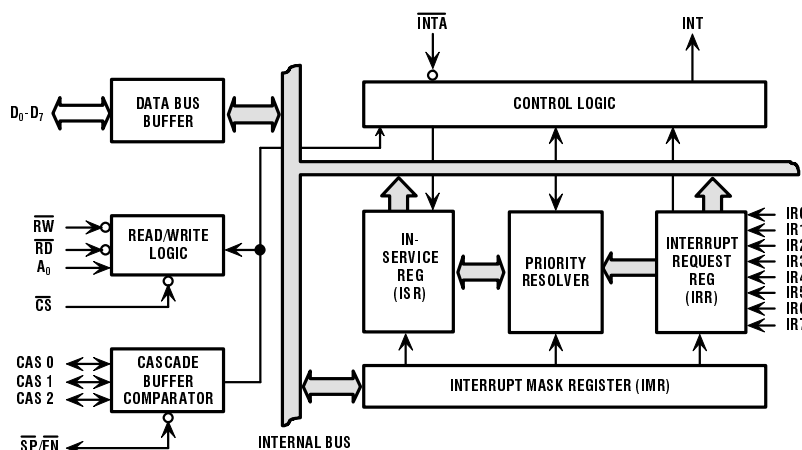
## ■ Controlador de Interrupciones, PIC 8259: Funcionamiento.



- Recibe las peticiones de interrupción de los dispositivos periféricos.
- Determina las prioridades de las diferentes peticiones.
- Posibilita el enmascaramiento individual de las interrupciones.
- Realiza la petición de interrupción al  $\mu$ procesador.
- Deposita el vector de interrupción en el bus de datos del sistema.

## Diagrama de bloques del 8259

### ■ Estructura interna del PIC:



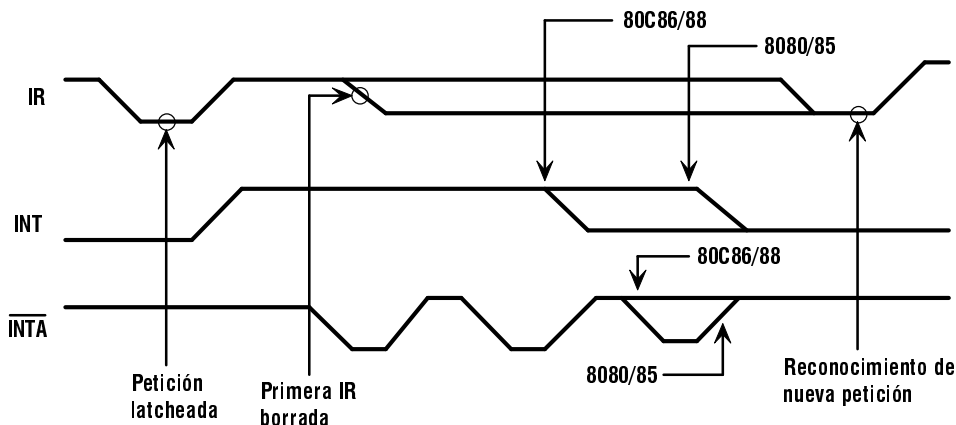
- IRR: Registra los periféricos que solicitan servicio de interrupción.
- IMR: Registro de 8 bits que contiene las máscaras de interrupciones. Un bit a "0" habilita la interrupción asociada a ese bit.
- ISR: Registra la petición de interrupción que está siendo atendida en ese momento.
- Priority Resolver: es una lógica intermedia que resuelve la petición más prioritaria que no esté enmascarada.
- Modos de Funcionamiento: Aislado o en cascada con estructura de Maestro -- Esclavo. Señal SP/EN.
- Cascade Buffer Comparator: Registra los códigos de los PIC esclavos cuando existe una estructura en cascada.



# Protocolo de comunicación del PIC

## I Modo de funcionamiento aislado:

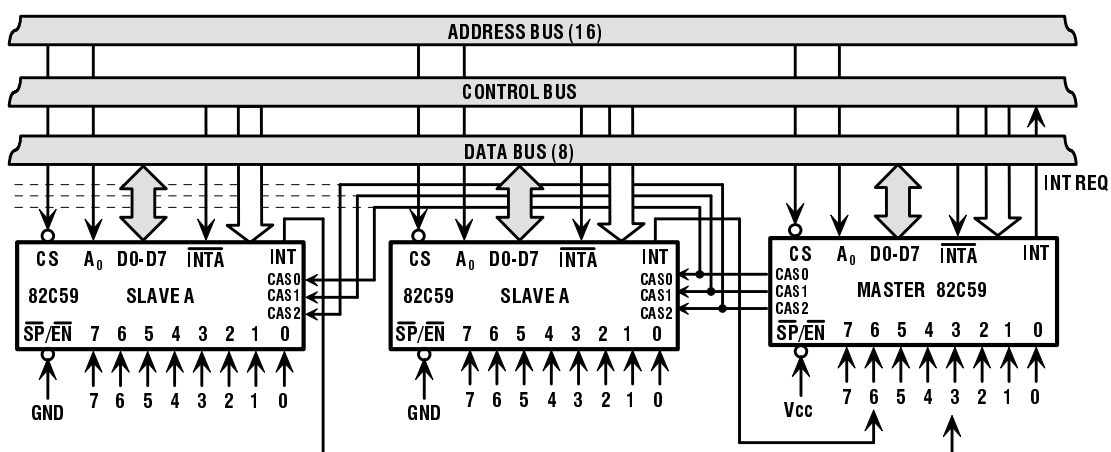
- | IR: Debe permanecer a nivel alto hasta el primer flanco de bajada de la señal de reconocimiento  $\overline{INTA}$ .
- | INT: Se activa al recibirse un IR no enmascarado. Desaparece en el segundo flanco de subida de la señal de reconocimiento.
- |  $\overline{INTA}$ : Con el primer ciclo el 8259 activa el bit de mayor prioridad que ha requerido interrupción. También resetea el bit correspondiente en el registro IRR. Con el segundo ciclo el 8259 deposita el vector de interrupción en el bus de datos.



## Conexión del PIC en cascada

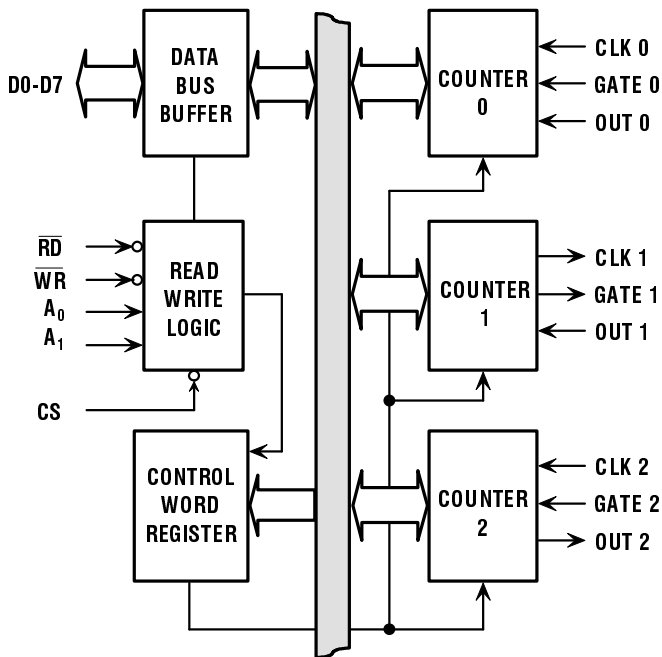
## I Modo de funcionamiento en cascada:

- | El 8259 puede funcionar como maestro (*master*) controlando un máximo de 8 esclavos (*slaves*) 8259, manejando hasta 64 posibles interrupciones.
- | Las peticiones de interrupción de los *slaves* son dirigidas a las entradas de interrupción del *master*.
- | El *master* controla los *slaves* a través de las líneas CAS0-2 que actúan como *chip-select* de éstos durante los ciclos de  $\overline{INTA}$ .



# Controlador programable de tiempos. Timer

## ■ Diagrama de Bloques del 82C53/82C54 de Intel:



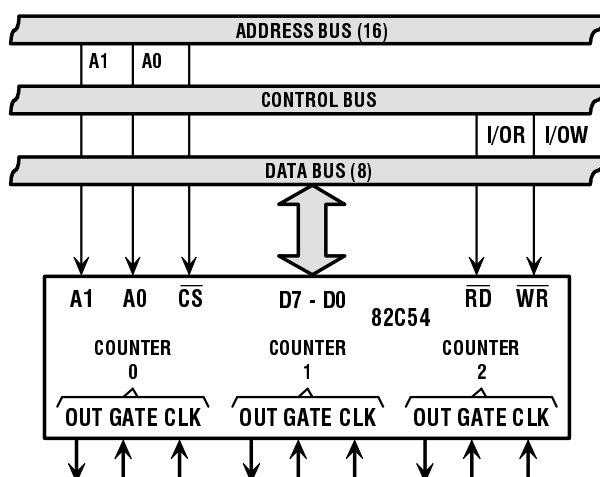
- Permite generar tiempos de retardo precisos, mediante control por programa.
- Comunicación con la CPU a través de interrupciones, mediante las salidas OUT0-3.
- Características:
  - Tres contadores de 16 bits, totalmente independientes, seleccionados con las señales  $A_0$  y  $A_1$ .
  - Entradas de Reloj independientes para cada contador, CLK0-3.
  - Contaje en binario o en BCD.
  - Velocidad de contaje: 4MHz, (12MHz para el 82C54).
  - Dispone de 6 modos de funcionamiento.

## Programación del Timer 82C53/4

### ■ Programación del 8253:

- Los contadores se programan escribiendo primero la Palabra de Control y, posteriormente, un contaje inicial.
- La palabra de control se escribe sobre el Registro de Control ( $A_0=1$  y  $A_1=1$ ).
- El contaje inicial se escribe sobre los contadores. Selección con  $A_0$  y  $A_1$ . Este contaje depende del modo de escritura seleccionado en la Palabra de Control.

### ■ Conexión del dispositivo



### Resumen de operaciones $\overline{RD}/\overline{WR}$ :

| CS | RD | WR | A1 | A0 | Operation                  |
|----|----|----|----|----|----------------------------|
| 0  | 1  | 0  | 0  | 0  | Write into Counter 0       |
| 0  | 1  | 0  | 0  | 1  | Write into Counter 1       |
| 0  | 1  | 0  | 1  | 0  | Write into Counter 2       |
| 0  | 1  | 0  | 1  | 1  | Write Control Word         |
| 0  | 0  | 1  | 0  | 0  | Read from Counter 0        |
| 0  | 0  | 1  | 0  | 1  | Read from Counter 1        |
| 0  | 0  | 1  | 1  | 0  | Read from Counter 2        |
| 0  | 0  | 1  | 1  | 1  | No-Operation (Three State) |
| 1  | X  | X  | X  | X  | No-Operation (Three State) |
| 0  | 1  | X  | X  | X  | No-Operation (Three State) |

# Palabra de control del 82C53

## I Formato de la Palabra de Control:

A1,A0=11; CS=0; RD=1; WR=0

| D7  | D6  | D5  | D4  | D3 | D2 | D1 | D0  |
|-----|-----|-----|-----|----|----|----|-----|
| SC1 | SC0 | RW1 | RW0 | M2 | M1 | M0 | BCD |

| SC  | Select Counter |                   |
|-----|----------------|-------------------|
| SC0 | SC1            |                   |
| 0   | 0              | Select Counter 0  |
| 0   | 1              | Select Counter 1  |
| 1   | 0              | Select Counter 2  |
| 1   | 1              | Read-Back Command |

| RW  | Read/Write |   |
|-----|------------|---|
| SC0 | SC1        |   |
| 0   | 0          | Counter Latch Command   |
| 0   | 1          | RD/WR least significant byte only                                 |
| 1   | 0          | RD/WR most significant byte only                                  |
| 1   | 1          | RD/WR least significant byte first,<br>then most significant byte |

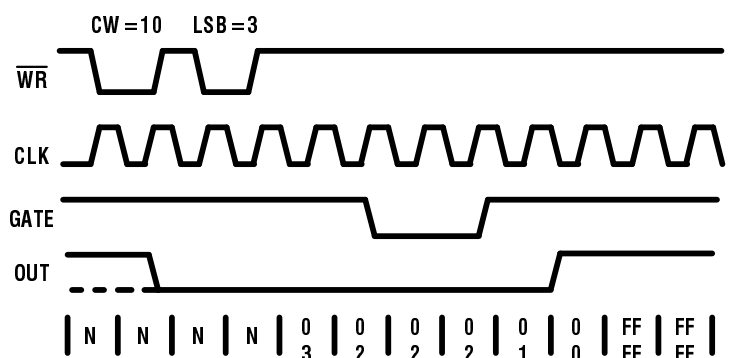
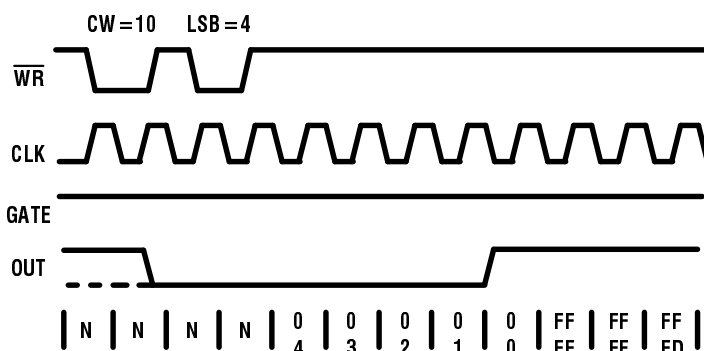
| BCD | Binary Code Decimal                            |
|-----|--|
| 0   | Binary Counter 16-Bit                          |
| 0   | Binary Coded Decimal (BCD) Counter (4 Decades) |

| M  | Mode |    |        |
|----|------|----|--------|
| M2 | M1   | M0 |        |
| 0  | 0    | 0  | Mode 0 |
| 0  | 0    | 1  | Mode 1 |
| X  | 1    | 0  | Mode 2 |
| X  | 1    | 1  | Mode 3 |
| 1  | 0    | 0  | Mode 4 |
| 1  | 0    | 1  | Mode 5 |

## 82C53: Funcionamiento en Modo 0

### I Modo 0: *Interrupción al final del conteaje.*

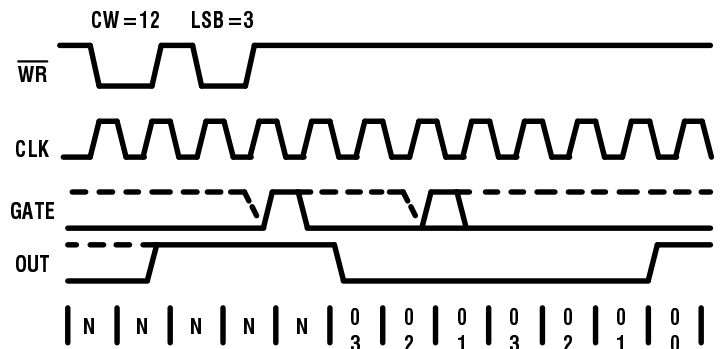
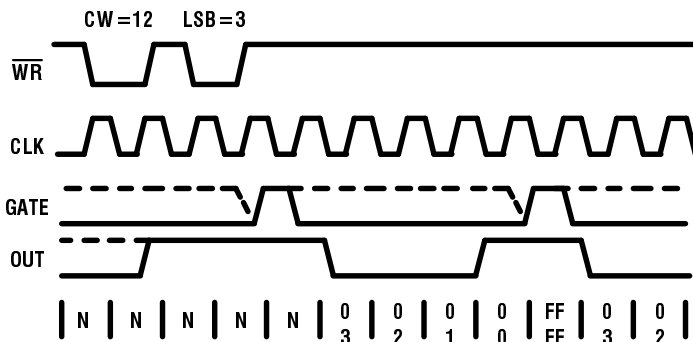
- Se utiliza para contar eventos.
- La salida pasa a nivel bajo cuando se programa. Al final del conteaje, la salida pasa a nivel alto.
- El conteaje comienza después de iniciar el contador.
- La señal GATE detiene el conteaje.



## 82C53: Funcionamiento en Modo 1

### I Modo 1: *Monoestable Programmable.*

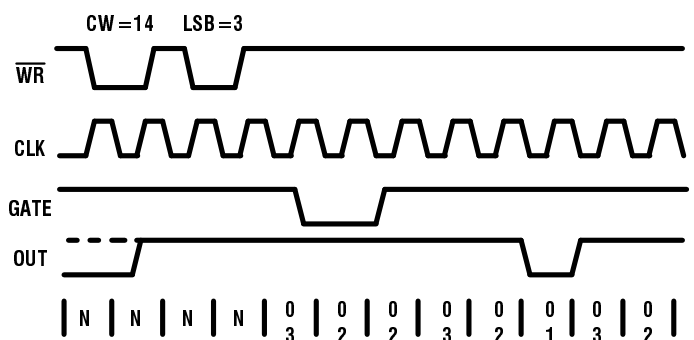
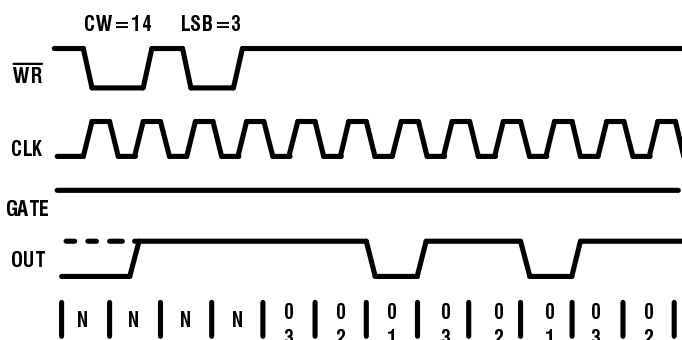
- I La salida pasa a nivel alto cuando se programa, y pasa a nivel bajo cuando comienza el conteo.
- I El conteo comienza después del TRIGGER de la señal GATE, y la salida pasa a nivel alto al final del conteo.
- I Un TRIGGER: al final del conteo lo inicializa, poniendo la salida a nivel bajo; durante el conteo lo inicializa, manteniendo la salida a nivel bajo.



## 82C53: Funcionamiento en Modo 2

### I Modo 2: *Divisor de Frecuencia.*

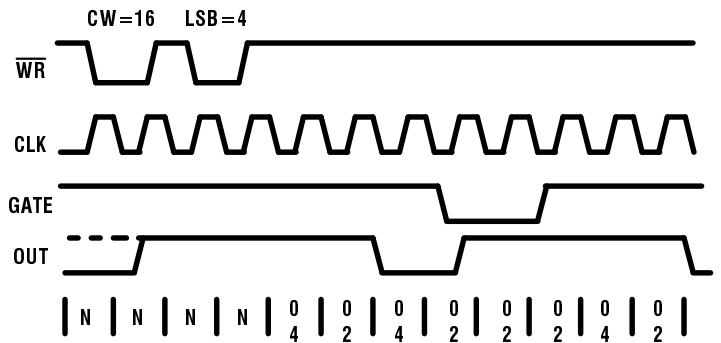
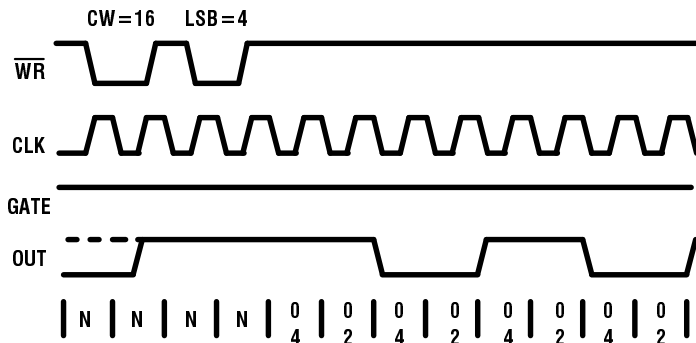
- I Funciona como un contador de módulo programable.
- I La salida pasa a nivel alto cuando se programa. Una vez programado, pasa a nivel bajo durante el último periodo de CLK.
- I Este modo es periódico; la secuencia se repite indefinidamente.
- I Un TRIGGER en la señal GATE inicializa el conteo.



## 82C53: Funcionamiento en Modo 3

### I Modo 3: *Generador de Onda Cuadrada.*

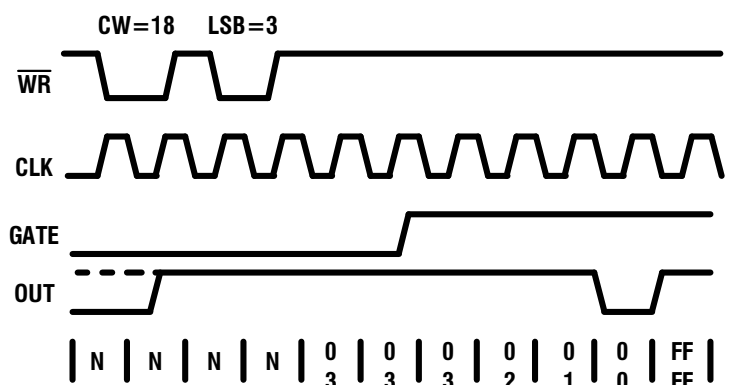
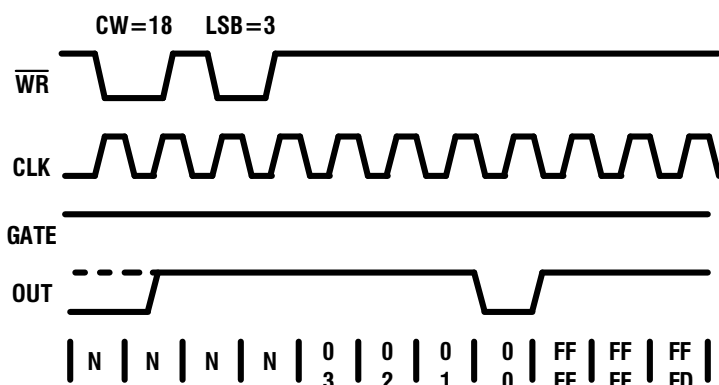
- I Similar al modo 2, excepto el ciclo de la salida OUT.
- I La salida pasa a nivel alto cuando se programa. La salida pasa a nivel bajo durante la segunda mitad del conteaje.
- I Para módulos impares la salida permanece a nivel alto un periodo más de la señal CLK.



## 82C53: Funcionamiento en Modo 4

### I Modo 4: *Retardo Activado por Software.*

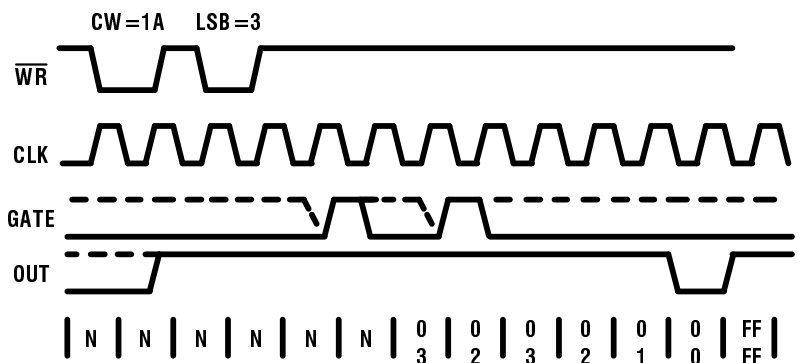
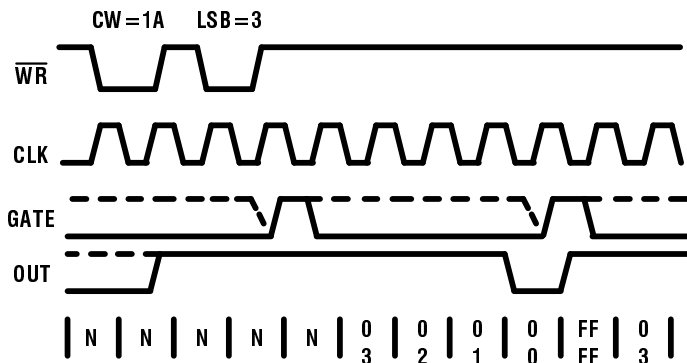
- I La salida pasa a nivel alto cuando se programa. Una vez programado, pasa a nivel bajo al final del conteaje, durante un periodo de CLK.
- I El conteaje se detiene cuando la entrada GATE pasa a nivel bajo.
- I El conteaje se dispara al escribir siempre un nuevo dato.



## 82C53: Funcionamiento en Modo 5

### I Modo 5: *Retardo Activado por Hardware.*

- I La salida pasa a nivel alto cuando se programa. Una vez programado, pasa a nivel bajo al final del conteaje, durante un periodo de CLK.
- I El conteaje se dispara por un flanco de subida de la señal GATE.
- I Un TRIGGER en la señal GATE inicializa el conteaje.



## 8253: Modo4 vs Modo 5

### I Modo 4 y 5:

- I Modo 4: El conteaje se inicializa siempre al escribir un nuevo dato. El conteaje se inicializa por *software*.
- I Modo 5: El conteaje se inicializa siempre al recibir un TRIGGER por la señal GATE. La escritura de un nuevo valor no afecta el conteaje hasta después del TRIGGER. El conteaje se inicializa por *hardware*.

