

## Practica 3 - Parte 2 - HAND-SHAKE y CDMA

24 de septiembre de 2013

# Hand-Shake

- Interfaz que nos permite comunicarnos facilmente con la Impresora, ya que realiza la temporización automáticamente
- Posee dos registros, de 8 bits.
  - DATO: Registro de datos. De lectura y escritura. Es el caracter a enviar o el ultimo enviado.
  - EST: Un registro de estado.
- Los dos registros estan a partir de la posición 40h.
  - 40h = DATO
  - 41h = EST

## Continuación

- El registro de estado
  - Bit 0: Línea Busy - Idem Impresora
  - Bit 1: Línea Strobe - Idem Impresora
  - Bit 2..6: No tienen sentido
  - Bit 7: Interrupción: 0 = Desactivada, 1 = Activada
- ¿Cuándo se dispara la interrupción? Cuando la línea BUSY se desactiva.
- Tenemos 2 maneras de utilizar el HAND-SHAKE: con interrupciones o sin interrupciones

# Interrupción

- Para utilizar HAND-SHAKE con interrupciones debemos usar el modo de configuración 2 (c2)
- La interrupción que genera el HAND-SHAKE se conectará a la interrupción de nivel 2 (INT2) del PIC
- Cuando el manejador de la interrupción sea invocado. La impresora va a estar lista para recibir un caracter.

## Como usar el HAND-SHAKE sin interrpciones

- El simulador debe estar en la Configuración 1
- Debemos esperar a que la impresora este lista, consultado el estado de la linea BUSY en el registro EST
- Cuando la impresora este lista, escribimos el caracter a imprimir en el registro DATO.
- ¿Qué diferencias hay con el uso de la impresora con el PIO?

```
POLL: IN AL, 41h
      AND AL, 1
      JNZ POLL

      MOV AL, PROX_CAR
      OUT 40h, AL
```

## Como usar el HAND-SHAKE con interrpciones

- El simulador debe estar en la configuración 2
- Debemos programar el PIC para que atienda la interrupción de nivel 2 (INT2)
- Debemos cargar la dirección de la rutina en el vector de interrupciones correspondiente
- En la rutina de la interrupción escribimos el caracter a enviar a la impresora

RUT\_HAND:

```
...  
MOV AL, PROX_CAR  
OUT 40h, AL  
...  
IRET
```

# Controlador de DMA

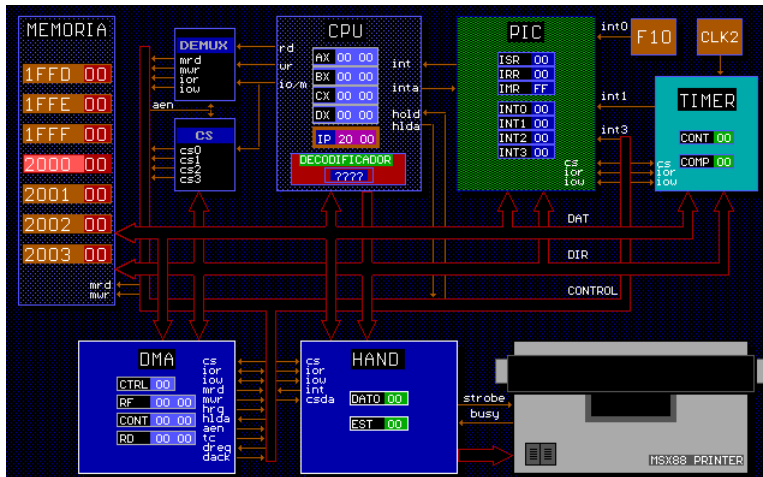
- ¿Que es DMA?
- Nos permite realizar transferencias de datos de 8 bits memoria-memoria, memoria-periférico o periférico-memoria.
- Por ser una transferencia DMA la CPU no interviene, pero si debe cederle el bus al CDMA.
- En el simulador hay que utilizar la Configuración 3 (c3).

# Características

- El simulador posee un solo canal DMA
- Puede realizar transferencias en dos modos diferentes:
  - Modo Bloque: Se enviará el bloque completo una vez iniciada la transferencia
  - Modo Bajo demanda: del periférico al que se encuentre conectado
- Cuando el CDMA termina de realizar la transferencia genera una interrupción.
- El CDMA está conectado a la línea de interrupción 3 del PIC.



# Conexión



# Registros

El controlador de DMA posee los siguientes registros

- *CTRL*: Un registro de control que nos permite configurar el funcionamiento del CDMA
- *RF*: Registro de direcciones fuente.
  - En transferencias memoria-periferico o periferico-memoria, indica la memoria de donde leer o a donde escribir.
  - En transferencias memoria-memoria, es la posición de memoria de los datos a copiar.
- *RD*: Registro de direcciones destino.
  - Solo tiene sentido si es memoria-memoria.
- *CONT*: Registro Contador. Indica el número de bytes a transferir
- *ARRANQUE*: Registro de Arranque. Accediendo a este registro, se inicia la transferencia.

## Registro de Control



- El formato del registro *CTRL* depende si lo estamos leyendo o escribiendo.
- En lectura
  - *STOP*:
    - 0: Transferencia en Curso
    - 1: Transferencia detenida por la CPU
  - *TC*:
    - 0: Transferencia no finalizada
    - 1: Transferencia ya finalizada

# Registro de Control

- En escritura
  - *STOP*:
    - 0: No tiene sentido
    - 1: Detener momentaneamente la transferencia en curso
  - *TT*: Tipo de Transferencia
    - 0: Transferencia Periferico-Memoria o Memoria-Periferico
    - 1: Transferencia Memoria-Memoria
  - *ST*: Sentido de la Transferencia (solo TT=0)
    - 0: Sentido Periférico-Memoria
    - 1: Sentido Memoria-Periferico
  - *MT*: Modo de Transferencia
    - 0: Por demanda
    - 1: Por Bloques

## Direccionamiento

- Los registros se ubican a partir de la dirección 50h
  - *RF*: es de 16 bits
    - *RFL*: 050h - parte baja de *RF*
    - *RFH*: 051h - parte alta de *RF*
  - *CONT*: es de 16 bits
    - *CONTL*: 052h - parte baja de *CONT*
    - *CONTH*: 053h - parte alta de *CONT*
  - *RD*: es de 16 bits
    - *RDL*: 054h - parte baja de *RD*
    - *RDH*: 055h - parte alta de *RD*
  - *CTRL*: 056h
  - *ARRANQUE*: 057h

## ¿Cómo lo usamos?

- Demasiadas opciones, demasiadas configuraciones...
- Veamos para que lo vamos a utilizar en el simulador...
  - Para copiar una parte de la memoria a otra posición
  - Para mandar caracteres a la impresora a través del HAND-SHAKE

# Memoria-Memoria

- Configuramos el registro de Control
  - Tipo Transferencia: Memoria-Memoria
  - Modo Transferencia: Bloques
- Configuramos el registro *RF* con la posición origen de la memoria
- Configuramos el registro *RD* con la posición destino de la memoria
- Configuramos el registro *CANT* con la cantidad de bytes a transferir
- Configuramos el PIC y un manejador de interrupción para saber cuando la transferencia terminó.
- Acceder al registro *ARRANQUE* para iniciar la transferencia

## Ejemplo Memoria-Memoria

Veamos el ejercicio 10 de la práctica 3



## Memoria-Periferico

- Configuramos el registro de Control
  - Tipo Transferencia: Memoria-Periferico
  - Sentido Transferencia: Memoria-Periferico
  - Modo Transferencia: bajo demanda
- Configuramos el registro *RF* con la posición origen de la memoria
- Configuramos el registro *CANT* con la cantidad de bytes a transferir
- Configuramos el PIC y un manejador de interrupción para saber cuando la transferencia terminó.
- Habilitamos el uso de interrupciones del HAND-SHAKE (conectado al CDMA)
- Acceder al registro *ARRANQUE* para iniciar la transferencia

## Ejemplo Memoria-Periferico

Veamos el ejercicio 11 de la práctica 3