

1) Las transferencias DMA podemos dividirlas en 3 etapas :

- Inicialización de la transferencia : interviene la CPU
- Transferencia propiamente dicha : no interviene la CPU, la transferencia es controlada por CDMA (Controlador de Acceso Directo a Memoria)
- Finalización de la transferencia : interviene la CPU.

Inicialización :

- Escribe en el registro correspondiente del PIC, INT3 (dirección 27H = PIC+7) la posición seleccionada en la tabla de vectores de interrupción, en este caso N_DMA=20.
- Carga en los registros de CDMA la dirección donde comienza el bloque de memoria a transferir. Registro RFL= reg fuente parte baja, dirección DMA=50H y RFH= reg fuente parte alta, dirección DMA+1=51H. Las direcciones son de 16 bits y estos registros de 8 bits, por eso hay dos.
- Carga en los registros del CDMA la cantidad de bytes a transferir, dirección de los reg CONTL, DMA+2=52H parte baja y CONTH DMA+3=53H parte alta.
- Carga en los registros del CDMA la dirección destino, es decir donde se copia el bloque de memoria, DMA+4=54H – RDL=reg dirección de destino parte baja y DMA+5=55H – RDH= reg dirección de destino parte alta.
- Configura el DMA para transferencia memoria a memoria por bloque. Reg CTRL. dirección del reg DMA+6=56H.
- Habilitar en el reg IMR (máscara) del PIC, dirección PIC+1=21H, la int3 que se produce al terminar la transferencia.
- Escribiendo en el reg de arranque del CDMA, dirección DMA+7=57H, el número 7 arranca la transferencia.

Como el programa lo pide se muestra en pantalla (en el programa principal), el contenido del bloque de memoria a transferir (mensaje original). Esto no es parte de la inicialización, pero va junto con ella en ORG 2000H (programa principal).

Transferencia:

La transferencia es entre el CDMA y la memoria, no interviene la CPU, por eso no hay ninguna instrucción que “copie” los datos. Durante la transferencia el CDMA actúa como si fuese “CPU” controlando la transferencia, está destinado a esta tarea específica.

Recordar que la CPU no es interrumpida para realizar la transferencia, sólo detenida, por lo que no hay que salvar direcciones de retorno, contexto, etc. Durante la transferencia no se usan los reg de la CPU sino los del CDMA.

Finalización:

El CDMA genera una interrupción, llega al PIC por int3, y aquí sí interviene la CPU, lo que hace es deshabilitar en el IMR del PIC la int3.

Luego se muestra la cadena transferida sólo porque es pedido en el enunciado, no porque forme parte de la finalización.

Arquitectura de Computadoras – Fac. de Informática - UNLP - Prof. Jorge M. Runco
Curso 2014 – TP N° 3 : Acceso Directo a Memoria (DMA)

```
ORG 2000H  
CLI
```

```
MOV AL,N_DMA  
OUT PIC+7,AL ; carga en el registro INT3 del PIC  
; el tipo de int.
```

```
MOV AX,OFFSET MSJ  
OUT DMA,AL  
MOV AL,AH  
OUT DMA+1,AL ; carga en el CDMA la dirección de  
; comienzo del bloque de datos a  
; transferir
```

```
MOV AX,OFFSET FIN-OFFSET MSJ  
OUT DMA+2,AL  
MOV AL,AH  
OUT DMA+3,AL ; carga en el CDMA la cantidad de  
; bytes a transferir
```

```
MOV AX,OFFSET COPIA  
OUT DMA+4,AL  
MOV AL,AH  
OUT DMA+5,AL ; carga en el CDMA la dirección de  
; destino del bloque de datos
```

```
MOV AL,0AH  
OUT DMA+6,AL ; configura el CDMA para transf. mem-  
; mem por bloque
```

```
MOV AL,0F7H  
OUT PIC+1,AL ; habilita INT3
```

```
STI
```

```
MOV BX,OFFSET MSJ  
MOV AL,OFFSET FIN-OFFSET MSJ  
MOV NCHAR,AL  
INT 7 ;muestra en pantalla el mensaje  
; original
```

```
MOV AL,7H  
OUT DMA+7,AL ;arranque de la transferencia
```

```
INT 0  
END
```

Arquitectura de Computadoras – Fac. de Informática - UNLP - Prof. Jorge M. Runco
Curso 2014 – TP N° 3 : Acceso Directo a Memoria (DMA)

```
PIC      EQU      20H
DMA      EQU      50H
N_DMA    EQU      20      ; Lugar en la tabla de vectores

          ORG 80      ; 20x4=80 Dirección para el "lugar" 20
IP_DMA   DW      RUT_DMA

          ORG 1000H    ; Mensaje a transferir origen
MSJ      DB      "FACULTAD DE INFORMATICA"
FIN      DB      ?
NCHAR    DB      ?

          ORG 1500H    ; Memoria destino del mensaje
COPIA    DB      ?

          ORG 3000H      ; rutina de interrupción del CDMA
RUT_DMA: MOV AL,0FFH
          OUT PIC+1,AL    ; inhabilita las interrupciones del
                          ; PIC

          MOV BX,OFFSET COPIA
          MOV AL,NCHAR
          INT 7            ; muestra en pantalla la cadena
                          ; copiada

          MOV AL,20H
          OUT PIC,AL      ; EOI
          IRET
```

Un dato más : la línea tc del CDMA es la conectada a la int3 del PIC. Se produce una interrupción cuando el reg CONT llega a cero: finalización de la transferencia. Por medio de la línea hrq el CDMA le pide a la CPU el bus para realizar la transferencia. La CPU contesta por la línea hlda. Entonces durante la transferencia por ej. : se lee un byte de la dirección 1000H de memoria, este dato viaja al CDMA y se almacena en un reg DATO del CDMA (actúa como CPU) y luego se almacena en la dirección 1500H (memoria destino).

2) En este ej. también están las tres etapas con las variantes que requiere el caso. No se escribe en los reg RDH y RDL, porque el destino no es una posición de memoria sino un periférico. El DMA trabaja en modo de demanda. Hay que configurar al HAND para que trabaje en modo de int.

El pedido sería : el HAND envía una señal por la línea int conectada a la línea dreq del CDMA. El CDMA por la línea hrq conectada a hold de la CPU le pide el bus a ésta. Cuando se lo da contesta con hlda. Aquí el CDMA comienza la transferencia (un byte) y para leer en memoria activa aen=address enable (dir válida) y memr= memoria read

(lectura en memoria). Luego activa iow=input-output write (escritura en dispositivo de i/o) y el dato va al HAND. Con la línea dack el CDMA le avisa al HAND que está completando la transferencia. Así dato por dato. Cuando el reg CONT llega a 0, envía por la línea tc una int al PIC línea int3, para finalizar la transferencia.