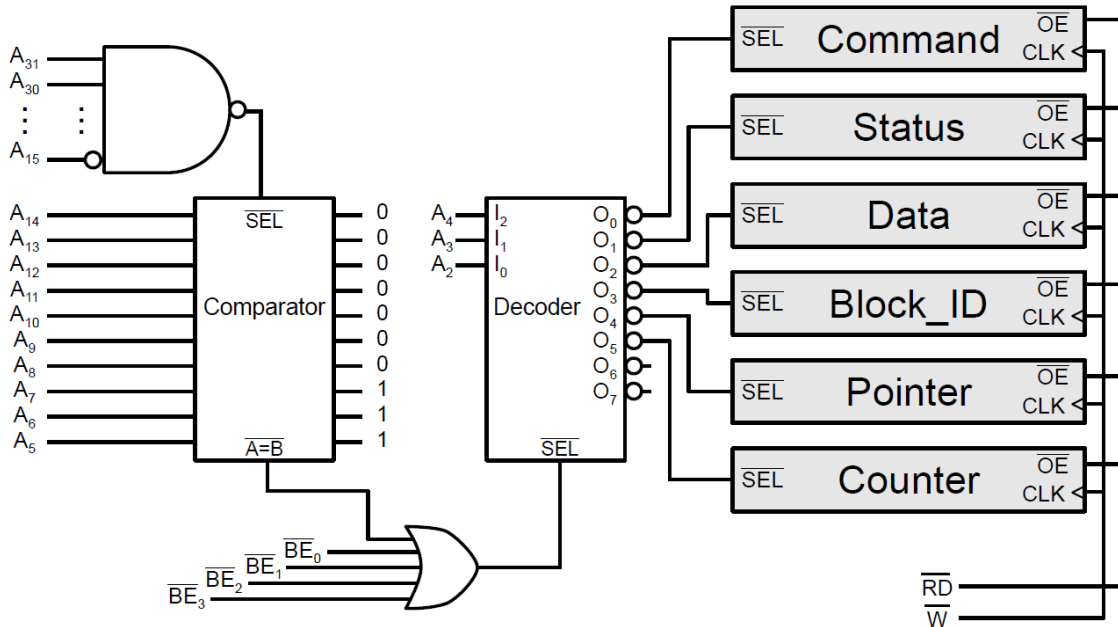


ESTRUCTURA DE COMPUTADORES

Tema 9/Ejercicio de clase_1

La siguiente figura muestra el adaptador de cierto **dispositivo de bloques**, el cual puede emplear los mecanismos de PIO/DMA para realizar las transferencias de datos:



STATUS:

- **R:** (bit 0) se establece a 1 por el hardware cuando hay disponible un nuevo dato en el registro de Datos (Modo PIO) o cuando finaliza la transferencia (modo DMA).

COMMAND:

- **PIO/DMA:** (bits 1 y 0) a 00 el interfaz se configura como **PIO**; a 11 se configura como **DMA**.
- **A:** (bit 2) a 1 ordena el inicio de la transferencia a/desde disco
- **R/W:** (bit 3) a 0, se transfiere de disco a memoria (Lectura); a 1, de memoria a disco (Escritura)
- **IE:** (bit 4) a 1 habilita la interrupción int0*. IE=1 se asocia con que el periférico está preparado para iniciar la transferencia (caso de **PIO**) o que la transferencia ha finalizado (caso de **DMA**).
- **CL:** (bit 7) a 1 hace R=0 (cancelando la interrupción int0* si la hay)

Asumiendo que **los seis registros del adaptador son de 32 bits de tamaño**, se pide:

a) ¿Cuál es la dirección base (DB) de este dispositivo?

0xFFFF00E0

b) Examinando las conexiones de las líneas A4, A3 y A2, indíquense las direcciones en que se ubican cada uno de los 6 registros del adaptador, expresándolas en términos de DB+<desplazamiento>.

Command: DB	Block_ID: DB + 12
Status: DB + 4	Pointer: DB + 16
Data: DB + 8	Counter: DB + 20

c) **Transferencia de un solo carácter (modo PIO):** Escríbase el código para consultar el dispositivo hasta que esté preparado, y entonces proceder a transferir la palabra del registro de Datos a la variable *Nuevo_Dato* del programa.

```

bucle:    la $t0, 0xFFFF00E0
          lw $t1, 4($t0)      # Bucle de consulta de estado (bit R)
          andi $t1, $t1, 1
          beqz $t1, bucle
          li $t1, 0x80        #Cancelación de R. Se ha hecho con <li> porque sabemos con total certeza cómo
          sw $t1, 0($t0)      #queremos que quede el resto de bits del registro Status
          lw $t1, 8($t0)
          sw $t1, Nuevo_Dato

```

- d) Escribese el código para configurar y llevar a cabo una **transferencia de bloque en modo PIO** desde el dispositivo a memoria (transferencia de lectura). Se desea leer el bloque cuyo identificador (ID) es 0x44442222. El tamaño del bloque es 1024 bytes, aunque se debe tener en cuenta que en cada ciclo se transfiere una palabra completa de 32 bits (4 bytes). Asimismo, se desea almacenar el bloque en la dirección de memoria etiquetada como *Mem_Block*. La sincronización se realiza **por consulta de estado**.

```

# Código de configuración de transferencia bloque en modo PIO y sincronización por consulta de estado
la $t0, 0xFFFF00E0
li $t1, 0x44442222
sw $t1, 12($t0)          #Carga ID del bloque en Block_ID
la $a0, Mem_Block        #Carga el puntero a memoria en $a0
li $a1, 256              #Carga el contador de ciclos de transferencia (1024B/4B=256 ciclos) en $a1
li $t1, 0x4              #(PIO/DMA= 00; A=1; R/W= 0; IE= 0)
sw $t1, 0($t0)           #Inicializa registro COMMAND

# Código de sincronización por consulta de estado previa a iniciar la transferencia de bloque
bucle1: lw $t1, 4($t0)      # Bucle de consulta de estado (bit R)
        andi $t1, $t1, 1
        beqz $t1, bucle1
        li $t1, 0x80        #Cancelación de R
        sw $t1, 0($t0)

# Código de transferencia de bloque
bucle2 : lw $t1, 8($t0)      #Lee word del registro Data
        sw $t1, 0($a0)      #Lo escribe a memoria en la dirección apuntada por $a0
        addi $a0, $a0, 4    #Incrementa el puntero $a0 en 4
        subi $a1, $a1, 1    #Decrementa contador $a1 en 1
        bnez $a1, bucle2    #Continúa en el bucle de transferencia mientras $a1<>0

```

- e) Escribese el código para configurar una transferencia **DMA** desde el dispositivo a memoria (transferencia de lectura). Como antes, se desea leer el bloque cuyo identificador (ID) es 0x44442222. El tamaño del bloque es 1024 bytes, aunque téngase en cuenta que en cada ciclo que se transfiere una palabra completa de 32 bits (4 bytes). Asimismo, se desea almacenar el bloque en la dirección de memoria etiquetada como *Mem_Block*. La sincronización se realiza **por interrupción**. *Nota: El contador se decrementa automáticamente en 1 con cada ciclo de transferencia*

```

# Código de configuración de transferencia bloque en modo DMA y sincronización por interrupción
la $t0, 0xFFFF00E0
li $t1, 0x44442222
sw $t1, 12($t0)          #Carga ID del bloque en Block_ID
la $t1, Mem_Block
sw $t1, 16($t0)          #Carga el puntero a memoria en Pointer
li $t1, 256
sw $t1, 20($t0)          #Carga el contador de ciclos de transferencia (1024B/4B=256 ciclos) en Counter
li $t1, 0x17             #(PIO/DMA= 11; A=1; R/W= 0; IE= 1)
sw $t1, 0($t0)           #Inicializa registro COMMAND

# Al ser una transferencia DMA se requiere el empleo de registros (Pointer y Counter), en el propio interfaz ,
# dedicados a realizar la transferencia de forma automática, sin intervención del software, más allá del código de
# configuración previo

```