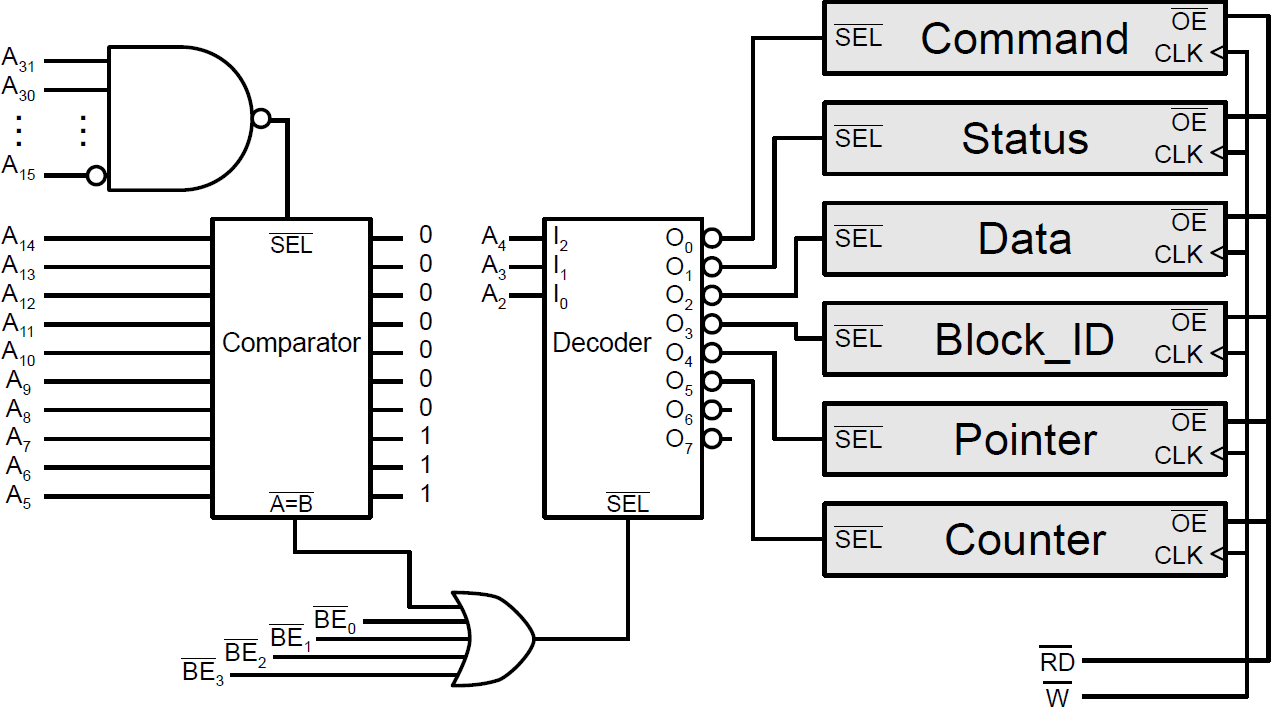
**ESTRUCTURA DE COMPUTADORES**

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:

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

0xFFFF00E0

1. 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>.

Block\_ID: DB + 12 Pointer: DB + 16 Counter: DB + 20

Command: DB Status: DB + 4 Data: DB + 8

1. **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.

la $t0, 0xFFFF00E0

**# Bucle de consulta de estado (bit R)**

bucle:

lw $t1, 4($t0)

andi $t1, $t1, 1 **# Comprobar que bit R = 1 (tenga algo que transferir)**

beqz $t1, bucle **# Si R ≠ 1 -> realizar bucle de nuevo**

**# Transferencia simple**

lw $t1, 8($t0)

sw $t1, Nuevo\_Dato

**# Cancelación**

li $t1, 128 **# Cancelación de R. Poner a 1 bit CL de COMMAND**

sw $t1, 0($t0)

1. Escríbase 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 **# carga de la DB**  li $t1, 0x44442222 **# coordenadas sector**  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, 4 **# (PIO/DMA = 00; A = 1; R/W = 0; IE = 0) – bits de control**  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, 128 **# Cancelación de R (bit CL = 1 de COMMAND)**  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 | **# Continua en el bucle de transferencia mientras $a1 ≠ 0** |

1. Escríbase 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.*

# 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 previa

#Carga el puntero a memoria en **Pointer**

# Carga el contador de ciclos de transferencia (1024B/4B=256 ciclos) en **Counter**

# (PIO/DMA= 11; A=1; R/W= 0; IE= 1)

# Inicializa registro COMMAND

#Carga ID del bloque en Block\_ID

sw $t1, 12($t0)

la $t1, Mem\_Block sw $t1, 16($t0)

li $t1, 256

sw $t1, 20($t0) li $t1, 23

sw $t1, 0($t0)

# Código de configuración de transferencia bloque en modo DMA y sincronización por interrupción la $t0, 0xFFFF00E0 # carga de la DB

li $t1, 0x44442222 # coordenadas sector