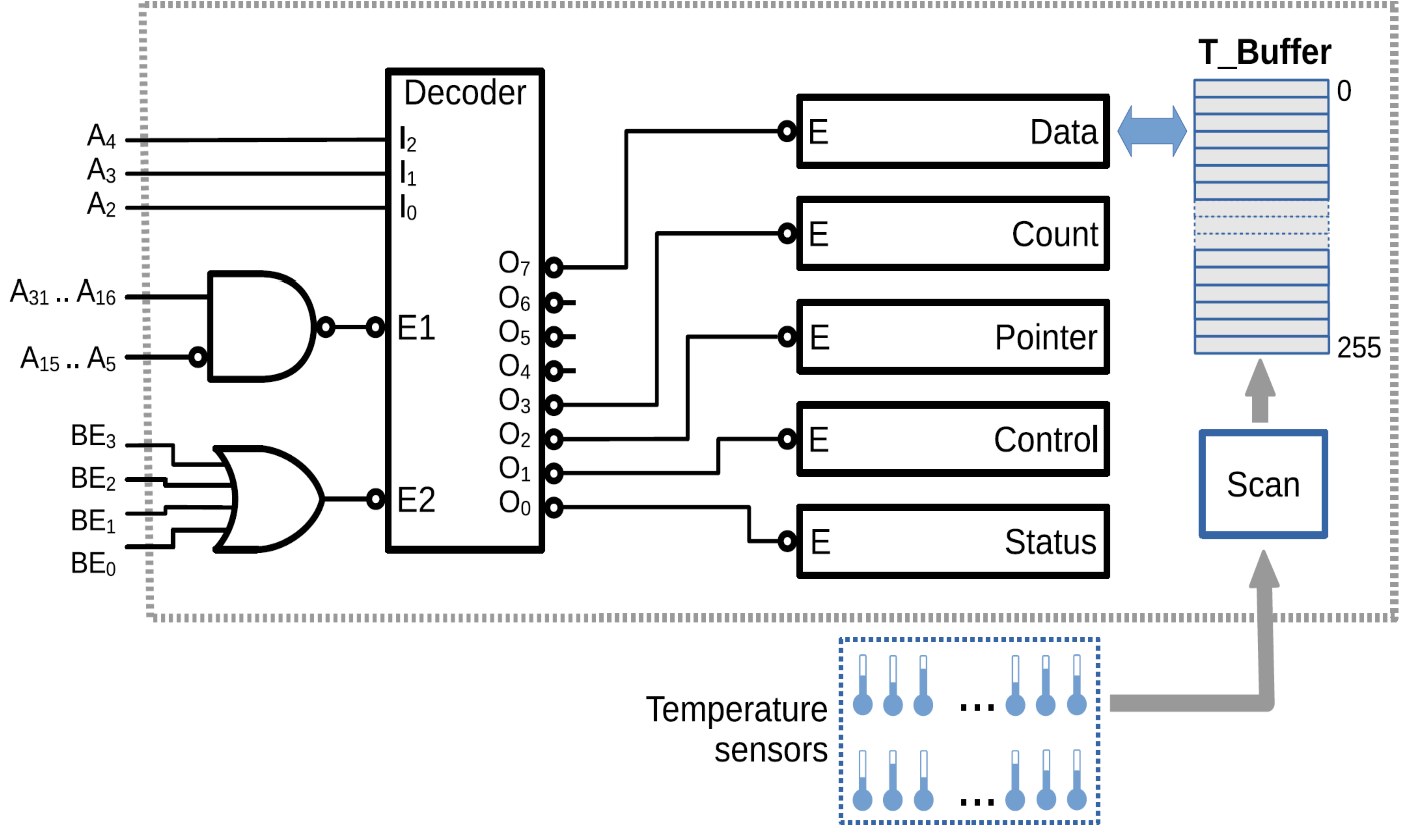
**1**

La figura muestra los detalles del adaptador de un dispositivo de adquisición de temperaturas conectado a un procesador MIPS R2000. Dicho dispositivo puede ser configurado para registrar las temperaturas de hasta 256 sensores ubicados en las estancias de un edificio. Los valores de temperatura se hallan codificados como enteros con signo de 32 bits. Todos los registros del interfaz son de tamaño 32 bits.



La operación del dispositivo se inicia mediante una orden de *UPDATE*, escrita en el registro de Control. En respuesta a dicha orden, el sistema escaneará las medidas de temperatura de hasta 256 sensores y las almacenará en un buffer interno (*T\_Buffer*), en posiciones consecutivas. Una vez *T-Buffer* se actualice con las medidas de temperatura, existen dos posibles modos de transferir su contenido al espacio de memoria de usuario: modos *PIO* y *ADM*. En modo *PIO*, el dispositivo se encontrará *Ready* tan pronto *T\_Buffer* se actualice, de modo que la transferencia de datos desde *T\_Buffer* a memoria de usuario debe hacerse por programa. En modo *ADM*, una vez *T\_Buffer* se actualiza, los datos se transfieren por *ADM* a un buffer en memoria de usuario, de modo que el dispositivo se encontrará *Ready* al término de la transferencia por *ADM*. El adaptador contiene los siguientes registros:

**STATUS** (lectura y escritura)

* **RDY** (bit 0). Bit *Ready*. Es puesto a 1 por el hardware dependiendo del modo de transferencia o Modo *PIO*: una vez que *T\_Buffer* ha sido actualizado con las medidas de temperatura o Modo *ADM*: cuando el contenido de *T\_Buffer* ha sido totalmente transferido a la memoria de usuario por *ADM*.

Para hacer RDY = 0, basta con escribir un 0 sobre dicho bit (obsérvese que no hay un bit específico de cancelación en este interfaz).

* **IEN** (bit 1). Bit *Interrupt Enable*. Si IEN = 1, la puesta a 1 del bit RDY causa la activación de INT3\*.

**CONTROL** (solo escritura)

* **UPD** (bit 0). Bit *UPDATE*. Cuando se pone a 1 por software, el sistema inicia el registro de temperaturas de los distintos sensores.
* **MOD** (bit 1). Bit *MODO*. A 0 para *PIO* y a 1 para *ADM*. Se ignora cuando UPD = 0.

**POINTER** (solo escritura)

En modo *ADM*, este registro debe actualizarse con la dirección de inicio del buffer en memoria de usuario, donde el contenido de *T\_Buffer* es transferido por *ADM*. No tiene ningún uso en modo *PIO*.

**COUNT** (solo escritura)

Debe actualizarse con el número de sensores de temperatura a registrar por el sistema con cada orden *UPDATE*.

**DATA** (lectura y escritura)

Este registro (contiene un valor de temperatura cada vez) se usa solo en modo *PIO*, durante la transferencia por programa del contenido de *T\_Buffer*. El registro almacena temporalmente el siguiente valor de temperatura a ser transferido desde *T\_Buffer*. Cada vez que se lee el registro, éste se actualiza automáticamente con el siguiente valor de temperatura almacenado en *T\_Buffer*.

1. Calcule la dirección base del adaptador y los desplazamientos de cada uno de los registros (**en hexadecimal**).

|  |  |  |
| --- | --- | --- |
| **Dirección Base (DB):** |  | 0xFFFF0000 |

**¿Los del decodificador siempre se ponen a 0? -> Sí.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | **Registro** | **Desplazamiento** | | Count | BA + 0x0C = BA + 12 | | Data | BA + 0x1C = BA + 28 |  |  |  | | --- | --- | | **Registro** | **Desplazamiento** | | Status | BA | | Control | BA + 0x04 | | Pointer | BA + 0x08 | |

1. ¿De acuerdo a su descripción, se trata de un dispositivo de bloques o de caracteres? Justifique la respuesta.

|  |
| --- |
| Se trata de caracteres, ya que no tenemos el registro de bloque. |

1. Escriba el código de la función del sistema encargada de dar una orden *UPDATE* para escanear *N* sensores y, una vez actualizado *T\_Buffer*, almacenar su contenido en la dirección de memoria proporcionada por el usuario **empleando modo *PIO***. Asuma que *N* se pasa en el registro $a0 y el que el puntero al buffer de memoria de usuario lo hace en $a1. La transferencia se debe **sincronizar mediante *consulta de estado***.

|  |
| --- |
| **PIO\_Update:**  la $t0, 0xFFFF0000 **# carga de la DB**  sw $a0, 12($t0) **# Actualización de Count con N transferencias**  sw $zero, 0($t0) **# inhabilitación de las interrupciones (RDY = 0)**  li $t1, 1 **# UPD = 1**  sw $t1, 4($t0) **# actualización de UPD en Control**  **# Bucle de consulta de estado**  **Bucle:**  **# Código transparencias -> Comprobará hasta que R = 1 (haya datos a transferir)**  lw $t1, 0($t0)  andi $t1, $t1, 1 **# cogemos el bit R**  beqz $t1, bucle **# Si no tiene datos a transferir, vuelve al bucle**  **Transferencia:**  **# Código transparencias - transferencia**  lw $a2, 28($t0) **# acceso a los datos**  sw $a2, 0($a1) **# almacenamiento de los datos en el búfer del usuario**  **#** **Código transparencias – actualización de punteros**  addi $a1, $a0, 4 **# 4 debido a que son registros de 32 bits (4B)**  addi $a0, $a0, -1 **# Disminuimos el número de transferencias**  bnez $a0, Transferencia  b retexc |

1. Escriba el código de la función del sistema encargada de dar una orden *UPDATE* para escanear *N* sensores y, una vez actualizado *T\_Buffer*, almacenar su contenido en la dirección de memoria proporcionada por el usuario **empleando modo *ADM***. Asuma que *N* se pasa en el registro $a0 y el que el puntero al buffer de memoria lo hace en $a1. La espera a que la transferencia por *ADM* se complete, se hará asumiendo un **entorno multitarea**.

|  |
| --- |
| DMA\_Update:  **# Parámetros de transferencia**  la $t0, 0xFFFF0000 **# carga de la DB**  sw $a0, 12($t0) **# Actualización de Count con N transferencias**  sw $a1, 8($t0) **# Actualización del puntero**  **# Habilitación de las interrupciones**  li $t1, 2 **# IEN = 1**  sw $t1, 0($t0) **# Habilitamos las interrupciones**  li $t1, 3 **# MOD = 1 (DMA), UPD = 1**  sw $t1, 4($t0) **# Actualización de UPD en Control**  **# Suspender al proceso**  jal suspende\_este\_proceso    b retexc |