

Ejercicio 1 (4 Puntos)

Sea un periférico de almacenamiento no volátil basada en memorias Flash. Este periférico contiene una memoria de 64GB organizada en 16 Mbloques de 4KB cada uno ($2^{24} \times 2^{12}$ Bytes). El acceso a dicha memoria se realiza mediante una interfaz específica, compuesta por 6 registros como se indica en la figura 1 adjunta.

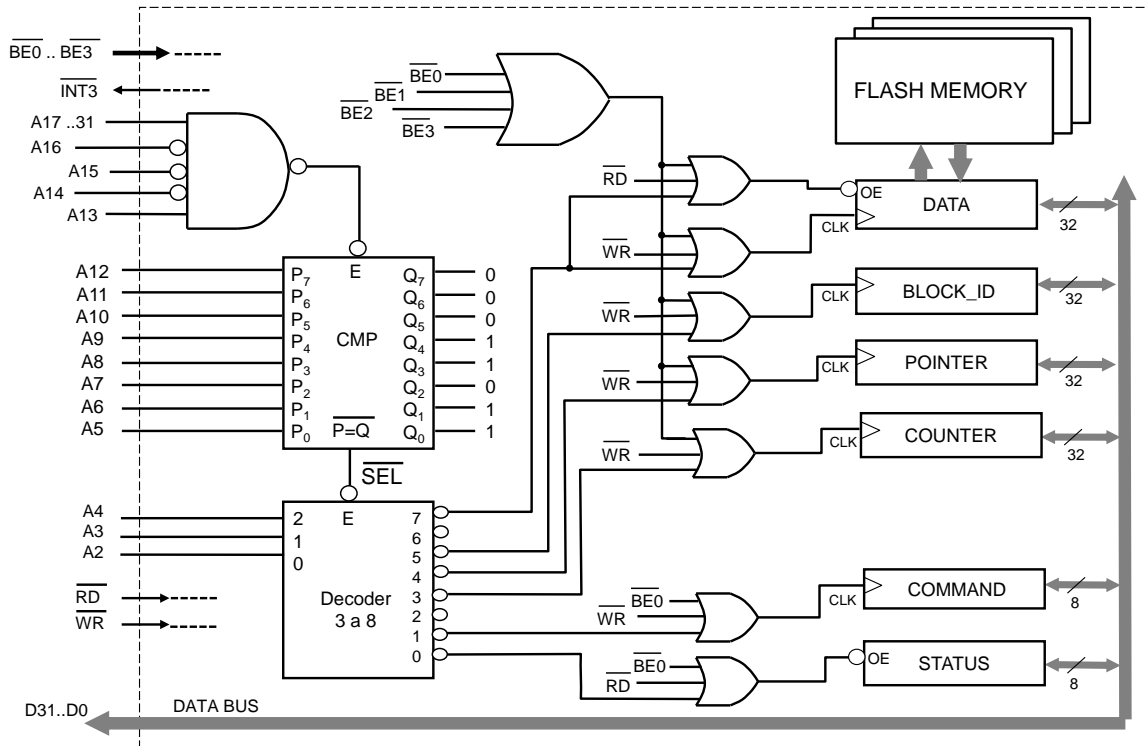


Figura 1 – Adaptador de la memoria flash para el MIPS R2000.

Indique:

- (0.5 puntos) Cuál es la dirección base de la interfaz.

Dirección Base (DB)=

- (0.5 puntos) Indica la dirección de los seis registros (en formato BASE + n) y el tipo de operación (Lectura/Escritura) que se puede realizar sobre ellos.

REGISTRO	Dirección	Operación
STATUS	DB +	
COMMAND	DB +	
COUNTER	DB +	
POINTER	DB +	
BLOCK_ID	DB +	
DATA	DB +	

- (0.5 puntos) Si la dirección del registro de datos de la interfaz es *Dir_DATA*, indique qué pasaría si se ejecuta la siguiente secuencia de instrucciones :

```
.kdata
buffer: .space 512

.ktext
.....
la $t0, Dir_DATA
lb $t1, 0($t0)
sb $t1, buffer
```

4. (0.5 puntos) Suponiendo que el MIPS tuviera un mapa de E/S separado del mapa de memoria, indique qué modificaciones haría en el esquema de la interfaz para que el acceso a los registros se realizara en el mapa de E/S y no en el mapa de memoria. Dibuje lo que sea necesario sobre el esquema.
5. (0.5 puntos) El contenido de los registros del periférico del ejercicio anterior es el siguiente:
- Registro **STATUS**: indica el estado del periférico. Bits:
 - Bit 4: R **Ready** – Preparado.
 - Para transferencias por programa este bit se activa (R=1) cuando hay un dato disponible en el registro DATA.
 - Para órdenes leer/escribir en modo DMA, se activa (R=1) cuando la transferencia por acceso directo a memoria (lectura o escritura) ha terminado.
 - Registro **COMMAND**: programa la operación sobre el periférico. Bits:
 - Bits 0...3. Orden
 - 0x0: No operación
 - 0x1: Leer bloque en modo DMA
 - 0x2: Escribir bloque en modo DMA
 - Otros códigos están reservados
 - Bit 7: E Enable - Habilitación de la interrupción INT3
 - Si este bit se activa (E=1) se genera la interrupción INT3 cada vez que se activa el bit R de preparado.
 - Bit 5: C Clear - Borrado.
 - Activando este bit (C=1) se vuelve a poner a cero el bit R.
 - Registro **BLOCK_ID**: Identificador del bloque de memoria a leer/escribir.
 - Registro **POINTER**: Para operaciones DMA indica la dirección inicial de memoria principal donde leer o escribir el bloque.
 - Registro **COUNTER**: Indica el número de palabras (WORDS) que se van a transferir en las operaciones de lectura/escritura.

Escriba un fragmento de código para leer por programa una palabra del registro de datos (DATA), realizando la sincronización por consulta de estado del bit R (Preparado) del registro de estado (STATUS), y transferirla a la variable de memoria *New_Data*.

Se desea escribir un grupo de funciones del sistema para manejar este periférico, como se indican en la tabla adjunta:

Función	Código	Parámetros de entrada	Parámetros de salida
<i>Read_DMA</i>	\$v0=900	\$a0: Puntero al buffer de memoria	----
<i>Write_DMA</i>	\$v0=901	\$a1: N° del bloque (Block_id) \$a2: N° de palabras a transferir	----

6. (1 punto) Escriba el código correspondiente a la función '*Read_DMA*', que debe realizar una lectura de un bloque de la memoria flash mediante ADM, cuya terminación se indicará mediante la interrupción INT3. Se debe programar convenientemente la operación, habilitar la interrupción y suspender el proceso.

Read_DMA:

b suspende_este_proceso
b retexc

7. (0.5 puntos) Escriba la rutina de la interrupción INT3. Como sólo se desea que la interrupción se produzca una vez, se debe inhabilitar la interrupción en la interfaz del dispositivo.

Int3:

b activa_proceso_en_espera
b retexc

Ejercicio 2 (4 Puntos)

La figura 2 muestra esquemáticamente la instalación de un montacargas en un edificio de dos plantas. Los sensores TOP y BOTTOM detectan la llegada del montacargas a cada una de las dos plantas. El motor M en la parte superior es el encargado de subir y bajar la cabina. La operación del montacargas es controlada por una palanca de tres posiciones (UP – STOP – DOWN). En las posiciones UP y DOWN se sube o baja la cabina respectivamente. Mover la palanca a la posición de STOP detiene el motor, si éste se está moviendo, y no tiene efecto en otro caso. La palanca sólo puede estar en una de las tres posiciones.

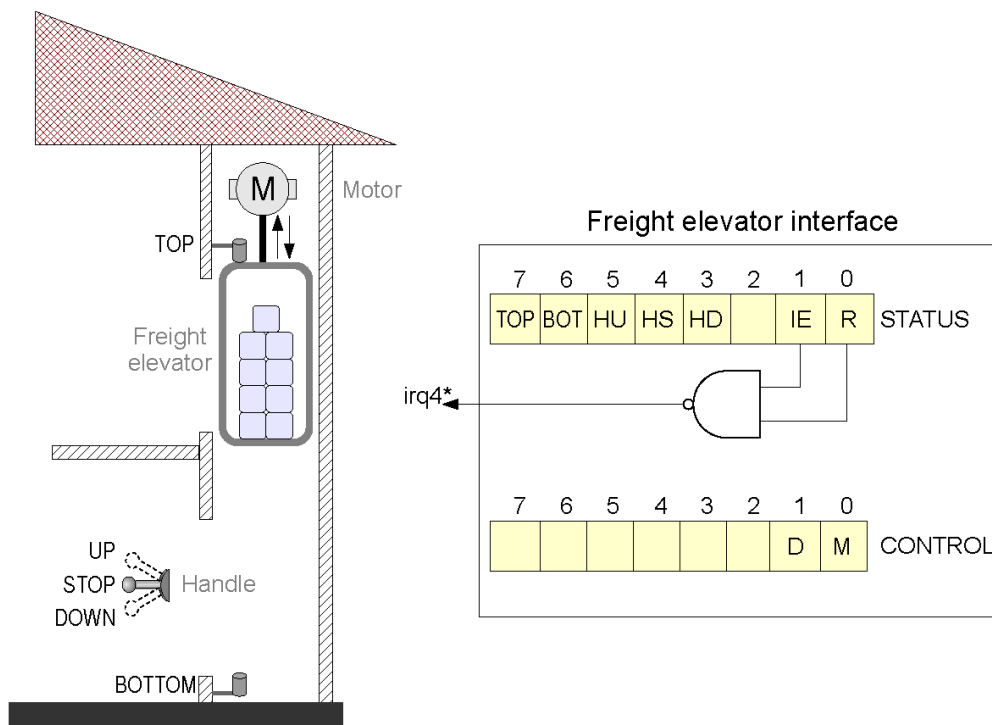


Figura 2.- Esquema del montacargas (izquierda) y registros de la interfaz (derecha).

El montacargas está conectado a un computador MIPS R2000 a través de una interfaz apropiada. Los registros de esta interfaz se muestran en la parte derecha de la figura 2, y su descripción es la siguiente:

- Registro **STATUS**: (Lectura/Escritura. Dirección 0xFFFF0010):
 - Bit 7 - TOP: la interfaz lo pone a 1 cuando la cabina llega al piso superior y a 0 en otro caso.
 - Bit 6 - BOT: La interfaz lo pone a 1 cuando la cabina llega al piso inferior y a 0 en otro caso.
 - Bit 5 - HU: La interfaz lo pone a 1 cuando la palanca está en posición UP y a 0 en otro caso.
 - Bit 4 - HS: La interfaz lo pone a 1 cuando la palanca está en posición STOP y a 0 en otro caso.
 - Bit 3 - HD: La interfaz lo pone a 1 cuando la palanca está en posición DOWN y a 0 en otro caso.
 - Bit 1 - IE: Al poner este bit a 1 se habilita la interrupción (INT4) de la interfaz.
 - Bit 0 - R: Se pone a 1 cuando alguno de los dos sensores (TOP, BOTTOM) se activa, lo cual indica que la cabina ha llegado a la planta correspondiente. Si el bit IE=1 la interrupción INT4 se activará. El bit R debe cancelarse por programa, escribiendo un 0. Una vez cancelado, el bit R se mantiene a cero aunque la cabina permanezca detenida en la planta y el sensor siga activo.

Al escribir en los bits IE y R se debe tener el cuidado de no alterar el resto de bits del registro, que deben quedar inalterados.

- Registro **CONTROL**: (Escritura. Dirección 0xFFFF0014):
 - Bit 0. M: Activa el motor. M=1 pone en marcha el motor y M=0 lo para.
 - Bit 1. D: Dirección del movimiento. D=1 para subir y D=0 para bajar.

1. (0.4 puntos) De acuerdo con la descripción de la interfaz del montacargas, indique si la sincronización de la interfaz debe ser sólo por prueba de estado, sólo por interrupciones o con cualquiera de las dos técnicas. Justifique la respuesta.

2. (0.4 puntos) El código de inicio del sistema, justo antes de saltar al programa de usuario (main), incluye el siguiente fragmento:

```

.....
1      la $t0, 0xFFFF0010
2      lb $t1, 0($t0)
3      ori $t1, $t1, 0x2
4      sb $t1, 0($t0)
5      li $t0, 0x3402
6      mtco $t0, $13
7      jal main
.....

```

¿Este fragmento habilita la gestión del montacargas mediante interrupciones? En caso de que esto no fuera así indique qué habría que hacer para que la habilitación tuviera lugar. La figura 3 describe el contenido de los registros de Causa y Estado del MIPS R2000.

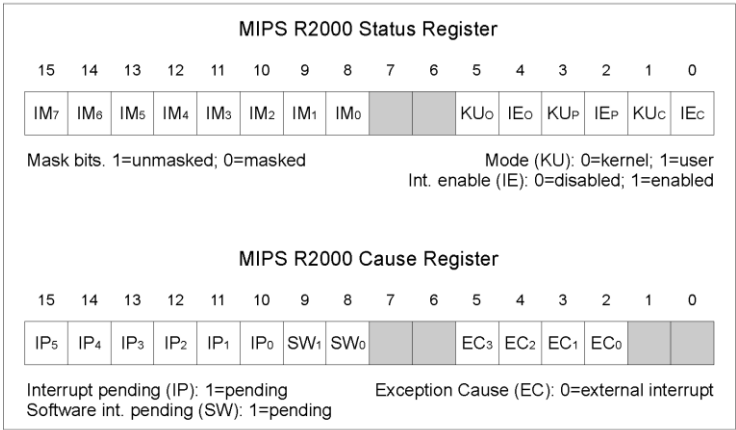


Figura 3 – Registros de Estado (\$12) y de Causa (\$13) del MIPS R2000.

3. (0.4 puntos) El siguiente fragmento de código del manejador de excepciones del MIPS es el encargado de decidir qué interrupción se debe atender. Indique cuál de las cuatro etiquetas (`cod_int{W | X | Y | Z}`) se corresponde con la interrupción del montacargas.

4. (0.4 puntos) Escriba el código para las tres subrutinas `move_up`, `move_down` y `stop`, que permiten subir, bajar o parar el montacargas. Las subrutinas residen en el kernel y sólo pueden ser llamadas con la CPU en modo supervisor. Sólo se permite el uso de los registros `$t0` y `$t1` dentro de este código.

move_down:

stop:

5. (2 puntos) Escriba el código de la rutina de servicio del montacargas. El comportamiento requerido es que el sistema actúa apropiadamente ante cambios en la palanca de control y a la activación de los sensores TOP y BOTTOM. Mover la palanca a la posición UP (subir) hace que la cabina suba al piso superior y se pare allí, excepto si el sensor TOP ya está a 1. Similarmente con la palanca en la posición DOWN (bajar). Mover la palanca a la posición STOP hace que la cabina se pare allí donde esté. La acción correspondiente a la activación de los sensores TOP y BOTTOM es, simplemente, parar el motor. Puede usar las subrutinas (move_up, move_down, stop) del ejercicio anterior. Asuma que el manejador de excepciones preserva los registros \$t0 .. \$t4 y \$s0..\$s4.

```
elevator_int: ...
check_HU:      if HU=1 then          # Palanca arriba
                if TOP=1 then        # ¿la cabina ya está arribar?
                    b stop_M         # Si, para motor
                else
                    move elevator up # No, subir cabina
                    b continue
                end if
            end if
check_HD:      if HD=1 then          # Palanca abajo
                if BOT=1 then        # ¿la cabina ya está abajo?
                    b s top_M         # Si, parar motor
                else
                    move elevator down # No, bajar cabina
                    b continue
                end if
            end if
stop_M: stop motor
continue:      .....
```

6. (0.4 puntos) Escriba el código para una función del sistema con la siguiente especificación:

Función	Índice	Argumentos	Resultado
<i>Elevator_position</i>	\$v0 = 333	-----	\$v0 = Código de posición

El código de posición devuelto por la función indica si el montacargas está en la planta baja (valor 1), en la planta alta (valor 2) o en cualquier posición entre ambas plantas (valor 0). Observe que estos valores están inspirados en la codificación de los bits TOP y BOTTOM del registro de estado del montacargas. Observe también que estos bits no pueden estar nunca ambos a 1.

Ejercicio 3 (2 Puntos)

La figura 4 muestra la estructura de buses y periféricos de un computador. Se desea transferir un archivo de imagen de 1 GB ($\text{GB} = 10^9$) desde la memoria SD de la cámara hasta uno de los discos del computador (*hdb* o *pendrive*). Las transferencias siempre se hacen por ADM (cámara → MEM, MEM → disco). Las transferencias siempre se hacen por ADM (cámara → MEM, MEM → disco).

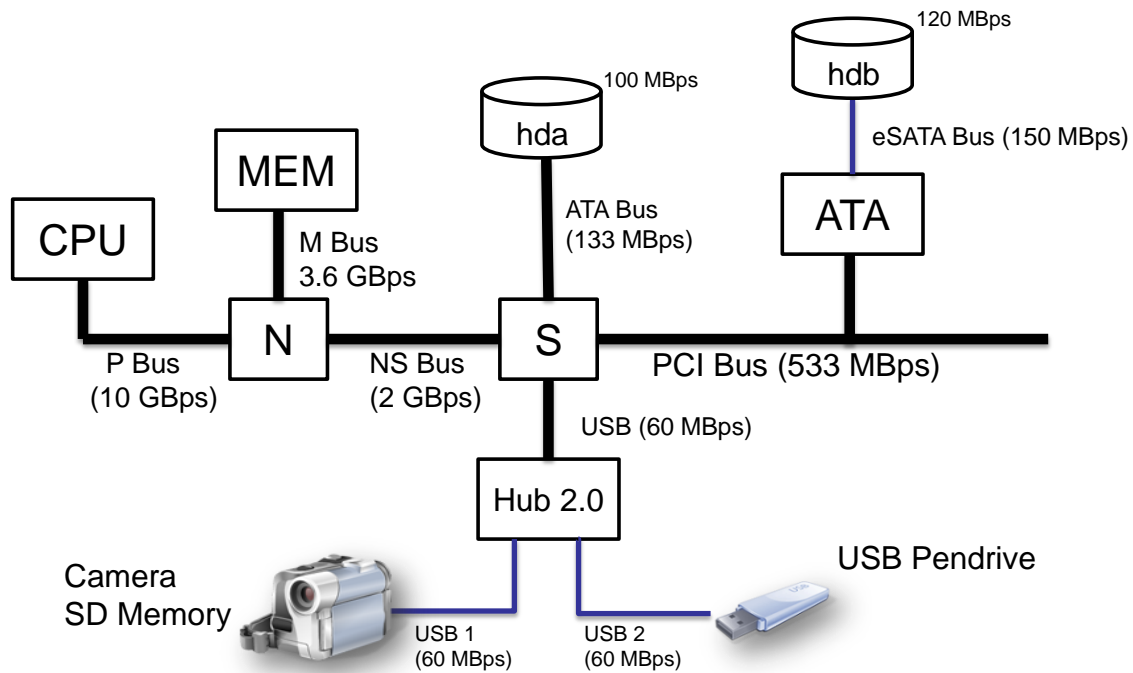


Figura 4.- Estructura de buses y periféricos de un computador.

1. (0.75 puntos) Suponiendo que las transferencias se hacen en SERIE, es decir primero se lee el archivo completo a memoria y luego se escribe en el disco, indique el tiempo mínimo requerido para hacer la operación de transferencia desde la cámara al disco hdb.

Transferencia CAMARA → hdb

2. (0.75 puntos) Suponiendo que las operaciones se hacen de manera CONCURRENTE, es decir que simultáneamente se lee de la cámara y se escribe en el disco de destino, indique el tiempo mínimo requerido para hacer la transferencia del archivo desde la cámara al pendrive.

3. (0.5 puntos) En el caso del apartado anterior indique la ocupación de los buses (%) durante la transferencia