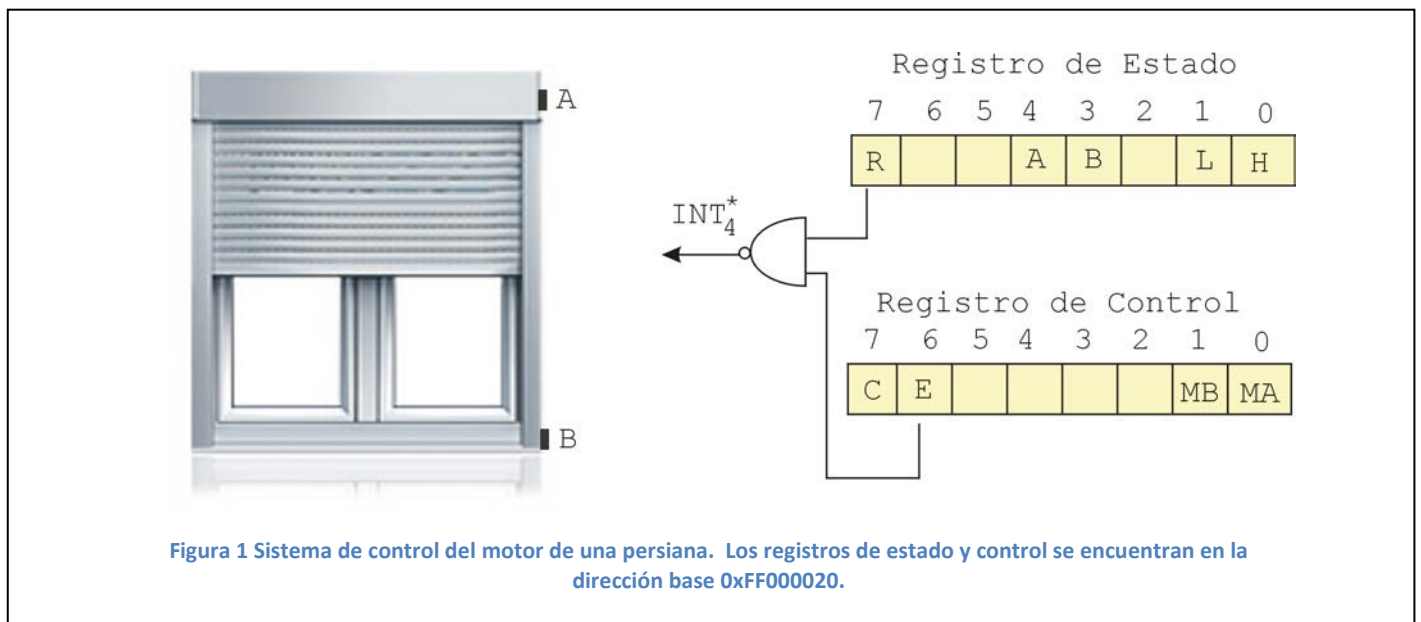


Ejercicio 1 (4 Puntos)

Las ventanas de un edificio están gobernadas por un sistema de control cuya interfaz muestra dos registros, (ver Figura 1). El sistema dispone de un sensor de humedad y un sensor de luz, y está conectado a un actuador que le permite subir o bajar la persiana de la ventana. Mediante dos sensores A y B se detecta la posición de la persiana, subida si A está a 1 y bajada si B está a 1. A y B nunca estarán activos a la vez, y pueden tener cero ambos a la vez cuando la persiana está en movimiento.



Este sistema está conectado a un computador MIPS R2000 a través de una interfaz apropiada. Los registros de esta interfaz se encuentran en la dirección base 0xFF000020 y su descripción es la siguiente:

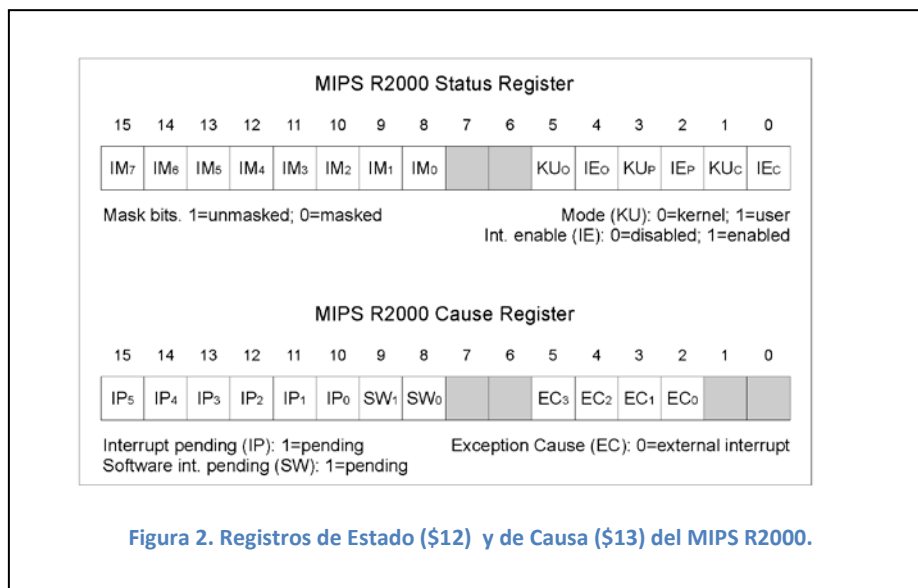
- Registro **ESTADO** (Sólo lectura):
 - Bit 0 - **H**: Se pone a 1 cuando detecta lluvia y se pone a 0 cuando deja de llover.
 - Bit 1 - **L**: Se pone a 1 cuando hay luz y a 0 cuando no la hay.
 - Bit 3 - **B**: Si está a 1 indica que la persiana está bajada, y a 0 en movimiento o arriba.
 - Bit 4 - **A**: Si está a 1 indica que la persiana está subida, y a 0 en movimiento o abajo.
 - Bit 7 - **R**: La interfaz lo pone a 1 cuando hay un cambio en alguno de los bits H o L.
- Registro **CONTROL** (Sólo escritura) :
 - Bit 0 - **MA**: Se pone a 1 para subir la persiana (si ya está subida no tiene efecto) y a cero no hace nada. El sistema detiene automáticamente el motor cuando ha terminado de subirla.
 - Bit 1 - **MB**: Se pone a 1 para bajar la persiana (si ya está bajada no tiene efecto) y a 0 no hace nada. El sistema detiene automáticamente el motor cuando ha terminado de bajarla.
 - Bit 6 - **E**: Se pone a 1 para activar la interrupción en la interfaz y a 0 para inhibirla. Cuando E es 1 y el bit R también, se activa la interrupción 4 del procesador (INT₄ *).
 - Bit 7 - **C**: Si se escribe un 1 entonces la interfaz pone el bit R a 0. Sirve por tanto para cancelar la petición de interrupción INT₄*.

1. (0.25 puntos) Programe las instrucciones que habilitan las interrupciones en la interfaz descrita.

```
la $t0, 0xFF000020
li $t1, 0x40          # bit E = 1
sb $t1, 0($t0)
```

2. (0.25 puntos) Programe el código que habilita la línea de interrupción 4 en el procesador y deja el procesador en modo usuario e interrupciones generales habilitadas. Los demás bits del registro de estado del procesador deben quedar inalterados. En la Figura 2 aparece información del formato de los registros del coprocesador cero.

```
mfc0 $t0, $12      # lee el registro de estado
ori $t0, $t0, 0x1003 # IM4=KUC=IEC=1
mtc0 $t0, $12      # escribe en reg. estado
```



3. (1 punto) Programe una rutina que realice el acceso a dicha interfaz por el mecanismo de consulta del estado. El comportamiento debe ser el siguiente:

- Cuando el bit R se activa:
 - SI hay luz ENTONCES
 - SI no llueve ENTONCES
 - Subir persiana
 - SI NO
 - Bajar persiana
 - FIN SI
 - SI NO
 - Bajar persiana
 - FIN SI

```
Ventana:    la $t0, 0xFF000020
bucle:      lb $t1, 0($t0)           # leer estado
            andi $t2, $t1, 0x80     # Filtrar el bit R
            beqz $t2, bucle
            andi $t2, $t1, 2        # Filtro el bit L
            beqz $t2, Sino          # Si L = 0 Salto a Sino
Entonces:   andi $t2, $t1, 1        # Filtro el bit H
            bnez $t2, Sino          # Si H = 1, llueve, y salto a Sino2
Entonces2:  li $t1, 0x81            # C=MA=1, Subir persiana y cancelar
            sb $t1, 0($t0)
            jr $ra
Sino:       li $t1, 0x82            # C=MB=1, Bajar persiana y cancelar
            sb $t1, 0($t0)
            jr $ra
```

4. (1 punto) Se necesita disponer de un conjunto de llamadas al sistema para programar el acceso a las persianas desde una consola de control. Programe la siguiente llamada al sistema:

| Función | Índice | Argumentos | Resultado |
|------------------------|-----------|------------|---|
| <i>Estado_persiana</i> | \$v0 = 30 | ----- | \$v0 = 0 (Intermedio) \$v0 = 1 (Bajada) \$v0 = 2 (Subida) |

Tenga en cuenta que el contenido del registro de estado se puede leer en cualquier momento y mantiene la información de la persiana permanentemente actualizado. Programe estrictamente la sección del sistema que atiende esta llamada, suponga que empieza en la etiqueta Estado_persiana.

```
Estado_persiana:  la $t0, 0xFF000020
                  lb $t1, 0($t0)           # leer estado
                  andi $t0, $t1, 0x10       # Filtrar bit A
                  beqz $t0, VerB            # Si A= 0, Salto a VerB
                  li $v0, 2                 # Estaba subida
                  b retexc
VerB:             andi $t0, $t1, 0x08       # Filtrar bit B
                  beqz $t0, Medio          # Si B=0, Salto a Medio
                  li $v0, 1                 # Estaba bajada
                  b retexc
Medio:           li $v0, 0                  # Estaba intermedio
                  b retexc
```

Opción (2) Código más compacto:

```
Ventana:         la $t0, 0xFF000020
                  andi $t1, $t1, 0x18       # Filtrar bits A y B
                  srl $v0, $t1, 0x3        # Desplazar a derecha tres bits
                  b retexc
```

5. (1,5 puntos) Se sabe que el siguiente fragmento de código pertenece a la implementación de una llamada al sistema.

```
funcion_incognita: la $t0, 0xFF000020
                   lb $t1, 0($t0)
                   andi $t1, $t1, 1
                   bnez $t1, retexc
                   li $t1, 0x40
                   sb $t1, 0($t0)
                   jal suspende_este_proc # Suspende el proceso
                   b retexc
```

a) Dados las siguientes afirmaciones indique si son verdaderas o falsas:

| | |
|--|--------------|
| Esta rutina suspende el proceso si no llueve | V / F |
| Esta rutina no siempre suspende el proceso | V / F |
| Esta rutina es incorrecta porque no sincroniza | V / F |
| Esta rutina subiría la persiana y suspendería el proceso | V / F |

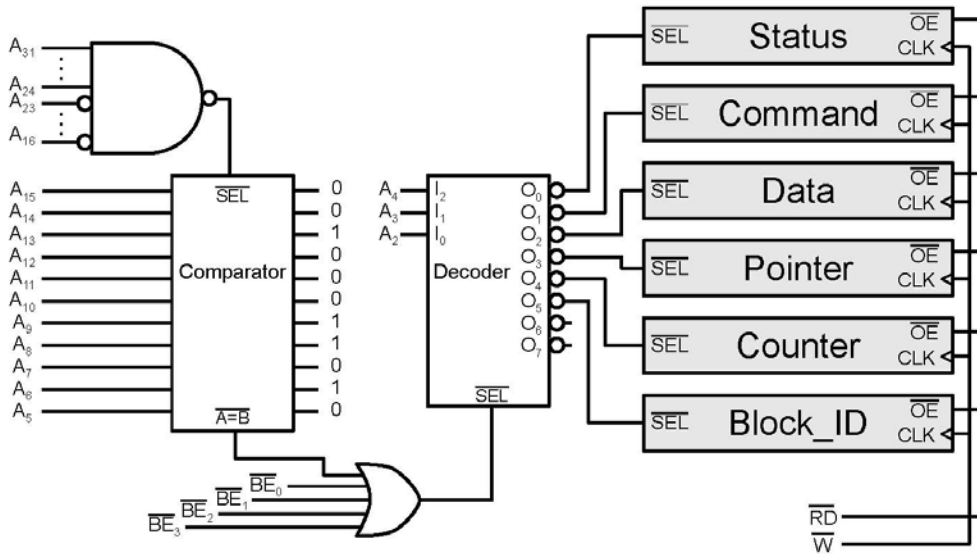
b) Programe el código de la rutina de interrupción que trataría la línea INT₄* del procesador para despertar al proceso dormido por la función incógnita anterior (como única acción, no se deben realizar las acciones del comportamiento de la pregunta tres). Indique exactamente qué se pretende hacer con cada instrucción. Para despertar el proceso utilice la subrutina *despierta_este_proc*. Observe que el proceso ha de ser despertado cuando NO ocurra el evento que ha hecho que se suspenda.

```

INT4:    la $t0, 0xFF000020
         lb $t1, 0($t0)           # leer estado
         andi $t1, $t1, 1         # Filtrar bit H
         beqz $t1, fin            # Si H = 0, salto a fin
         li $t1, 0x80             # C=1, E=0 Cancelar e inhibir interr.
         sb $t1, 0($t0)
         jal despierta_este_proc # Activa el proceso
         b retexc
fin:     li $t1, 0xC0             # C=E=1, cancelo pero interr. activas
         sb $t1, 0($t0)
         b retexc

```

Ejercicio 2 (3 Puntos)



La

Figura 3 muestra el adaptador de cierto dispositivo con acceso directo a memoria (ADM). Este dispositivo se encuentra conectado a un procesador MIPS R2000.

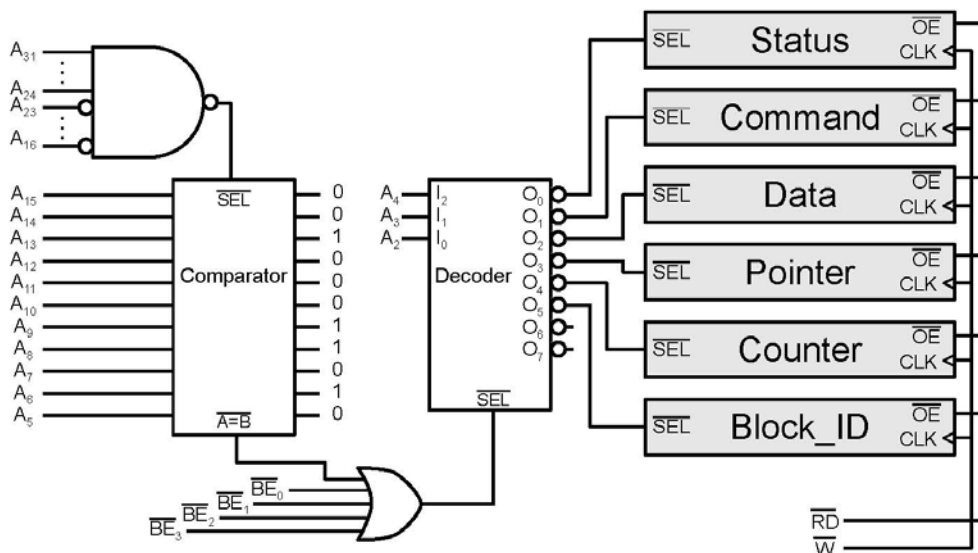


Figura 3. Adaptador de dispositivo con acceso directo a memoria (ADM)

Note que todas las líneas de dirección desde A₃₁ hasta A₂₄ están conectadas directamente a la puerta NAND, mientras que las líneas A₂₃ hasta A₁₆ están conectadas a través de un inversor. Los seis registros de la interfaz tienen 32 bits.

1. (1 punto) ¿Cuál es la dirección base de la interfaz?

Dirección Base (DB)= 0xFF002340

2. (0.6 puntos) Indica la dirección de los seis registros (en formato BASE + n), con el desplazamiento expresado en hexadecimal.

| | | | |
|---------|----------------------|----------|-----------------------|
| Status | BASE | Pointer | BASE+C ₁₆ |
| Command | BASE+4 ₁₆ | Counter | BASE+10 ₁₆ |
| Data | BASE+8 ₁₆ | Block_ID | BASE+14 ₁₆ |

3. (0.4 puntos) Asumiendo que el registro \$t0 tiene la dirección del registro *Status*, indica cuál de las siguientes afirmaciones son verdaderas o falsas (V/F) con respecto a la ejecución de la instrucción **lb \$t1, 0(\$t0)**:

| | |
|--|---|
| La instrucción carga \$t1 con el contenido total del registro <i>Status</i> . | F |
| La instrucción carga el byte menos significativo del registro <i>Status</i> en \$t1. | F |
| La instrucción no accede al registro <i>Status</i> . | V |
| La instrucción carga \$t1 con la dirección del registro <i>Status</i> . | F |

4. (1 punto) Escribe el código que permite programar una transferencia de lectura desde el dispositivo a la memoria. La operación de lectura empieza cuando los bits 0 y 3 del registro *Command* se escriben a uno, y los demás a cero. Se desea leer el bloque cuyo identificador es 0xABCD1234. El tamaño del bloque es de 512 bytes pero el bus de datos es de 32 bits, por tanto, en cada ciclo de ADM se transfiere una palabra de 32 bits. El registro contador se decreuenta de uno en uno con cada transferencia. La dirección de almacenamiento del bloque en la memoria está etiquetada con Mem_Block.

```

la $t0, 0xFF002340
li $t1, 128          # Número de transferencias (512/4)
sw $t1, 0x10($t0)    # Registro Counter
la $t1, Mem_Block
sw $t1, 0xC($t0)      # Registro Pointer
li $t1, 0xABCD1234
sw $t1, 0x14($t0)    # Registro Block_ID
li $t1, 9            # Orden lectura, bits 0 y 3 a 1
sw $t1, 0x4($t0)     # Registro Command

```

Ejercicio 3 (1 Punto)

Considera el disco duro magnético definido por los siguientes parámetros:

Velocidad rotacional: 6000 RPM
 Densidad lineal: 5000 pistas/pulgada
 Número de caras: 4
 Radio interno: 1"
 Radio externo: 3"
 Tamaño Sector: 512 bytes

Formato : ZCAV con dos zonas

| Zona | Límite | Sector/pista |
|------|--------|--------------|
| 0 | 2"-3" | 500 |
| 1 | 1"-2" | 300 |

1. (0,5 puntos) Calcule la velocidad de transferencia (en MB/s) cuando el disco accede a la zona 0. Justifique la respuesta.

$$\frac{500 \text{ sectores} \times 512 \text{ B/sector}}{10 \text{ ms}} = 25600000 \text{ B/s} = 25,6 \text{ MB/s}$$

10 ms = Tiempo/vuelta = 60 s / 6000 vueltas

2. (0.5 puntos) Calcule la capacidad total (en MB) del disco (1 M = 10⁶). Justifique la respuesta.

$$1 \text{ pulgada por cada zona} \times 5000 \text{ pistas/pulgada} = 5000 \text{ cilindros por zona}$$

$$5000 \text{ pistas/cara} \times (500 + 300) \frac{\text{sectores}}{\text{pista}} \times 512 \text{ B/sector} \times 4 \text{ caras} = 8192000000 \text{ B} = 8192 \text{ MB}$$

Ejercicio 4 (2 Puntos)

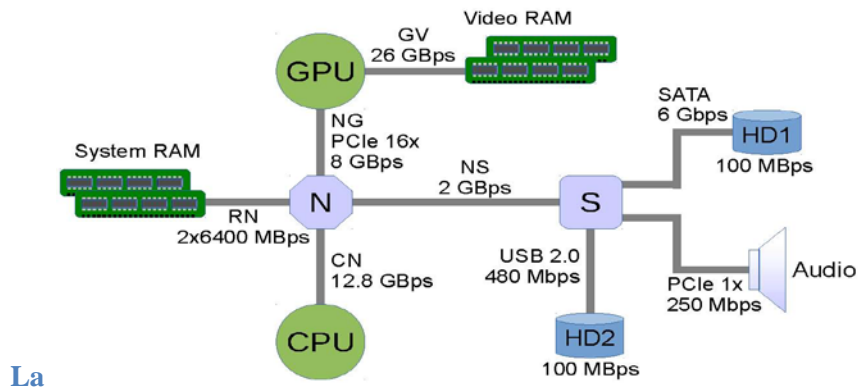
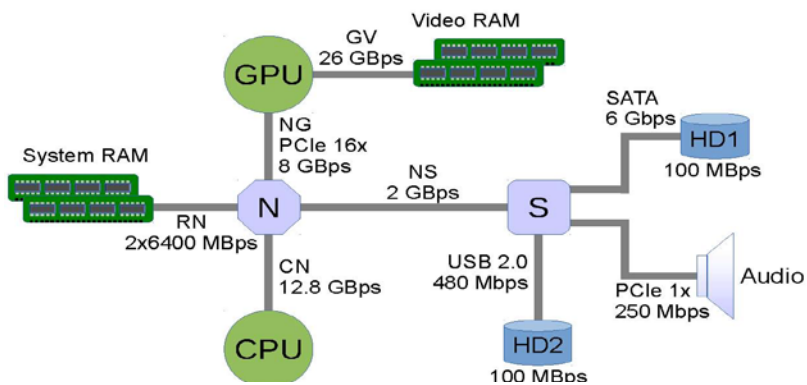


Figura 4 muestra la estructura de los buses de un determinado computador. Todos los anchos de banda indicados son efectivos, exceptuando los buses SATA y PCIe 1x, los cuales codifican un byte en 10 bits. El disco duro HD2 almacena un archivo que contiene una película de 2 minutos comprimida en sistema H265, a 30 Mbps. La película descomprimida está formada por escenas de $1920 \times 1200 \times 24$ bits a 30 escenas/segundo; y audio 5.1 a 24 bits por canal y muestreada a 96 KHz.



- (0,5 puntos) Calcule el tamaño de la película comprimida. Justifique los cálculos.
 $30 \text{ Mb/s} \times 120 \text{ s} = 3600 \text{ Mb} = 450 \text{ MB}$
- (0.5 puntos) Calcule el tiempo necesario para copiar el vídeo comprimido desde el disco HD2 al disco HD1. La copia consiste en leer desde el disco HD2 a la memoria, y desde la memoria al disco HD1. Ambas transferencias se realizarán por ADM. Asuma que ambas transferencias se hacen concurrentemente. Justifique los cálculos.

Bus RN = 12800 MB/s

Bus NS = 2000 MB/s

Bus USB = 60 MB/s (No codificado)

Bus SATA (codif. 8/10) = $6 / 10 \text{ GB/s} = 600 \text{ MB/s}$

$AB = \min\{\text{disco}_{HD2}, \text{USB}, \text{NS}, \text{RN}, \text{SATA}, \text{disco}_{HD1}\} = 60 \text{ MB/s}$

$T = 450 \text{ MB} / 60 \text{ MB/s} = 7,5 \text{ s}$

- (1 punto) Calcule el porcentaje de utilización de todos los buses involucrados en dicha transferencia. Justifique la respuesta.

Bus RN = $(60 \times 2) / 12800 = 0,94 \%$

Bus NS = $(60 \times 2) / 2000 = 6 \%$

Bus USB 2.0 = 100 %

Bus SATA = $60/600 = 10\%$

4. Ahora asuma que el vídeo está descomprimido en *System RAM*. Justifique que el ancho de banda del bus NG es suficiente para reproducir la película.

$$AB = 1920 \times 1200 \times 24 \times 30 \text{ bps} = 1658880000 \text{ bps} = 207,36 \text{ MBps}$$

Bus NG tiene 8000 MBps luego está utilizado en un $207,36/8000 = 2,592 \%$ por tanto es suficiente para visualizar el vídeo descomprimido.