

Estructura de Computadores

Parcial 2 Jerarquía de memoria y Entrada/Salida

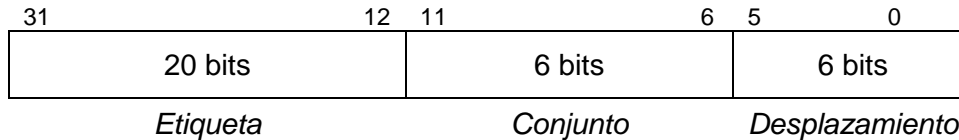
1 Junio - 2016

Nombre:

Grupo:

1) (3 puntos) Un sistema basado en el procesador MIPS R2000 dispone de cache L1 segregada para Instrucciones y Datos. Las configuraciones para cada una de ellas son las siguientes:

- Cache de **Instrucciones**: 8 KB, correspondencia directa, tamaño de bloque de 16B.
- Cache de **Datos**: 8 KB y un formato de direcciones de memoria principal como el que se muestra:



a) (0,5 puntos) Calcule los siguientes parámetros:

	Cache Instrucciones	Cache Datos
Tamaño de bloque (bytes)	16 B	64 B
Número de líneas	512	128
Número de vías	1	2
Número de conjuntos	512	64

b) El siguiente fragmento de código realiza la rotación (una posición) a derechas de un vector V de números enteros { $V[i+1] \leftarrow V[i] \quad \forall i \quad 0 \leq i \leq 4094, \quad V[0] \leftarrow V[4095]$ }

```

.data 0x2B000800
V:    .word 1, 2, 3, ..., 4096    # vector de 4096 enteros

.text 0x00400000
_start: li $t3, 4095              # carga contador
        lui $t0, 0x2B00           # carga puntero a V
        ori $t0, $t0, 0x0800
        lw $t1, 0($t0)            # lee V[0] en $t1
buc:    lw $t2, 4($t0)             # lee V[i+1] y lo almacena en reg. temporal $t2
        sw $t1, 4($t0)            # V[i+1] ← V[i]
        or $t1, $t2, $zero        # mueve $t2 a $t1
        addi $t0, $t0, 4          # incrementa puntero
        addi $t3, $t3, -1         # decrementa contador
        bnez $t3, buc            # mientras contador≠0, seguir en el bucle
        lui $t0, 0x2B00          # vuelve a carga puntero a V
        ori $t0, $t0, 0x0800
        sw $t1, 0($t0)           # V[0] ← V[4095]
        .end
    
```

b.1) (0,75 puntos) Para la cache **de Instrucciones**, calcule los siguientes datos:

Número de bloques que ocupa el código	Dirección del primer bloque	Dirección del último bloque	Total de FALLOS de código
4	0x0040000	0x0040003	4
Total de ACCESOS a código (Indicar el cálculo)	4 + 6x4095 + 3 = 24577		
Tasa de aciertos	24573/24577 = 0,9998 (99,98%)		

b.2) (1 punto) Para la **cache de Datos**, sabiendo que la política de escritura es de ubicación (*write allocate*), que la de actualización es de copia posterior (*write-back*) y que el algoritmo de reemplazo es LRU, calcule los siguientes datos:

Núm. de bloques que ocupa el vector	Dirección del primer bloque	Conjunto al que se mapea el primer bloque	Total de ACCESOS a datos
256	0xAC0020	0x20	8192
Total de FALLOS de datos (Indicar el calculo)	256 de inicio + 1 de capacidad= 257		
Tasa de aciertos	$1 - (257/8192) = 0,9686$ (96,86%)		
Núm de reemplazos de bloque	129		
Núm. de escrituras a MP (palabras)	2064		

b.3) (0,5 puntos) Calcule el tamaño de la memoria de control para las cache de Instrucciones y Datos:

	Cache Instrucciones	Cache Datos
Número de entradas o palabras de la memoria de control	512	128
Número de bits de cada entrada y campos que la integran	V(1)+etiqueta(19)	V(1)+etiqueta(20)+M(1)+LRU(1)
Tamaño total de la memoria de control (en bits)	512 x 20 bits = 10240 bits	128 x 23 = 2944 bits

b.4) (0,25 puntos) Comente brevemente las posibles ventajas o inconvenientes que tendría en este caso el empleo de correspondencia totalmente asociativa en la cache de datos. Razona la respuesta

No tendría ninguna ventaja, por cuanto todos los fallos son de inicio, a excepción del último, que es de capacidad. Por el contrario, tendría el inconveniente de un mayor retardo y consumo.

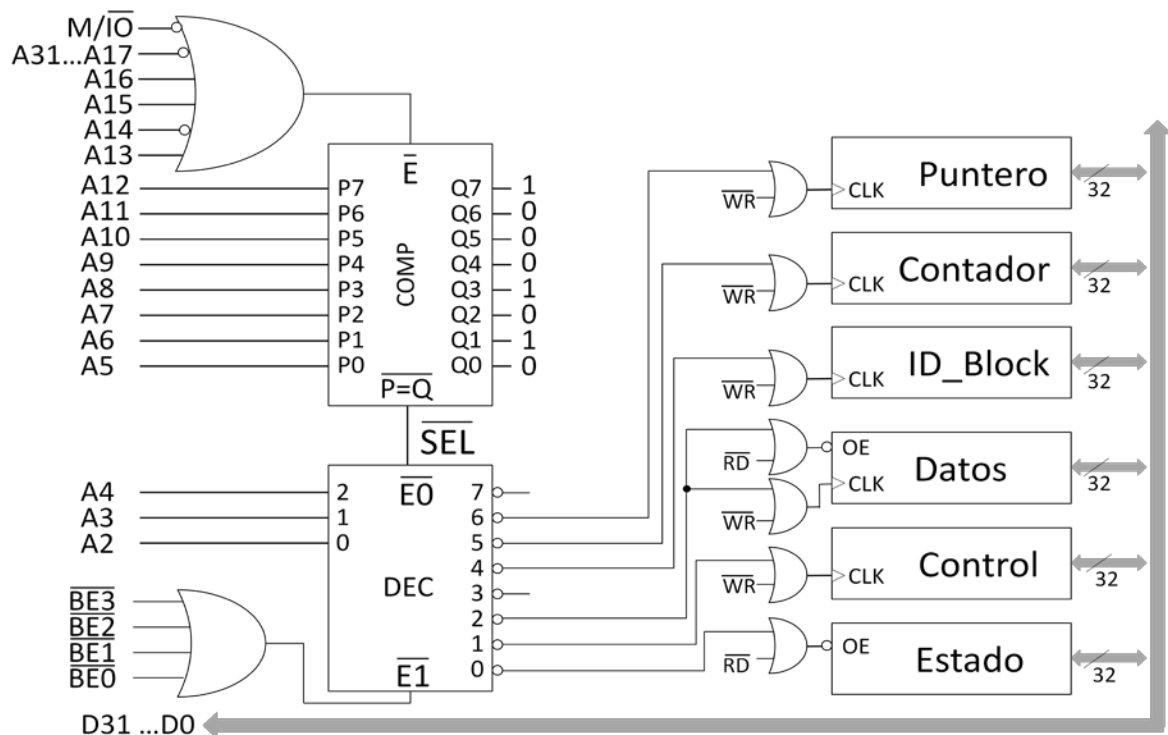
2) (2,5 puntos) En la figura se muestra el esquema de un interfaz de E/S para ser conectado a una CPU MIPS R2000 que ha sido modificada para incluir dos espacios de direccionamiento: el espacio de memoria y el espacio de entrada/salida. Por ello, su conjunto de instrucciones se amplía con las correspondientes instrucciones de lectura/escritura en puertos de E/S: **InputW/InputH/InputB** y **OutputW/OutputH/OutputB**, con una sintaxis similar a las **loads/stores** de memoria. Dicho interfaz ha sido concebido para trabajar con dispositivos de bloques (discos magnéticos) y soporta transferencias PIO y ADM. Los registros Estado y Control poseen los siguientes bits significativos:

Registro **CONTROL**:

- **MOD** (bits 7 y 6). Permite seleccionar entre el modo PIO (MOD=00) y el modo ADM (MOD=11)
- **A** (bit 3), a 1 ordena al interfaz el inicio de una operación de lectura/escritura sobre el periférico (disco magnético)
- **R/W** (bit 1), indica al interfaz si se trata operación de lectura (R/W= 0) o de escritura (R/W=1) sobre el dispositivo de bloques
- **CL** (bit 0), a 1 hace R=0

Registro **ESTADO**:

- **R** (bit 2) se activa a 1 cuando el bloque está listo para empezar a ser transferido a/desde memoria (modo PIO) o bien cuando la transferencia a/desde memoria a concluido (modo ADM)



- a) (0,25 puntos) Calcule la dirección base (DB) del interfaz

0xFFFFE5140

- b) (0,25 puntos) Indique el tipo de espacio de direccionamiento en el que se halla mapeado el interfaz. Justifica la respuesta

Espacio de memoria, puesto que el interfaz se selecciona con $M/\overline{IO}^*=1$

- c) (0,5 puntos) Calcule la dirección de puerto (DB+X) de cada uno de los registros del interfaz

Registro	Dirección de puerto
ESTADO	DB
CONTROL	DB+4
DATOS	DB+8

Registro	Dirección de puerto
ID_BLOCK	DB+16
CONTADOR	DB+20
PUNTERO	DB+24

- d) (0,25 puntos) Indique el tamaño en bytes que ocupa este interfaz en el espacio en el que se halla mapeado y que no está disponible, pues, para otros dispositivos. Justifica la respuesta

32 bytes (2^5 , donde 5 corresponde al número de líneas de dirección que no intervienen en el seleccionado del interfaz)

- e) (1,25 puntos) El driver del dispositivo de bloques controlado a través del interfaz del esquema anterior dispone de la siguiente función:

Función	Índice (en \$v0)	Argumentos
Read_Disk	400	\$a0: Puntero a buffer de memoria \$a1: Número de ciclos de transferencia \$a3: Identificador del bloque

Se desea implementar la función Read_Disk de modo que la transferencia se realice en modo PIO y que la sincronización con el dispositivo se realice mediante CONSULTA DE ESTADO. Suponemos que la sincronización se realiza al nivel de bloque y no palabra a palabra. También se asume que los registros \$a0, \$a1 y \$a3 han sido, en el momento de la llamada a la función Read_Disk desde la aplicación, debidamente inicializados con la dirección inicial del buffer en memoria, el número de ciclos de transferencia que requiere el bloque y el identificador del bloque que se desea leer de disco, respectivamente

```

Read_Disk:  la $t0, 0xFFFE5140
            sw $a3, 16($t0)           #inicializa reg. ID_BLOCK
            li $t1, 0x08
            sw $t1, 4($t0)           #configura modo PIO y ordena inicio operación lectura

            buc1: lw $t1, 0($t0)
                andi $t1, $t1, 0x04
                bnez $t1, buc1        #bucle de consulta
                li $t2, 1
                sw $t2, 4($t0)        #cancela R (R=0), bit CL a 1

            buc2: lw $t1, 8($t0)       #lectura palabra desde reg. DATOS
                sw $t1, 0($a0)         #escritura en memoria mediante el reg. puntero $a0
                addi $a0, $a0, 4       #actualiza reg. puntero
                addi $a1, $a1, -1      #decrementa contador de num. Transferencias $a1
                bnez $a1, buc2         #mientras contador<>0, continuar transferencia

            j retexc

```

3) (3 puntos) El sistema de control de un coche está gobernado por un MIPS R2000. Entre las interfaces de entrada/salida que controla se encuentran dos:

Interfaz primera: Un sensor distancia que cuando está activo (bit A) mide la distancia media de la parte trasera del vehículo al obstáculo situado enfrente. Cuando hay un cambio de al menos 5 mm en la distancia medida entonces se actualiza el registro de distancia y los bits correspondientes del registro de estado. Si además el bit E es uno, se activará la interrupción INT₁*. La descripción de los bits de la interfaz del sensor de distancia es:

Registro **ESTADO** (Sólo lectura, dirección base 0xFFFE0028):

- Bits 1..0 - **D**: Se actualizan por comparación entre la distancia y la referencia (ambas en los registros con dicho nombre):
 - D₁ = 0, D₀=0 si la distancia > referencia
 - D₁ = 0, D₀=1 si la referencia ≥ distancia > 1/2 referencia
 - D₁ = 1, D₀=0 si la 1/2 referencia ≥ distancia
- Bit 7 - **R**: La interfaz lo pone a 1 cuando hay un cambio en la distancia mayor de 5mm.

Registro **CONTROL** (Sólo escritura, dirección base 0xFFFE0028):

- Bit 0 - **A**: 1 para activar la interfaz y que mida distancias, si se deja a cero deja de medir.
- Bit 6 - **E**: Se pone a 1 para habilitar la interrupción en la interfaz y a 0 para inhibirla. Cuando E es 1 y el bit R también, se activa la interrupción 1 de la entrada del procesador (INT₁*).
- Bit 7 - **C**: Bit de cancelación. Si se escribe un 1 entonces la interfaz pone el bit R a 0.

Registro **DISTANCIA** (Sólo lectura, dirección base +4). Registro de 8 bits.

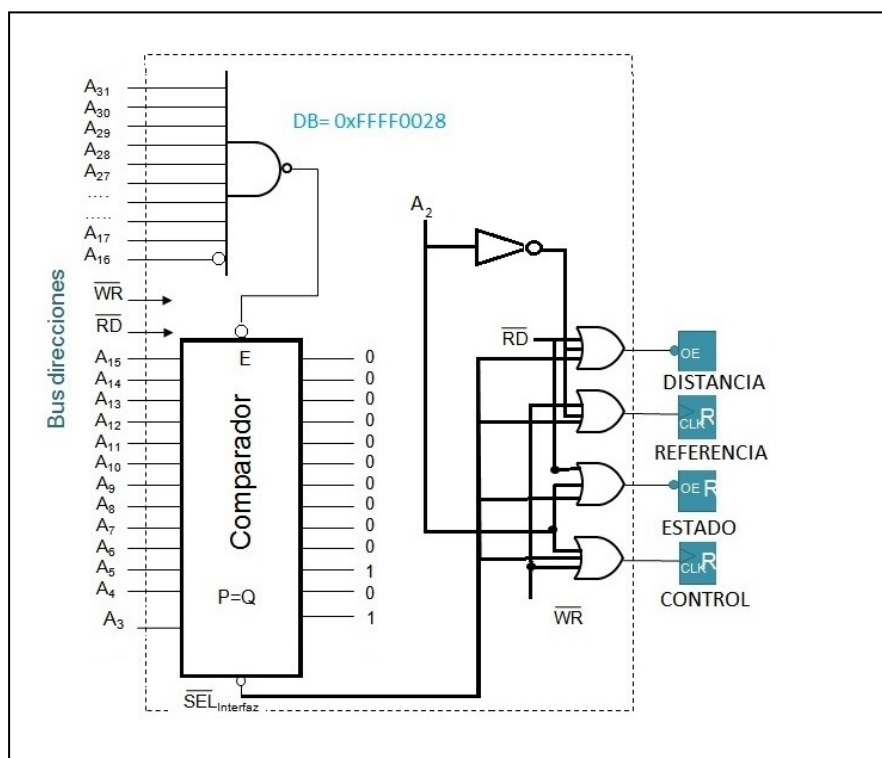
Registro **REFERENCIA** (Sólo escritura, dirección base +4). Registro de 8 bits.

Interfaz segunda: un actuador sonoro que hace sonar una alarma a una frecuencia determinada. Este actuador solo tiene un registro que funciona por sincronización o E/S directa:

Registro **CONTROL** (Sólo escritura, dirección base 0xFF000000):

- Bit 0 - **S**: 1 para hacer sonar la alarma a la frecuencia indicada en los bits frecuencia. Si se pone este bit a 0 deja de sonar la alarma, y si no estaba sonando no tiene efecto.
- Bits 7..3- **Frecuencia**: Frecuencia de la alarma. Si se ponen a cero suena la alarma a la frecuencia base, valores mayores de cero la aceleran hasta el valor máximo.

- a) (0,5 puntos) Realice un dibujo del conexionado interno de selección de los cuatro registros del sensor de distancias, tenga en cuenta todas las líneas que intervienen en la selección (patillas de direcciones A_i , líneas BE_i^* y señales de lectura RD^* y escritura WR^*). Puede utilizar un comparador de 13 entradas, activo por nivel bajo (como se ha utilizado en los ejercicios de clase) y puertas not, nand y or de las entradas que necesite.



- b) (2 puntos) Programe la rutina del sistema asociada a la línea INT1. Suponga que en el sistema operativo se encuentra definida la siguiente variable que puede utilizar:

.kdata

frecuencia: .byte 0x20 # frecuencia de alarma sonora

Asuma el siguiente comportamiento:

Caso 1: ($D_1=0$ y $D_0=1$) "hacer sonar la alarma a frecuencia base"

Caso 2: ($D_1=1$ y $D_0=0$) "hacer sonar la alarma a frecuencia almacenada en la variable *frecuencia*"

Caso 3: ($D_1=0$ y $D_0=0$) "parar la alarma"

AVISO: El contenido de la variable frecuencia se encuentra alineada en los bits 7..3 (campo frecuencia del registro)

Se pueden emplear los registros \$t0 a \$t4

```

INT1:    la $t0, 0xFFFF0028
         lb $t1, 0($t0)           # $t1 ← Estado
         andi $t1, $t1, 3         # Filtro bits Di
         beqz $t1, Caso3
         li $t2, 2
         beq $t1, $t2, Caso2
Caso1:   li $t1, 1                # frecuencia base es 0, S=1
         la $t0, 0xFF000000
         sb $t1, 0($t0)          # $t1 → Control Sonoro
         b cancelar
Caso2:   lw $t1, frecuencia       # frecuencia en variable SOP
         ori $t1, $t1, 1         # activo bit S
         la $t0, 0xFF000000
         sb $t1, 0($t0)          # $t1 → Control Sonoro
         b cancelar
Caso3:   la $t0, 0xFF000000       # Parar la alarma, S=0
         sb $zero, 0($t0)        # $zero → Control Sonoro
cancelar: la $t0, 0xFFFF0028
         li $t1, 0xC1            # Cancelar C=E=A=1
         sb $t1, 0($t0)
         b retexc

```

c) (0,5 puntos) Programe las siguientes llamadas al sistema:

Función	Índice	Argumentos	Resultado
<i>Activar_Sensor</i>	\$v0 = 30	\$a0= distancia de referencia	Guarda en el registro de referencia del sensor distancia el valor en \$a0 Activa la captura de distancias (bit A) y las interrupciones (bit E)
<i>Desactivar_Sensor</i>	\$v0 = 31	-----	Desactiva la captura de distancias (bit A) y las interrupciones (bit E)

```

Activar_Sensor:  la $t0, 0xFFFF0028
                 sb $a0, 4($t0)    # $a0 → Referencia
                 li $t1, 0x41      # bits E y A a "1"
                 sb $t1, 0($t0)    # Escribo en control
                 b retexc

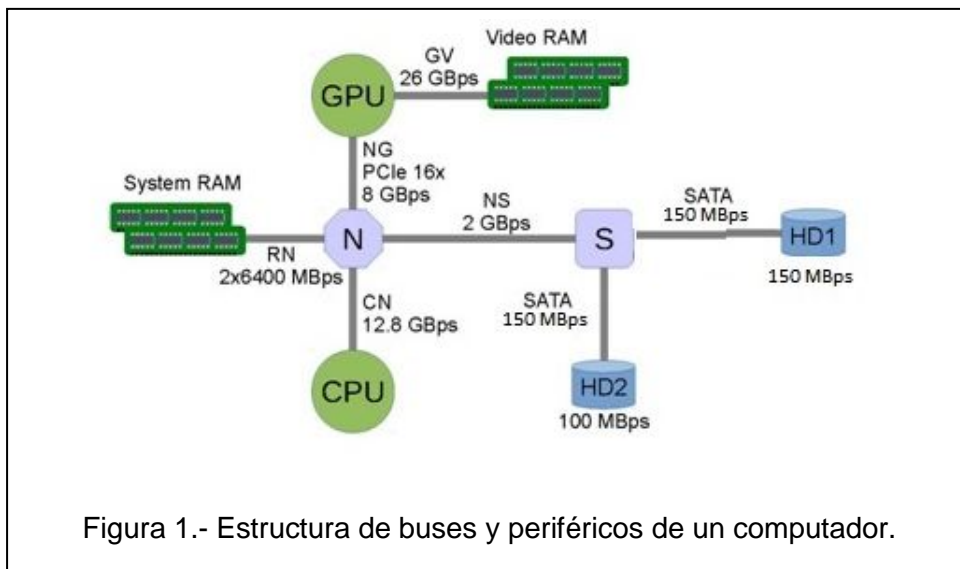
```

```

Desactivar_Sensor: la $t0, 0xFFFF0028
                  sb $zero, 0($t0) # bits E y A a "0"
                  b retexc

```

- 4) (1 punto) La Figura 1 muestra la estructura de buses y periféricos de un computador.



Se pide:

- a) Calcule el tiempo empleado en realizar la copia de un archivo de 1GB (10^9 B) desde HD2 a HD1 si mientras tanto se está visualizando una película muda a baja resolución formada por escenas de $800 \times 600 \times 24$ bits a 60 escenas/segundo almacenada en el disco HD1, y cuyo tráfico es prioritario.
NOTA: Las transferencias se hacen por ADM y el tiempo de procesador es despreciable.

La película necesita un ancho de banda de $800 \times 600 \times 24 \times 60 / 8 \text{ Bps} = 86,4 \text{ MBps}$
 Puesto que el flujo de la película tiene prioridad, la transferencia del archivo desde HD1 hasta HD2 se deberá hacer al ancho de banda que quede en HD1, que será $150 - 86,4 = 63,6 \text{ MBps}$.

Luego el tiempo será:

$$\frac{10^9 \text{ B}}{63,6 \times 10^6 \text{ Bps}} = 15,72 \text{ s}$$

- b) Calcule el porcentaje de ocupación del bus NS

El bus NS soporta $86,4 \text{ MBps}$ de la película + $63,6 \text{ MBps} \times 2$ del archivo = $213,6 \text{ MBps}$

$$\frac{213,6 \times 10^6 \text{ Bps}}{2 \times 10^9 \text{ Bps}} = 0,1068 \quad (10,68\%)$$

5) **(0,5 puntos)** Considerando los siguientes parámetros de un disco duro, calcule el tiempo en transferir un archivo de 640 KB ($K=10^3$), si se sabe que se encuentra almacenado de modo óptimo en la zona 0. El tiempo medio de posicionamiento es 3 ms.

Velocidad rotacional: 6000 RPM
 Densidad lineal: 4000 pistas/pulgada
 Número de caras: 6
 Radio interno: 1"
 Radio externo: 5"
 Tamaño Sector: 512 bytes

Formato : ZCAV con dos zonas

Zona	Límite	Sector/pista
0	3"-5"	520
1	1"-3"	290

640000/512 = 1250 sectores. Como un cilindro de la zona cero tiene 3120 sectores, cabe en uno de ellos si está almacenado de forma óptima.

Tiempo = Tiempo de acceso medio + (bytes archivo / V_0)

Tiempo acceso medio = Tiempo de posicionamiento medio + 1/2 Tiempo/vuelta

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

velocidad de transferencia zona 0 (MB/s) = V_0 = 26,6 MB/s

$$\frac{520 \text{ sectores} \times 512 \text{ B/sector}}{10 \text{ ms}} = 26624000 \text{ B/s} = 26,6 \text{ MB/s}$$

Tiempo = 8 ms + (640 KB / 26,6 MBps) = 8 + 24,06 = 32,06 ms