

# ESTRUCTURA DE COMPUTADORES

## Examen Recuperación Segundo Parcial

25/Junio/2012

Apellidos y Nombre	DNI	Grupo

1.- (1 punto) Un procesador monociclo que trabaja a una frecuencia de reloj de 20 MHz se ha rediseñado para ofrecer una versión segmentada de cinco etapas como la vista en clase. Las etapas LI, DI, EX, M, ER tienen un retardo de cada una de ellas, respectivamente, de 10, 8, 14, 10 y 6 ns. Los registros de segmentación añaden un retardo adicional de 2 ns. Indíquese para la versión segmentada:

a) Frecuencia máxima de reloj a la que puede trabajar. ¿cómo se obtiene? **0.4**

--

b) Máxima productividad que conseguiría en condiciones ideales. Justifíquese. **0.3**

--

c) ¿Cuál es la aceleración máxima que se podría conseguir respecto de la versión original monociclo?

**2.- (2 puntos)** En el procesador segmentado anterior se van a ejecutar los siguientes fragmentos de código escritos en el ensamblador del MIPS R2000. Rellene los siguientes diagramas instrucciones/tiempo indicando para cada caso los riesgos encontrados y aplicando las soluciones propuestas en cada enunciado. En todos los casos se desea evitar al máximo el uso de soluciones conservativas como inserción de NOP o ciclos de parada ya que el objetivo principal es incrementar la productividad del procesador

**a. Primera secuencia. 0.5**

(1) sw \$1, 500(\$2)

(2) 1w \$3, 100(\$1)

Instrucción/ciclo	1	2	3	4	5	6	7	8	9	10	11
-------------------	---	---	---	---	---	---	---	---	---	----	----

SW \$1, 500(\$2)

1w \$3, 100(\$1)

[illegible]

**Utilice cortocircuitos**, si es posible, para solucionar o minimizar el impacto de los riesgos encontrados. En ese caso especifique claramente el tipo de cortocircuito empleado.

Riesgos detectados:	Tipo de cortocircuito empleado:
Número de ciclos requeridos para ejecutar el fragmento:	

**b. Segunda secuencia. 0.75**

- (1) lb \$3, B(\$5)
- (2) add \$4, \$5, \$3
- (3) addi \$5, \$5, -1
- (4) sub \$3, \$4, \$3
- (5) sb \$3, c(\$5)

Instrucción/ciclo	1	2	3	4	5	6	7	8	9	10	11
-------------------	---	---	---	---	---	---	---	---	---	----	----

lb \$3, B(\$5)											
add \$4, \$5, \$3											
addi \$5, \$5, -1											
sub \$3, \$4, \$3											
sb \$3, c(\$5)											

**Utilice cortocircuitos**, si es posible, para solucionar o minimizar el impacto de los riesgos encontrados. En ese caso especifique claramente el tipo de cortocircuito empleado.

Riesgos detectados:	Tipo de cortocircuito empleado:
Número de ciclos requeridos para ejecutar el fragmento:	

**c. Tercera secuencia.** En este caso se trata de solucionar un riesgo de control. Sabiendo que la latencia de salto en este procesador es 1 (delay slot =1), utilice la reordenación de código para solucionar o minimizar el riesgo **0.75**

- (1) lb \$3, B(\$5)
- bucle: (2) add \$4, \$5, \$4
- (3) addi \$5, \$5, -1
- (4) sub \$4, \$4, \$3
- (5) bgtz \$4, bucle
- (6) sb \$4, c(\$5)

[illegible]

--

--

Todas las líneas de dirección comprendidas entre la  $A_{31}$  y la  $A_{15}$  se conectan directamente a la puerta NAND (entradas no invertidas), excepto la línea  $A_{15}$ . Asumiendo que **los seis registros del adaptador son de 32 bits de tamaño**, se pide:

d) ¿Cuál es la dirección base (DB) de este dispositivo? **0.5**

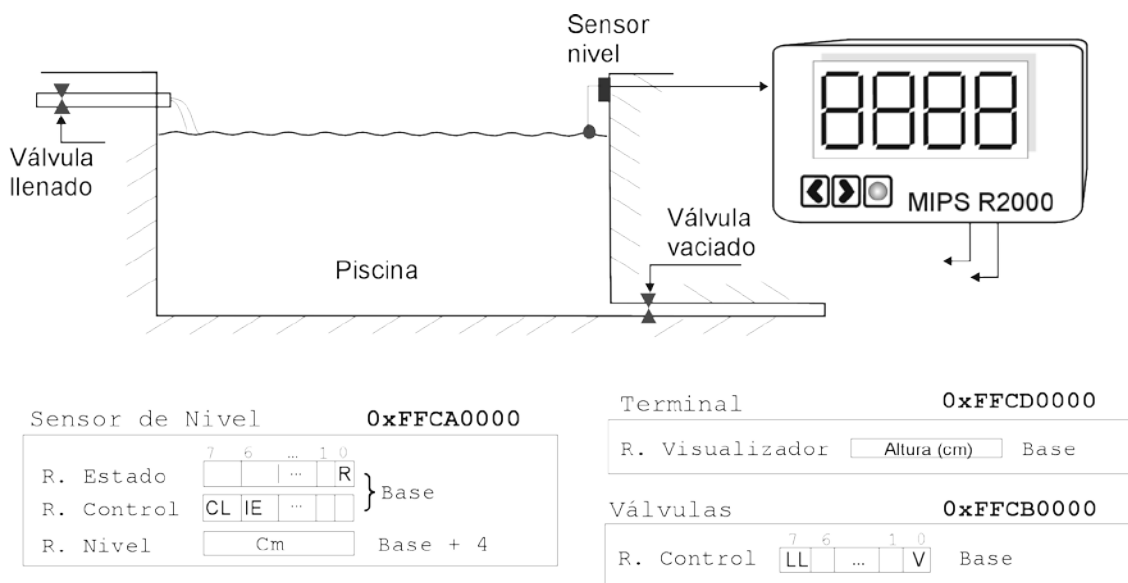
e) Examinando las conexiones de las líneas  $A_4$ ,  $A_3$  y  $A_2$ , indíquense las direcciones en que se ubican cada uno de los 6 registros del adaptador, expresándolas en términos de  $DB + \langle \text{desplazamiento} \rangle$ . **0.4**

f) Supóngase que el registro de Estado contiene un bit READY localizado en el bit 0. Este bit se establece a 1 por el hardware cuando hay disponible un nuevo dato en el registro de Datos. Escribase 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 **0.8**

g) Escribase el código para programar una transferencia DMA desde el dispositivo a memoria (transferencia de lectura). La transferencia se inicia cuando los bits 0 y 1 del registro de Command se establecen a 1. 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. Asimismo, se desea almacenar el bloque en la dirección de memoria etiquetada como `Mem_Block`. *Nota: El contador se decrementa en 1 con cada ciclo de transferencia* **0.8**

**4.- (3 puntos)** El nivel de agua de una piscina varía debido a las lluvias o a la evaporación por el sol. El control de nivel de dicha piscina se realiza mediante un equipo que incorpora un procesador MIPS R2000 y dispone de los siguientes periféricos: un medidor de nivel de agua, un control de electroválvulas de llenado y vaciado y un visualizador. Un esquema del sistema se muestra en la figura adjunta.

El sensor de nivel registra cambios permanentes (no transitorios debidos al oleaje) de  $\pm 1$  cm en la altura del nivel de agua. El equipo compara este nivel con una consigna o nivel de referencia almacenado en la memoria del sistema y cuando difieren abre las válvulas de llenado o de vaciado según el caso. Si el nivel actual es igual a la consigna cierra las válvulas. El nivel actual del agua (en cm) se muestra constantemente en el visualizador.



Las interfaces del sistema se describen a continuación. El hardware ignora el valor de los bits no descritos.

- **Sensor de Nivel:** mide el nivel del agua en centímetros. Un cambio en  $\pm 1$  cm activa la interrupción INT1. Dirección base DB=0xFFCA0000.
  - **Registro estado** (lectura 8 bits DB): bit 0 (R ready) se pone a 1 cuando cambia el nivel. SI IE = 1 también activa la INT1.
  - **Registro de Control** (escritura 8 bits DB): bit 7 (CL:clear) = 1 pone a cero el bit R. Bit 6 (IE:interrupt enable)=1 habilita la interrupción.
  - **Registro de nivel** (lectura 8 bits DB+4): Altura del nivel de agua en centímetros.
- **Actuador de válvulas:** abre y cierra las válvulas. Dirección base DB=0xFFCB0000.
  - **Registro de Control** (escritura 8 bits DB): bit 7(LL) = 1 abre la válvula de llenado. Bit 1 (V) = 1 abre la válvula de vaciado. Un 0 cierra las válvulas.
- **Visualizador:** Visualiza la altura actual del agua en cm. Dirección base DB=0xFFCD0000.
  - **Registro del visualizador** (escritura 8 bits DB): Valor que se visualiza (altura del agua en centímetros).

Se pretende programar el equipo para la gestión del sistema. Para ello se disponen de las siguientes funciones del sistema:

Función	Índice (en \$v0)	Parámetros	Resultados
Inicializar	\$v0 = 30	\$a0 = valor inicial variables	-----
Leer_nivel	\$v0 = 31	-----	\$v0 = nivel del agua en cm
Leer_referencia	\$v0 = 32	-----	\$v0 = nivel de referencia en cm

Activar_valvulas	\$v0= 33	\$a0: 0 cerrar todo 0x01 abrir valv. vaciado 0x80 abrir valv. llenado	-----
------------------	----------	--	-------

Así mismo se tiene las siguientes variables del sistema:

```

        nivel_agua:      .kdata
        nivel_agua:      .word 0 # Nivel del agua en cm
        nivel_referencia: .word 0 # Consigna o nivel de referencia en cm

```

Se pide:

- a) Escriba el código de la función del sistema 'Inicializar' que habilita el sistema de interrupciones tanto en el procesador como en el periférico.

(Se supone que el manejador de excepciones salta a la etiqueta 'Inicializar:' cuando se produce la excepción. Se pueden usar los registros \$t0, \$t1 y \$t2 en el manejador). **0.8**

- b) Observe el siguiente programa que se ejecuta en el controlador de nivel:

```

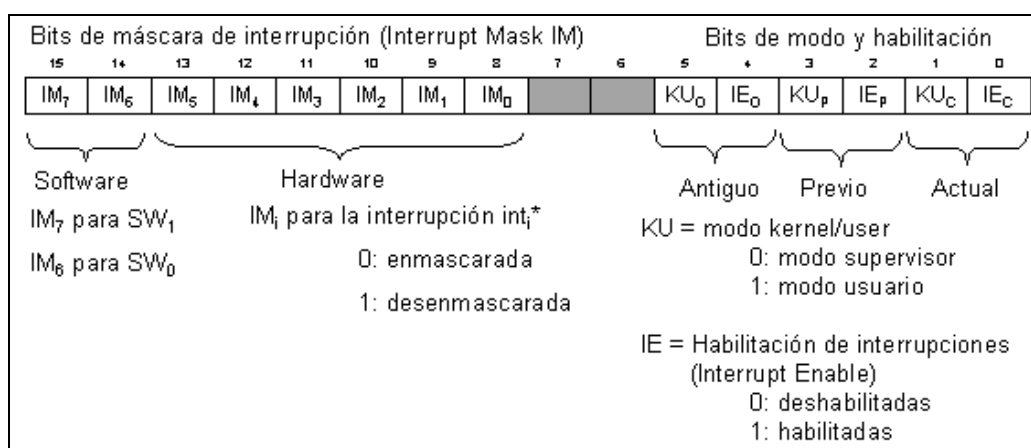
        .text
__start: li $a0, 140
        li $v0, 30
        syscall
Bucle:  li $v0, 31
        syscall
        bne $a0, $v0, _sigue
        li $a0, 0x00
        j _fin
_sigue: bgt $t0, $t1, _L1
_L2:    li $a0, 0x80
        j fin
_L1:    li $a0, 0x01
_fin:   li $v0, 33
        syscall
        j Bucle

```

Explique brevemente qué hace este programa **0.7**

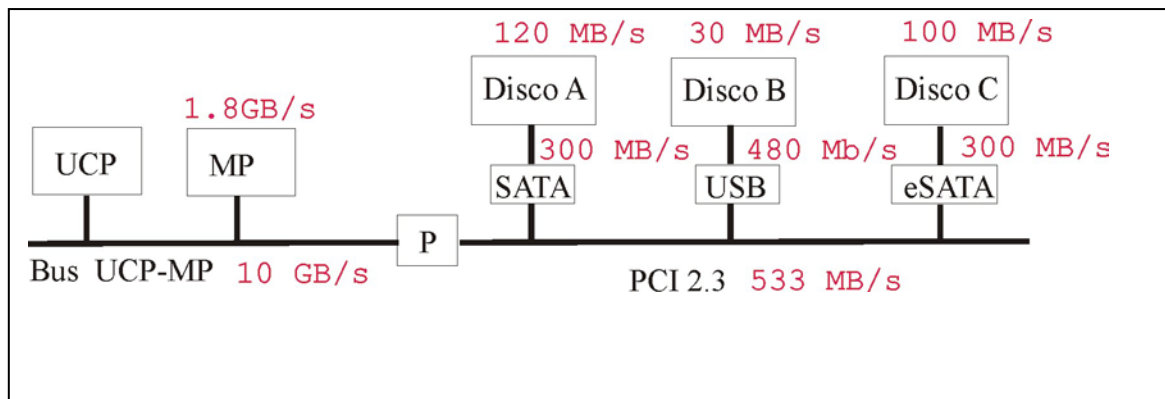
- |  |
|--|
|  |
|--|

- [illegible]



**5.- (1,5 puntos)** Considere el sistema de la figura, formado por un subsistema procesador (UCP) y memoria (MP), y un bus de expansión en el que hay instalados tres dispositivos de almacenamiento (discos A, B y C). Las velocidades de cada elemento son las siguientes:

- MP: 1.8 GB/s
- Bus SATA: 300 MB/s
- Bus USB: 480 Mb/s
- Bus eSATA: 300 MB/s
- Bus UCP-MP: 10 GB/s
- Bus PCI 2.3: 533 MB/s
- Disco A: 120 MB/s
- Disco B: 30 MB/s
- Disco C: 100 MB/s



En un instante dado, la demanda de memoria por parte del procesador es de 1200 MB/s, y se están leyendo dos archivos, uno desde el disco A y el otro desde el disco C. Considerando solo las limitaciones impuestas por el ancho de banda de los buses y los propios dispositivos de almacenamiento, responda a las siguientes cuestiones:

- a) Máximo ancho de banda a que se puede leer cada uno de los dispositivos disco A y disco C. **0.5**

- b) Teniendo en cuenta todas las transferencias que se están ejecutando, y para la velocidad máxima calculada en el apartado anterior, calcule el porcentaje de utilización del bus UCP-MP. **0.5**

- c) Si el procesador aumentara la demanda de memoria en 500 MB/s. ¿Cómo afectaría a la velocidad de lectura de los discos? (Asuma que el procesador tiene preferencia sobre los discos, y que se repartirá el ancho de banda restante equitativamente) **0.5**