

ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES

1º de I.T.I. Gestión, I.T.I. Sistemas e I. Superior

Examen de teoría – Parte II: cuestiones teórico-prácticas – 30 de junio de 2010

Apellidos:	Nombre:
D.N.I.:	
Grupo: Sistemas I <input type="checkbox"/> Sistemas II <input type="checkbox"/> Gestión I <input type="checkbox"/> Gestión II <input type="checkbox"/> Superior <input type="checkbox"/>	

PARTE II: CUESTIONES TEÓRICO-PRÁCTICAS

1. (2 puntos) Implementar, utilizando biestables tipo D disparados por flanco descendente, un circuito capaz de detectar un retraso de un ciclo entre dos señales de un bit. El circuito tendrá dos entradas, A y B, y una salida, S, que se pondrá a 1 siempre que la entrada B se corresponda con la A del ciclo anterior. En el momento en que esta condición no se cumpla la salida quedará fijada a 0 hasta que una señal de Reset, activa en baja, vuelva a iniciar el proceso de comprobación. En el primer ciclo de reloj, donde la salida todavía no tiene sentido, supondremos que se cumple la condición. [Pista: Observad que solamente necesito saber si hasta el momento se está cumpliendo la condición y el valor anterior de A para poder implementar el circuito.]
2. Dado el siguiente procedimiento escrito en ensamblador MIPS:

```
proc:    addi    $sp, $sp, -4
         sw    $s1, 0($sp)
         li    $v0, 0x80000000
L1:    beq    $a1, $0, L2
         lh    $s1, 0($a0)
         ble    $s1, $v0, L3
         move    $v0, $s1
L3:    addi    $a0, $a0, 2
         addi    $a1, $a1, -1
         j    L1
L2:    lw    $s1, 0($sp)
         addi    $sp, $sp, 4
         jr    $ra
```

- a) (0.75 puntos) Escriba el código máquina correspondiente a las instrucciones (o pseudoinstrucciones) marcadas en negrita, indicando también la dirección de cada una. Suponga que la etiqueta proc está en la dirección 0x04000040 y que la pseudoinstrucción de la tercera línea se traduce mediante una sola instrucción (lui). El contenido de los diversos campos se pueden escribir en decimal o hexadecimal si así se desea.
- b) (0.25 puntos) El procedimiento recibe dos parámetros: la dirección de un array de enteros de 16 bits en \$a0 y el número de elementos de dicho array en \$a1. Explique qué hace el procedimiento. (No se pide lo que hace cada instrucción individualmente, sino que se explique la utilidad de este procedimiento).
- c) (0.75 puntos) Considerando el volcado de memoria que se muestra a continuación, que el registro \$a0 contiene el valor 0x10018000 y el registro \$a1 el valor 5, calcule el tiempo de ejecución del procedimiento en un procesador monociclo a 100 Mhz. (Nota: estamos ante una arquitectura *Little Endian*).

	+0	+1	+2	+3
10018000	34	00	a1	00
10018004	ff	ff	73	00
10018008	0a	00	00	00

Nota: todos los valores están en hexadecimal. Cada casilla indica el byte almacenado en memoria en la dirección resultante de sumar el título de la fila y el título de la columna correspondientes.

- d) (0.25 puntos) Se quiere escribir este procedimiento sin incluir ni las dos primeras instrucciones, ni la penúltima ni la antepenúltima. Indique qué cambio o cambios adicionales serían necesarios para que el procedimiento siguiera siendo correcto.
3. (2 puntos) Suponga un sistema de memoria formado por una caché con un tamaño de 64 KB y bloques de 8 palabras (1 palabra = 32 bits). El bus que conecta la caché y la memoria principal (MP) utiliza un reloj de 500 MHz, está multiplexado y tiene un ancho de 32 bits. La transmisión de una dirección requiere un ciclo de bus mientras que el acceso a la MP tiene una latencia de 4 ciclos. El bus dispone de un modo ráfaga capaz de transmitir hasta un máximo de 4 palabras a la velocidad de una palabra cada dos ciclos.
- a) (0,25 puntos) Indique los ciclos que requiere una operación de lectura y una operación de escritura de un bloque (8 palabras) en MP teniendo en cuenta que se realizan dos accesos consecutivos al bus para transferir un bloque completo.
- b) (0,25 puntos) Si la caché usa una política de post-escritura y hay un 40% de bloques modificados, ¿cuántos ciclos de bus requiere, en promedio, una instrucción de carga que falla en caché? ¿y una instrucción de almacenamiento que falla en caché?
- c) (0,25 puntos) Sabemos que el tráfico que circula por el bus proviene de un 70% de instrucciones de carga y un 30% de almacenamientos, ¿cuál será el máximo ancho de banda efectivo?
- d) (0,5 puntos) Repita el apartado anterior suponiendo ahora que la caché usa escritura directa.
- e) (0,75 puntos) Supongamos que la MP está paginada en páginas de 8 KB y que definimos una operación flush p que saca de la MP al proceso p pero que antes transfiere aquellas páginas físicas con el bit de modificado activo al disco duro. Si el proceso p ocupa un total de 20 páginas físicas estando el 75% de ellas modificadas, ¿cuántos ciclos tardaremos en realizar un flush p si usamos un mecanismo por interrupciones que transfiere los datos en bloques de 16 palabras con una sobrecarga de 1000 ciclos por interrupción? ¿Y por DMA si cada transferencia es de 1 página completa y requiere 500 ciclos para inicializar el DMA y 1500 ciclos para la interrupción de terminación?

Solución al ejercicio 1

Tal y como se indica en la pista del ejercicio, el estado del sistema a implementar se reduce a conservar el último valor visto en la entrada A, así como si hasta el momento se cumple o no la condición. Por tanto podemos establecer los siguientes estados:

Estado 0 (q_0): Estado inicial en donde se supone que la entrada B es la entrada A retrasada un ciclo y no importa el valor de A.

Estado 1 (q_1): La entrada B es la entrada A retrasada un ciclo y el valor anterior de A fue 0.

Estado 2 (q_2): La entrada B es la entrada A retrasada un ciclo y el valor anterior de A fue 1.

Estado 3 (q_3): La entrada B no se corresponde con la entrada A retrasada un ciclo.

La tabla de transición de estados y de la salida será la que se indica a continuación (nos olvidaremos por ahora de la señal de reset):

Estado actual	AB=00	AB=01	AB=10	AB=11	SALIDA
q_0	q_1	q_1	q_2	q_2	1
q_1	q_1	q_3	q_2	q_3	1
q_2	q_3	q_1	q_3	q_2	1
q_3	q_3	q_3	q_3	q_3	0

Realizando la asignación de estados por defecto y teniendo en cuenta que se utilizan biestables D, con lo que la entrada de excitación del biestable coincide con la salida, tendremos la siguiente tabla:

	D_1D_0				
Q_1Q_0	AB=00	AB=01	AB=10	AB=11	SALIDA
00 (q_0)	01	01	10	10	1
01 (q_1)	01	11	10	11	1
10 (q_2)	11	01	11	10	1
11 (q_3)	11	11	11	11	0

Minimizando utilizando Mapas de Karnaugh, tenemos:

$$D_1 = A + BQ_0 + B'Q_1$$

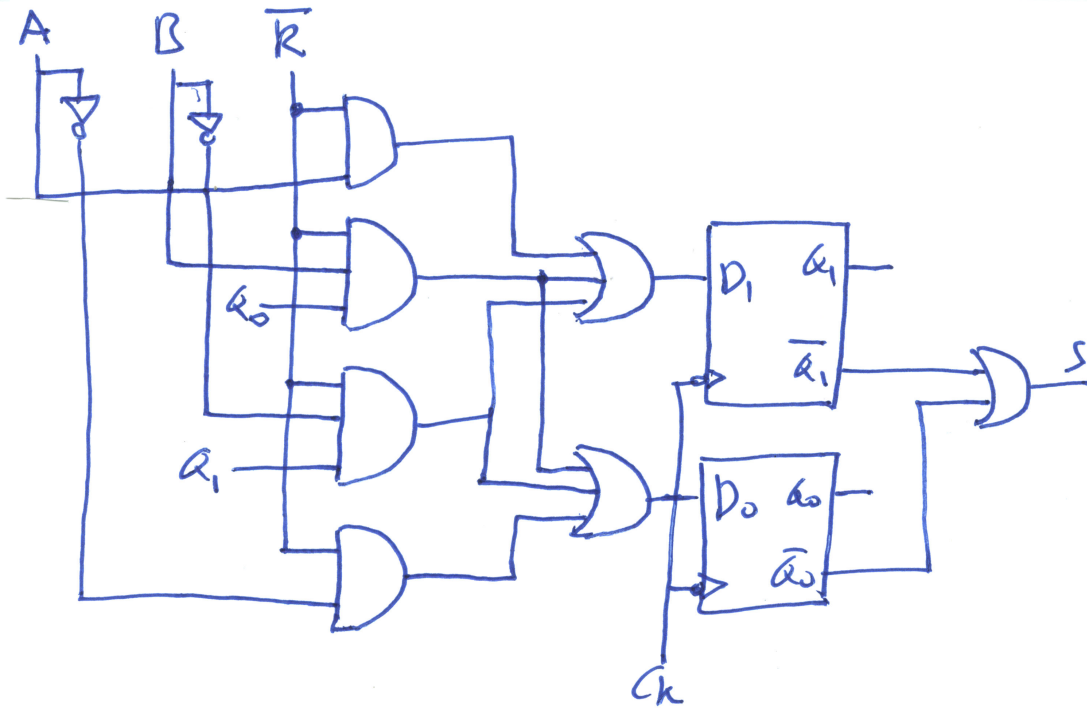
$Q_1Q_0 \backslash AB$	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	1	1	1	1
10	1	0	1	1

$$D_0 = A' + BQ_0 + B'Q_1$$

$Q_1Q_0 \backslash AB$	00	01	11	10
00	1	1	0	0
01	1	1	1	0
11	1	1	1	1
10	1	1	0	1

$$S = (Q_1Q_0)' = Q_1' + Q_0'$$

Para incluir la señal de reset (**R'**) bastará con asegurarnos que volvemos al estado 00 cuando dicha señal esté activa (sea 0). Para ello podemos simplemente multiplicar por la señal **R'** las entradas de ambos biestables. El circuito resultante será:



Solución al ejercicio 2

a)

Hay que tener en cuenta que la instrucción "**ble \$s1, \$v0, L3**" es una pseudoinstrucción que se traduce por:

```
slt $at, $v0, $s1
beq $at, $0, L3
```

y la instrucción "**move \$v0, \$s1**" es otra pseudoinstrucción que se traduce por:

```
add $v0, $0, $s1
```

Dirección	Instrucción	Codificación
0x04000040	addi \$sp, \$sp, -4	8 29 29 -4
0x04000044	sw \$s1, 0(\$sp)	43 29 17 0
0x04000050	lh \$s1, 0(\$a0)	33 4 17 0
0x04000054	slt \$at, \$v0, \$s1	0 2 17 1 0 42
0x04000058	beq \$at, \$0, L3	4 1 0 1
0x0400005c	add \$v0, \$0, \$s1	0 0 17 2 0 32
0x04000068	j L1	2 0x1000013

b)

El procedimiento recibe en \$a0 la dirección de comienzo de un array de enteros de 16 bits y en \$a1 el número de elementos del mismo. Devuelve en \$v0 el valor del mayor elemento del array, o el menor número entero representable con 32 bits si el array tiene 0 elementos.

c)

A partir del código y del volcado de memoria, podemos calcular el número de instrucciones que se ejecutan, teniendo en cuenta que algunas de las instrucciones del programa original son pseudoinstrucciones.

El programa recorre un array de 5 elementos, que son los siguientes:

Representación en memoria (hexadecimal)	Valor (decimal)
0034	52
00a1	161
ffff	-1
0073	115
000a	10

La instrucción "move \$v0, \$s1" se ejecuta sólo para aquellos elementos que son mayores que el mayor elemento encontrado hasta el momento, lo cual ocurre en la primera y en la segunda iteración únicamente, puesto que el máximo es el 161.

La siguiente tabla muestra cuántas veces se ejecuta cada instrucción:

Instrucción	Nº ejecuciones
addi \$sp, \$sp, -4	1
sw \$s1, 0(\$sp)	1
lui \$v0, 0x8000 # li	1
beq \$a1, \$0, L2	6
lh \$s1, 0(\$a0)	5
slt \$at, \$v0, \$s1 # ble	5
beq \$at, \$0, L3	5
add \$v0, \$0, \$s1 # move	2
addi \$a0, \$a0, 2	5
addi \$a1, \$a1, -1	5
j L1	5
lw \$s1, 0(\$sp)	1
addi \$sp, \$sp, 4	1
jrr \$ra	1
Total	44

Por tanto, se ejecutan 44 instrucciones, cada una de las cuales tarda un ciclo en ejecutarse por ser un procesador monociclo. El tiempo de ciclo es $\frac{1}{100 \times 10^6} \text{s} = 10 \text{ ns}$, por lo que el tiempo total de ejecución es de 440 ns.

d)

Las líneas que se quieren eliminar apilan y desapilan el registro \$s1. Es necesario apilar este registro si se utiliza debido a que es un registro preservado entre llamadas. Para evitar el uso de la pila en este procedimiento, sería necesario no utilizar ningún registro de dicho tipo, cambiando todas las referencias a \$s1 por otro registro no preservado entre llamadas (por ejemplo: \$t0).

Solución al ejercicio 3

a)

Lectura:

dir	lat	p1	p2	p3	p4	dir	lat	p1	p2	p3	p4
1	4	2	2	2	2	1	4	2	2	2	2

En total, la lectura de un bloque de memoria principal necesita 26 ciclos de bus.

Escritura:

dir	p1	p2	p3	p4	lat	fin	dir	p1	p2	p3	p4	lat	fin
1	2	2	2	2	4	1	1	2	2	2	2	4	1

En total, la escritura de un bloque en memoria principal necesita 28 ciclos bus.

b)

Ciclos que necesita una instrucción de carga que falla: $26 \text{ ciclos} + 0,40 \cdot 28 \text{ ciclos} = 37,2 \text{ ciclos}$, ya que la lectura de un bloque necesita 26 ciclos y el 40% de las veces hay que expulsar un bloque modificado cuya escritura requiere 28 ciclos de bus.

Ciclos que necesita una instrucción de almacenamiento que falla: $26 \text{ ciclos} + 0,40 \cdot 28 \text{ ciclos} = 37,2 \text{ ciclos}$, ya que, al utilizar una política de postescritura, primero hay que leer el bloque que

contiene la palabra a almacenar y el 40% de las veces hay que expulsar un bloque modificado cuya escritura requiere 28 ciclos de bus.

c)

Del apartado (b) sabemos que el número de ciclos promedio de un fallo de caché, tanto en cargas como en almacenamientos, es 37,2 ciclos. En esos ciclos, se envían $8 + 0,40 \cdot 8 = 11,2$ palabras. Por lo tanto, el máximo ancho de banda efectivo es:

$$AB = (11,2 \cdot 4 / 37,2) \cdot 500 \cdot 10^6 = 602,2 \text{ MB/seg.}$$

d)

Si la caché realiza escrituras directas, entonces, en las operaciones de carga nunca hay que guardar ningún bloque expulsado en memoria. Por lo tanto, los ciclos que necesita una operación de carga que falla son 26, que corresponden a la lectura de un bloque de 8 palabras de memoria principal.

En el caso de las operaciones de almacenamiento, por la escritura directa, siempre se escribe una única palabra que necesita 8 ciclos (1 ciclo para enviar la dirección, 2 ciclos para enviar la palabra, 4 ciclos de latencia y 1 ciclo para finalizar la operación).

Ciclos promedio en un fallo de caché: $0,70 \cdot 26 \text{ ciclos} + 0,30 \cdot 8 \text{ ciclos} = 20,6 \text{ ciclos}$.

Promedio de palabras enviadas = $0,70 \cdot 8 \text{ palabras} + 0,30 \cdot 1 \text{ palabra} = 5,9 \text{ palabras}$.

$$AB = (5,9 \cdot 4 / 20,6) \cdot 500 \cdot 10^6 = 572,8 \text{ MB/seg.}$$

e)

Por interrupciones:

- N° de páginas a transferir al disco duro = $20 \cdot 0,75 = 15$ páginas.
- Total bytes a transferir = $15 \text{ páginas} \cdot 8 \text{ KB/página} = 120 \text{ KB}$.
- N° de interrupciones necesarias = $120 \text{ KB} / (16 \cdot 4 \text{ bytes/interrupción}) = 1920$ interrupciones.
- N° de ciclos para el flush = $1920 \text{ int} \cdot 1000 \text{ ciclos/int} = 1.920.000$ ciclos.

Por DMA:

- N° de operaciones DMA necesarias = n° de páginas modificadas = 15 operaciones de DMA.
- N° de ciclos para el flush = $15 \text{ operDMA} \cdot (500 + 1500) \text{ ciclos/operDMA} = 30.000$ ciclos.

Nota: en los apartados del a) al d) podemos suponer que en una operación de escritura no hay un último ciclo de finalización de operación en el bus, por lo que la escritura de un bloque de 8 palabras en memoria principal necesitaría 26 ciclos de bus y la escritura de una única palabra necesitaría 7 ciclos.