



Estructura y Tecnología de Computadores

Examen final

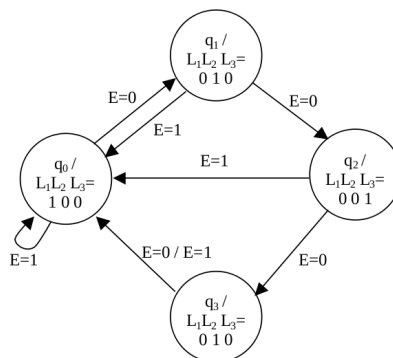
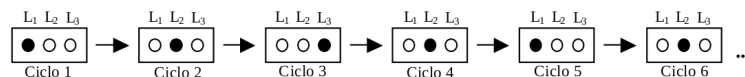
13 de septiembre de 2010

Grupo:	DNI:
Nombre:	Apellidos:

Instrucciones:

- **Sólo en una de las preguntas se podrá obtener una nota inferior a 3.** Si no se cumple esta condición, el examen estará suspenso.
- La nota obtenida en el examen, una vez se divida por 4, representará el 50% de la nota final de la asignatura, siendo necesario obtener una calificación mayor o igual que 5 para eliminar esta parte.
- No se dará por válida ninguna respuesta que no contenga el desarrollo de la solución paso a paso.
- El peso de cada ejercicio aparece entre paréntesis.
- Responda a las preguntas en bolígrafo.

1. (10 puntos) El siguiente autómata de Moore expresa el funcionamiento de un circuito secuencial síncrono que implementa un controlador de un sistema de tres luces (L_1 L_2 L_3) que se van encendiendo sucesivamente de izquierda a derecha y de derecha a izquierda, del modo indicado. Se observa que el autómata tiene una sola entrada E , que hace la función de *reset*, de modo que mientras vale 0 el sistema opera en el modo cíclico habitual anteriormente indicado, pero que cuando vale uno fuerza al sistema a colocarse en su estado inicial (L_1 encendido y L_2 y L_3 apagados).



Diseñe un circuito secuencial síncrono simplificado que implemente dicho autómata, utilizando flip-flops tipo J-K activos en el flanco descendente.

2. (10 puntos) Dado el siguiente fragmento de código MIPS:

```

1      li      $8, 0x10018008
2      move   $9, $0

```

```

3 loop: lb    $10, 1($8)
4         beq  $10, $0, fin
5         blt  $10, $0, neg
6         add  $9, $9, $10
7         move $10, $0
8 neg:    sub  $9, $9, $10
9         addi $8, $8, 4
10        j    loop
11 fin:    sll  $9, $9, 1
12        ...

```

Y suponiendo que la memoria contiene los siguientes valores (en hexadecimal) en las direcciones desde 0x10018004 hasta 0x1001803c:

Dirección	Contenido			
	+0	+1	+2	+3
0x10018004	00	00	01	00
0x10018008	00	05	F0	FF
0x1001800C	00	F6	0F	F0
0x10018010	00	02	00	00
0x10018014	00	F7	00	30
0x10018018	00	FF	8A	3F
0x1001801C	03	03	03	03
0x10018020	0A	07	06	FF
0x10018024	F4	F8	F0	00
0x10018028	00	00	00	00
...resto...	00	00	00	00

- a) Indique el valor que se obtiene en el registro \$9 después de ejecutar dicho código.
- b) Calcule el tiempo que tardará en ejecutarse en un procesador multiciclo como el visto en clase cuyas unidades funcionales tienen las siguientes latencias:
- Unidad de memoria: 30 ns.
 - ALU: 12 ns.
 - Banco de registros (lectura o escritura): 20 ns.

A la hora de resolver el ejercicio, para las instrucciones que no estuvieran presentes en el modelo original del procesador visto en clase suponga que tardan 4 ciclos en ejecutarse si son aritméticas o de desplazamiento, 5 ciclos si son de lectura de memoria y 3 ciclos si son saltos. Además, tenga en cuenta que algunas instrucciones de ensamblador pueden ser realmente pseudoinstrucciones que se implementen con una o más instrucciones reales distintas. Por último, suponga que la implementación MIPS utilizada para ejecutar el programa es *little-endian*.

3. (10 puntos) La máquina M posee un sistema de memoria virtual con las siguientes características:
- Tabla de páginas de 256 entradas.
 - Memoria física de 256 bytes.
 - Tamaño de página de 16 bytes.
 - TLB asociativo con 4 conjuntos de 2 vías (2 bloques por conjunto). El TLB sigue una estrategia de reemplazo LRU y una política de postescritura.

Se pide:

- a) Especificar detalladamente el formato de la dirección física y de la dirección virtual. Dibujar un esquema de la tabla de páginas y calcular su tamaño incluyendo los bits de control necesarios (justificando su inclusión). Se parte de una situación inicial en la que las páginas virtuales 1, 3, 5 y 7 se encuentran cargadas en memoria física en las páginas físicas 0, 1, 2 y 3, respectivamente.
 - b) Dibujar un esquema detallado del TLB y calcular su tamaño incluyendo todos los bits de control necesarios (justificando su inclusión). El TLB se encuentra inicialmente vacío.
 - c) Si durante la ejecución de un programa el procesador genera una lectura a la dirección virtual: 52_{10} , y una escritura a la dirección 190_{10} , indicar cuáles son sus correspondientes NPV (números de página virtual), si habrá acierto o fallo en la tabla de páginas, cuáles son sus NPF (números de página física) y la dirección física resultante. Suponer en este apartado que nuestra máquina no dispone de TLB. (Se sabe que la página física 4 se encuentra libre).
 - d) Repetir el apartado anterior, pero ahora teniendo en cuenta que nuestra máquina dispone de TLB.
4. (10 puntos) Considerar un bus con líneas de datos y direcciones multiplexadas de 32 bits de ancho y un procesador de 500MHz. La memoria tiene una latencia de 3 ciclos. Transmitir una dirección por el bus o una palabra de 32 bits requiere 1 ciclo. Además, el sistema de memoria soporta un modo ráfaga que a partir del 4º ciclo le permite transmitir una palabra por ciclo hasta un máximo de 4. Finalmente, las escrituras requieren un último ciclo para mandar un código de corrección de errores (CRC). Se pide:
- a) Calcular el ancho de banda máximo (en MB/seg) en los casos de que solo haya lecturas, solo haya escrituras, y que haya una mezcla de 65 % de lecturas y 35 % de escrituras.
 - b) Repetir el apartado anterior, pero suponiendo que ahora el bus usa líneas dedicadas para direcciones y datos (32 líneas en cada caso), de manera que al realizar una escritura, puede ponerse una palabra para escribir a la vez que la dirección. Además, el modo ráfaga permite escribir 3 palabras más a 1 por ciclo.
 - c) Supongamos que ahora se conecta al bus un disco duro externo capaz de transmitir 120 MB/seg. Si las transferencias tienen que hacerse mediante sondeo, teniendo en cuenta que una operación de sondeo para transmitir 4 palabras requiere 300 ciclos. ¿Cual debería ser la frecuencia de la CPU para que la transferencia solo requiera una dedicación de la CPU del 15 %?
 - d) Si en lugar de sondeo, se hace uso de interrupciones para avisar a la CPU cada vez que se transmiten 4 palabras, cuyo tratamiento implica 50 ciclos. ¿Que porcentaje del tiempo de procesamiento de la CPU se dedicaría a esta tarea?
 - e) Finalmente, suponer que en lugar de sondeo o interrupciones, se hace uso de una controladora de DMA para controlar la transferencia de los datos. La controladora trabaja con bloques de 4KB, su inicialización requiere 50 ciclos, y el tratamiento de la interrupción al terminar cada transferencia otros 100 ciclos. En este caso, ¿cual será el porcentaje de tiempo que la CPU dedica a controlar la transferencia de los datos?

Nota: tened en cuenta que el prefijo M expresa una potencia en base 10 (10^x) cuando se refiere a la frecuencia del procesador, mientras que en el resto de casos, los prefijos M y K expresan potencias en base 2 (2^x).

Soluciones

Ejercicio 1

Dado que tenemos 4 estados, necesitaremos dos flip-flops para codificarlos (Q_0 y Q_1). Utilizaremos la siguiente codificación para los estados:

Estado	Q_0	Q_1
q ₀	0	0
q ₁	0	1
q ₂	1	0
q ₃	1	1

Tendremos tres funciones de salida (o una función de 3 bits) cuya tabla de verdad será:

Q_0	Q_1	L_1	L_2	L_3
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	0	1	0

Y la tabla de la función de transición (Q_0^* y Q_1^*) y las entradas de los dos biestables (J_0, K_0, J_1 y K_1) será:

Q_0	Q_1	E	Q_0^*	Q_1^*	J_0	K_0	J_1	K_1
0	0	0	0	1	0	—	1	—
0	0	1	0	0	0	—	0	—
0	1	0	1	0	1	—	—	1
0	1	1	0	0	0	—	—	1
1	0	0	1	1	—	0	1	—
1	0	1	0	0	—	1	0	—
1	1	0	0	0	—	1	—	1
1	1	1	0	0	—	1	—	1

Las expresiones simplificada de las funciones de salida serán:

$$\begin{aligned}
 L_1 &= \overline{Q_0} \overline{Q_1} \\
 L_2 &= \overline{Q_0} Q_1 + Q_0 Q_1 = Q_1 \\
 L_3 &= Q_0 \overline{Q_1}
 \end{aligned}$$

La simplificación de la función J_0 es (otras soluciones también son posibles):

		Q_0Q_1			
		00	01	11	10
E	0	0	1	X	X
	1	0	0	X	X

$$J_0 = Q_1 \overline{E}$$

La de K_0 :

		Q_0Q_1			
		00	01	11	10
E	0	X	X	1	0
	1	X	X	1	1

$$K_0 = Q_1 + E$$

La de J_1 :

		Q_0Q_1			
		00	01	11	10
E	0	1	X	X	1
	1	0	X	X	0

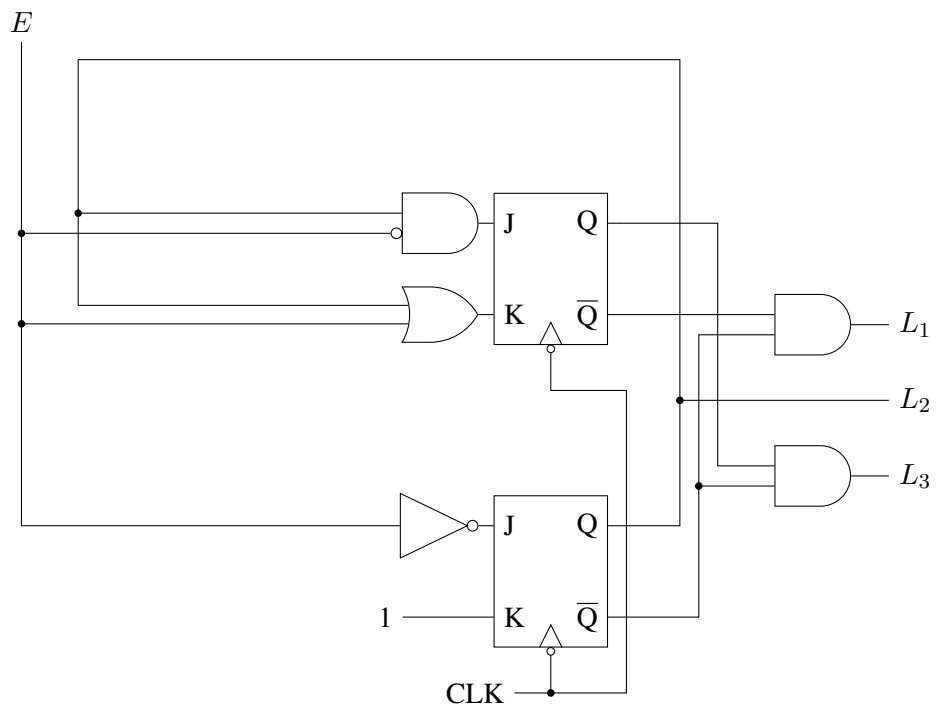
$$J_1 = \overline{E}$$

Y la de K_1 :

		Q_0Q_1			
		00	01	11	10
E	0	X	1	1	X
	1	X	1	1	X

$$K_1 = 1$$

Con lo que el circuito resultante usando puertas AND, OR y NOT será:



Ejercicio 2

Apartado a

El código tiene un bucle en el que se van leyendo los bytes apuntados por $\$8 + 1$ (línea 3). El registro $\$8$ se inicializa a $0x10018008$ (línea 1) y se incrementa en 4 en cada iteración (línea 9). El bucle termina cuando el elemento leído es un 0 (línea 4). La instrucción `lb` extiende el signo del byte leído de memoria, por lo que los valores leídos son 5, -10, 2, -9, -1, 3, 7, -8 y 0.

Para cada valor leído, se comprueba si es negativo (línea 5). Si no es negativo, el valor se suma al registro $\$9$ (línea 6), y si es negativo se resta (línea 8). La línea 7 evita que la línea 8 tenga ningún efecto en el caso de los valores positivos. Por tanto, teniendo en cuenta que $\$9$ se inicializa a 0 antes del bucle (línea 2), al final del bucle $\$9$ contendrá la suma de los valores absolutos de los valores leídos. Finalmente, la línea 11 multiplica el valor de $\$9$ por 2.

Con los datos de este problema, el valor final de $\$9$ será 90.

Apartado b

El tiempo de ejecución del programa vendrá dado por la expresión:

$$T_{ejec} = N_{inst} \times CPI \times T_{ciclo} = N_{ciclos} \times T_{ciclo}$$

El tiempo de ciclo (T_{ciclo}) está determinado por la unidad funcional más lenta, que en este caso es la memoria que tarda 30 ns.

Para hallar el número total de ciclos necesarios para la ejecución del programa, tendremos que seguir la ejecución del mismo paso a paso y contar cuántas veces se ejecuta cada instrucción.

Habrà que tener en cuenta que la primera instrucción (`li`) es en realidad una pseudoinstrucción que se traduce por dos instrucciones de código máquina (un `lui` y un `ori`), que las instrucciones de las líneas 2 y 7 (`move`) son también pseudoinstrucciones que se traducen con `add`, y que la instrucción de la línea 5 es otra pseudoinstrucción que se traduce con un `slt` y un `bne`.

Por tanto:

Instrucción	Ejecuciones	Ciclos por ejecución	Ciclos totales
<code>lui</code>	$\$8, 0x1001$	1	4
<code>ori</code>	$\$8, 0x8008$	1	4
<code>add</code>	$\$9, \$0, \$0$	1	4
loop: <code>lb</code>	$\$10, 1(\$8)$	9	45
<code>beq</code>	$\$10, \$0, fin$	9	27
<code>slt</code>	$\$at, \$10, \$0$	8	32
<code>bne</code>	$\$at, \$0, neg$	8	24
<code>add</code>	$\$9, \$9, \$10$	4	16
<code>add</code>	$\$10, \$0, \$0$	4	16
neg: <code>sub</code>	$\$9, \$9, \$10$	8	32
<code>addi</code>	$\$8, \$8, 4$	8	32
<code>j</code>	loop	8	24
fin: <code>sll</code>	$\$9, \$9, 1$	1	4
			Total: 264

Por tanto:

$$T_{ejec} = N_{ciclos} \times T_{ciclo} = 264 \times 30ns = 7.920\mu s$$

Ejercicio 3

Ejercicio 4