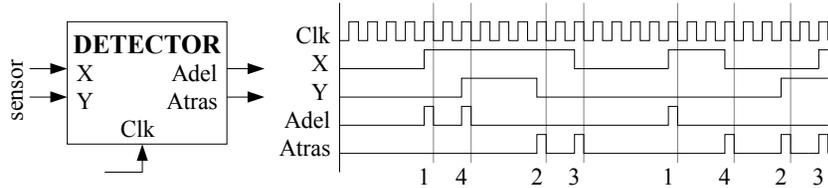


**CC41C Introducción al Hardware**  
**Examen - Otoño 2006**  
**Profesor: Luis Mateu**

**Pregunta 1**

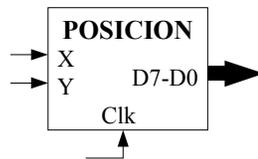
- a) Un sensor permite traducir el movimiento de un objeto en alteraciones en los valores de las líneas digitales X e Y, las que sirven de entrada al circuito DETECTOR de la figura. El diagrama de tiempo adjunto muestra cómo DETECTOR debe procesar X e Y para señalar en cada ciclo del reloj si el objeto dio (i) un paso hacia adelante, colocando un 1 en la salida ADEL, (ii) un paso hacia atrás, colocando un 1 en la salida ATRAS, o (iii) si no registró movimiento, colocando ambas líneas en 0.



Supongamos que X e Y son iguales. Un paso adelante ocurre cuando X cambia primero que Y (1), un paso atrás cuando Y cambia primero que X (2). Nunca cambian en el mismo ciclo del reloj. Cuando X e Y son distintos, un paso atrás ocurre cuando cambia X (3), un paso adelante cuando cambia Y (4).

Confeccione el *diagrama de estados* para el circuito DETECTOR. No necesita implementar este circuito.

- b) Utilizando diseño modular *implemente* el circuito POSICION de la figura. Este circuito entrega en D7-D0 la posición absoluta de un objeto en movimiento, contabilizando los pasos hacia adelante y descontando los pasos hacia atrás. Para ello recibe como entradas las mismas señales X e Y que provienen del sensor de la parte a). Incluya el DETECTOR de la parte a) para traducir X e Y en señales que indiquen si un registro se debe incrementar o decrementar. Se requiere que su diseño sea *eficiente* en el número de transistores.



**Pregunta 2**

- I. Diseñe una interfaz para el circuito POSICION de la pregunta 1 que permita que un programa pueda leer la posición absoluta de un objeto en

movimiento. Diseñe su interfaz para un microcontrolador con un bus de direcciones de 16 bits y un bus de datos de 8 bits.

- II. Escriba el procedimiento `leerPosicion()` que entrega la posición leída desde la interfaz de la parte I.
- III. Modifique el diseño de tal forma que ahora se comunique al procesador cualquier cambio en la posición mediante una *interrupción*. Para ello muestre todas las componentes que hay que *agregar* a POSICION y a la interfaz de la parte I. Tenga presente que dado que un programa puede inhibir las interrupciones, la línea de interrupción se debe activar hasta que la rutina de atención explícitamente solicite llevar la línea de interrupción a 0. Explique.

**Pregunta 3**

- i. La figura muestra un extracto del contenido de un *cache* de 8 KB de 2 grados de asociatividad y líneas de 16 bytes. El cache se organiza en 2 bancos, cada uno con 256 líneas. Por ejemplo en la línea 3b (en hexadecimal) del banco izquierdo se almacena la línea de memoria que tiene como etiqueta 53b (es decir, la línea que va de la dirección 53b0 en hexadecimal a la dirección 53bf).

línea cache	Banco 1		Banco 2	
	etiqueta	contenido	etiqueta	contenido
0a	b0a		80a	
3b	53b		13b	
81	481		681	

Un programa accede a las siguientes direcciones de memoria: b0a4, 13b0, 10a8, 4810, 6810, 4818, b0a, 80a. Indique qué accesos caen en la memoria cache y cuales no. Además muestre un posible estado del cache después de aquellos accesos.

- ii. La figura muestra a la izquierda un programa que se ejecuta en un procesador Risc, con una implementación superescalar de grado 2 y 5 etapas en sus 2 pipelines: *fetch, decode, analyze, execute* y *store*. Acceder a la memoria toma 2 ciclos del reloj.

```

loop:
  shiftl R1, 2, R2
  load [R6+R2], R3
  add R1, 1, R1
  add R3, R4, R4
  cmp R1, R5
  blt loop
  ...
  
```

instrucción	ciclo				
	1	2	3	4	5
shiftl R1,2,R2	F	D	A	...	
load [R6+R2],R3	F	D	A		
add R1,1,R1			F	D	
add R3,R4,R4				F	D
.				.	
.				.	

Complete el diagrama de ejecución de la derecha, considerando 2 iteraciones del ciclo `loop`. Concluya sobre la eficacia de un procesador superescalar para el caso de este programa.