

CC41C Introducción al Hardware

Examen - Otoño 2005

Profesor: Luis Mateu

Pregunta 1

Se desea agregar la instrucción STWx2 a M32 (*store 2 words*). Esta instrucción escribe 2 registros de M32 en 2 palabras consecutivas de la memoria. Por ejemplo:

```
STWx2 R3, R5, [R10]      ! M[R10]:= R3, M[R10+4]:= R5
```

Los dos primeros parámetros (R3 y R5) especifican los registros que se escriben. El tercer parámetro (R10) es un registro con la dirección de la palabra en donde debe quedar el primer parámetro (R3). El segundo parámetro queda en la siguiente dirección (es decir R10+4). La siguiente figura muestra la codificación de STWx2 en comparación con LDW y ADD:

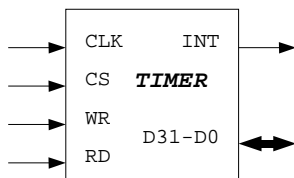
instrucción en assembler	cod. op.	reg. dest.	reg. src. 1	reg. src. 2	val. imm.	significado
STWx2 R3, R5, [R10]	24	10	3	0	5	M[R10] := R3, M[R10+4] := R5
ADD R3, R5, R10		10	3	0	5	R10 := R3+R5
LDW [R3+112], R10		10	3	1	112	R10 := M[R3+112]
	31	23	18	13		

Esta instrucción no se puede implementar con el diseño físico actual de M32, puesto que R5 aparece en la posición *reg. src. 2*. Los registros que aparecen en esa posición en M32 sólo pueden ser el segundo operando de alguna operación de la ALU.

a.- (2 puntos) Modifique el diseño físico de M32, para que esta instrucción sea implementable. Explique.

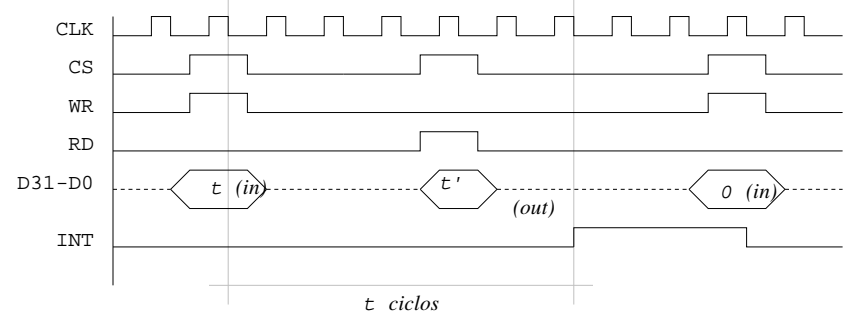
b.- (4 puntos) Indique ciclo por ciclo las señales de control necesarias para ejecutar STWx2 (no incluya la fase de fetch ni la fase de decodificación). Indique además para cada ciclo las transferencias entre registros que se realizan.

Pregunta 2



El *timer* de la figura es un dispositivo que se programa para que produzca una interrupción al cabo de una cierta cantidad de ciclos del reloj. Para programarlo se coloca simultáneamente un 1 en la entrada CS, un 1 en WR y la cantidad de ciclos t en

D31-D0. Como se muestra en el diagrama de tiempo de la figura, el timer activará la línea INT después de t ciclos del reloj, la que permanecerá en 1 hasta que se *desactive* el timer. El timer se *desactiva* colocando un 1 en CS, un 1 en WR y un 0 en D31-D0. También se puede recuperar la cantidad de ciclos que restan para que se active la interrupción, lo que se logra colocando un 1 en CS y un 1 en RD. La cantidad de ciclos restantes t' aparece por D31-D0 (0 si el timer está desactivado).



a.- Implemente una interfaz que conecte el *timer* de la figura con un procesador M32. Haga que cada vez que se escribe un dato t en la dirección 0xff ff 00 00 se programa el timer para que interrumpa en t ciclos del reloj, si es que t es mayor que 0, o se desactiva el timer, si t es cero. Además haga que cada vez que se lea esa misma dirección se obtiene la cantidad de ciclos que restan para la interrupción.

b.- Programe en C los siguientes procedimientos:

- `progTimer(int t, void (*f)())`: programa el timer para que produzca una interrupción en t ciclos del reloj. Además registra el procedimiento f para que se invoque cuando ocurra la interrupción. Si t es 0, se desactiva el timer.
- `handleTimer()`: rutina de atención de la interrupción que *desactiva el timer* e invoca el procedimiento f .

(Simplificación: la invocación de `progTimer` cancela cualquier otra invocación previa de `progTimer`.)

Pregunta 3

Implemente el timer de la pregunta 2 utilizando técnicas de diseño modular. Puede usar las componentes de diseño modular vistas en clases, como multiplexores, registros, sumadores, etc. Si usa un circuito que resta, debe implementarlo. Si necesita circuitos secuenciales o combinatoriales, no necesita implementarlos, solo especifíquelos con un diagrama de estados o una tabla de verdad.

Hint: Utilice un circuito secuencial que distinga los estados activo y desactivo. Este circuito tiene como salidas la línea INT y las señales que controlan el funcionamiento de las componentes de su circuito.