Auxiliar 10

Profesor: Pablo Guerrero.
Auxiliar: Ian Yon
Viernes 24 de octubre de 2014

Departamento de Ciencias de la Computación
UNIVERSIDAD DE CHILE

Problema 1: Control 3 2011-1

Se desea agregar la instrucción SWAP a M32. Esta instrucción intercambia el valor de un registro con el de una palabra en memoria, es decir:

Notación assembler	Instrucciones	Formato
SWAP $[r_1 + \{imm \mid r_2\}], r_d$	aux ← Mem$[r_1 + \{imm \mid r_2\}]$ Mem$[r_1 + \{imm \mid r_2\}]$ ← r_d r_d ← aux	$31-24$ *Instrucción* $23-19$ r_d $18-14$ r_1 13 *Det. si imm o reg* $12-0$ *imm o* r_2[1]

1. Explique porqué no es posible implementar esta instrucción con el actual diseño de M32.
2. Modifique ligeramente M32 de tal forma que sí se pueda implementar SWAP, especificando componentes y señales de control adicionales.
3. Especifique ciclo por ciclo las señales de control que son necesarias para ejecutar SWAP. No es necesario que especifique las señales para la carga de la instrucción y la decodificación.

Problema 2: Control 3 2011-1

En la figura 3 se muestra un circuito guardián que se encarga de vigilar que la temperatura ambiente no salga de un rango dado. El guardián tiene dos puertas de E/S que permiten especificar la mínima y la máxima temperatura aceptable. Cuando la temperatura se sale de estos parámetros, el sensor activa la línea INT.

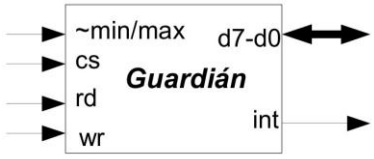

Figura 1: Circuito guardián

El siguiente procedimiento sirve para configurar el guardián:

[1] Si es r2 entonces los bits 7 a 0 no están especificados.

```
void setTempRange(char min, char max){
	/*min y max son enteros de un byte*/
	char *port_min = (char*)0xFF00;
	char *port_max = (char*)0xFF01;
	*port_min = min;
	*port_max = max;
}
```

Los valores 0 y 255 se usan para indicar al guardián que no debe producir ninguna interrupción.

1. Implemente la interfaz que se necesita para conectar el guardián con el bus del procesador, de modo que el procedimiento anterior funcione correctamente. Considere un microcontrolador con un bus de datos de 8 bits y un bus de direcciones de 16 bits.
2. Implemente en C la rutina de atención alertTemp() que debe procesar la interrupción causada por el guardián. Esta rutina debe desplegar un mensaje en pantalla (usando printf) y evitar que se siga produciendo la interrupción.
3. (Propuesta) Considere que cuando se lee en las posiciones de timer se lee el valor de la máxima y mínima temperatura configuradas en el guardián. Ahora, implemente la función pedida en el punto anterior, pero después de imprimir, que se restauren las interrupciones y deje al guardián con la misma configuración que antes de la interrupción.

Problema 3: entrada y salida con interrupciones. Examen 2005-1

En la figura se muestra el dispositivo TIMER, que se programa para que produzca una interrupción al cabo de t 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.

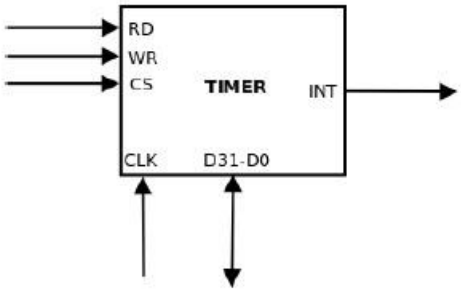

Figura 2: Módulo TIMER

Como se muestra en el diagrama de tiempo, 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_0 aparece por D31 - D0 (0 si el timer está desactivado).

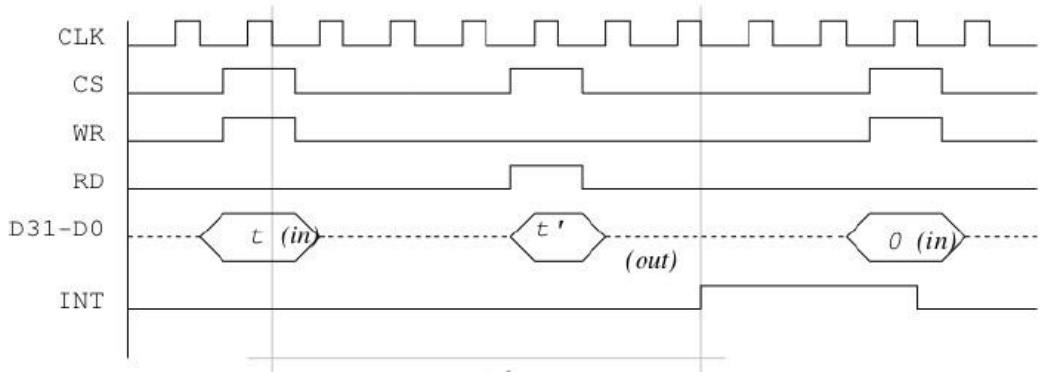

Figura 3: Diagrama de tiempo TIMER

1. 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 0xffff0000 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.

2. Programe en C los siguientes procedimientos:

   a. $progTimer(int\ t, void\ (*f)())$: 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.

   b. $handleTimer()$: rutina de atención de la interrupción que desactiva el timer e invoca el procedimiento f.