

CC4301 Arquitectura de Computadores

Auxiliar 11

Prof. Aux.: Gaspar Pizarro V.

12 de noviembre de 2012

1. P1 Examen 2005-1

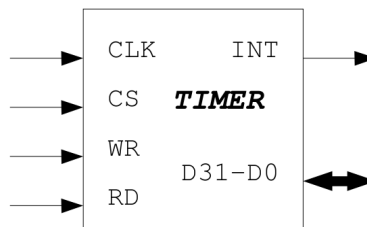


Figura 1: Circuito timer

El timer de la figura 1 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 2, 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).

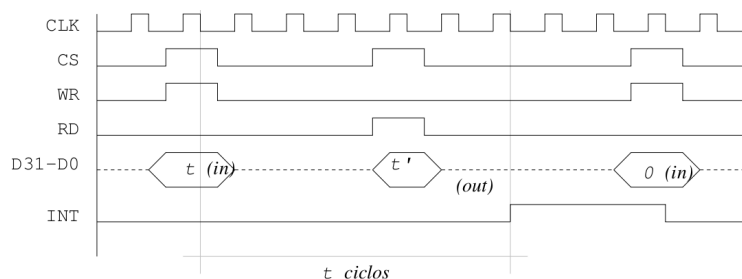


Figura 2: Diagrama de tiempo de timer

1. Implemente una interfaz que conecte el timer de la figura con un procesador M32. Haga que cada vez que se escriba un dato t en la dirección `0xFFFF0000` se programe el timer para que interrumpa en t ciclos del reloj, si es que t es mayor a cero, o se desactive el timer si t es cero. Además haga que cada vez que se lea esa misma dirección se obtengan la cantidad de ciclos que restan para la interrupción.
2. Programe en C los siguientes procedimientos:

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

2. P2 Control 3 2011-1

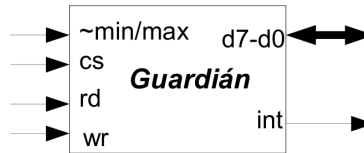


Figura 3: Circuito guardián

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.

El siguiente procedimiento sirve para configurar el guardián.

```
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. (*original*) 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.