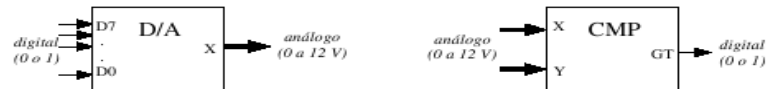


## Auxiliar: Entrada/Salida CC41C

Rodrigo Canovas

22 de Junio del 2006

### 1. Problema 1

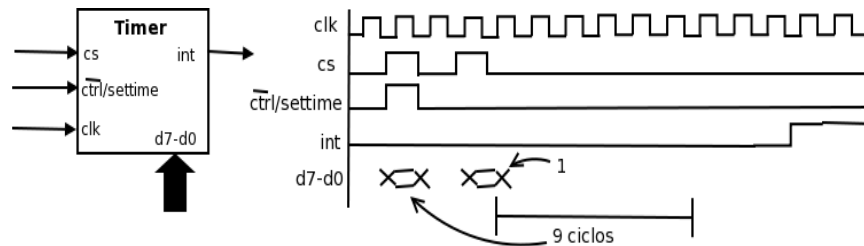


La figura muestra un convertor digital/análogo (D/A) y un comparador (CMP). El convertor digital/análogo recibe como entrada un número entero binario en D7-D0 (entre 0 y 255) y entrega en X una señal analógica consistente en un voltaje entre 0 y 12V proporcional al valor de la entrada. El comparador recibe dos valores analógicos X e Y y los compara entregando un valor binario 1 que indica que X es mayor que Y o 0 en caso contrario.

- Diseñe e implemente una interfaz de entrada/salida para ambas componentes. Considere un microprocesador con un bus de datos de 8 bits y un bus de direcciones de 16 bits. Haga que al escribir en la dirección 0xffff se establezca la entrada del convertor D/A (preocúpese que esta entrada se mantenga constante hasta que se reescriba un nuevo valor) y al leer en la misma dirección se lea la salida GT de CMP. Las señales X e Y salen del sistema para ser conectadas por el usuario.
- Explique cómo un usuario puede usar estas componentes para convertir una señal analógica entre 0 y 12V en un valor binario entre 0 y 255. Escriba en C la rutina **convertirAD()** que hace esta conversión (busque la simplicidad en esta rutina y no la eficiencia).

## 2. Problema 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. El *timer* se programa colocando simultáneamente un 1 en la entrada *cs*, un 1 en *ctrl/settime* y la cantidad de ciclos *t* en *d7-d0*. Para activar el conteo se debe colocar un 1 en *cs*, un 0 en *ctrl/settime* y un 1 en *d7-d0*. El *timer* activará la línea *int* después de *t* ciclos del reloj. Para detener el *timer* se coloca un 1 en *cs*, un 0 en *ctrl/settime* y un 0 en *d7-d0*.



- Implemente el dispositivo *timer* usando diseño modular. Use sumadores, multiplexores, registros (flip/flop data), etc. Si necesita un circuito secuencial, sólo indique las entradas, salidas y el diagrama de estados.
- Implemente una interfaz que conecte el *timer* de la figura con un procesador que posee un bus de datos de 8 bits y un bus de direcciones de 16 bits. En este procesador los accesos a la memoria duran exactamente un ciclo del reloj. Haga que cada vez que se escriba un dato *t* en la dirección 0xf800 se especifique *t* como lapso para la interrupción, y que cada vez que se escriba *b* en la dirección 0xf801 se active o detenga el conteo según *b* sea 1 o 0 respectivamente.
- Programa el siguiente procedimiento en C:

```
void progTimer(char time){
    ...
}
```

Cuando se invoque este procedimiento son un valor positivo para *time*, se debe programar el *timer* en ese número de ciclos y activar el conteo. Si *time* es 0, se debe detener el conteo.