

# Auxiliar 6

Profesor: Pablo Guerrero.

Auxiliar: Ian Yon

viernes, 15 de mayo de 2015



## Problema 1

Se dispone de chips de memoria de  $128k \times 8b$ . Diseñe una interfaz de memoria que provea 512KB a un procesador con un bus de datos de 8b y un bus de direcciones de 19b.

## Problema 2: Entrada - Salida

Ud. dispone del teclado de la Figura 1, que está organizado como una matriz de 16 columnas de 6 teclas cada una. Para leer el estado de una columna completa (6 teclas), se suministra como entrada en  $C_3-C_0$  el número de la columna. El teclado entrega en  $K_5-K_0$  el estado de las 6 teclas pertenecientes a la columna. No es posible leer dos columnas al mismo tiempo.

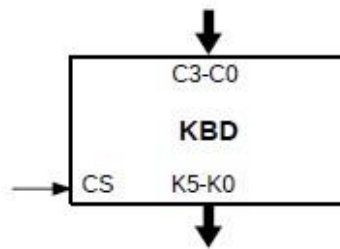


Ilustración 1: Circuito KBD

1. Construya una interfaz para este teclado considerando un computador con un bus de direcciones de 16 bits y un bus de datos de 8 bits. Su interfaz debe poseer 16 puertos de entrada en las direcciones  $0xFF00$  a  $0xFF0F$  (en hexadecimal). Cada uno de estos puertos sirve para leer una columna del teclado. No implemente KBD.
2. Programe la rutina *int consultar tecla(int fila, int columna)*, que entrega *verdadero* si la tecla de coordenadas (fila, col) está presionada.

## Problema 3: entrada y salida con interrupciones

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 enciende simultáneamente la entrada CS, WR y se especifica la cantidad de ciclos  $t$  en  $D_{31}-D_0$ .

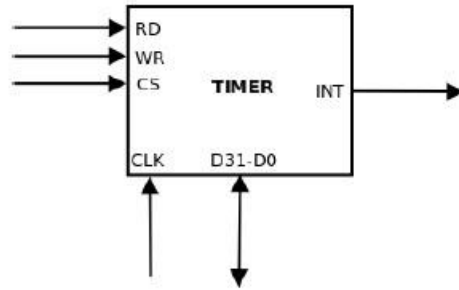


Ilustración 2: Módulo TIMER

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 encendiendo CS, WR y poniendo ceros en  $D_{31}-D_0$ . También se puede recuperar la cantidad de ciclos que restan para que se active la interrupción, lo que se logra encendiendo CS y RD. La cantidad de ciclos restantes  $t_0$  aparece por  $D_{31}-D_0$  (0 si el timer está desactivado).

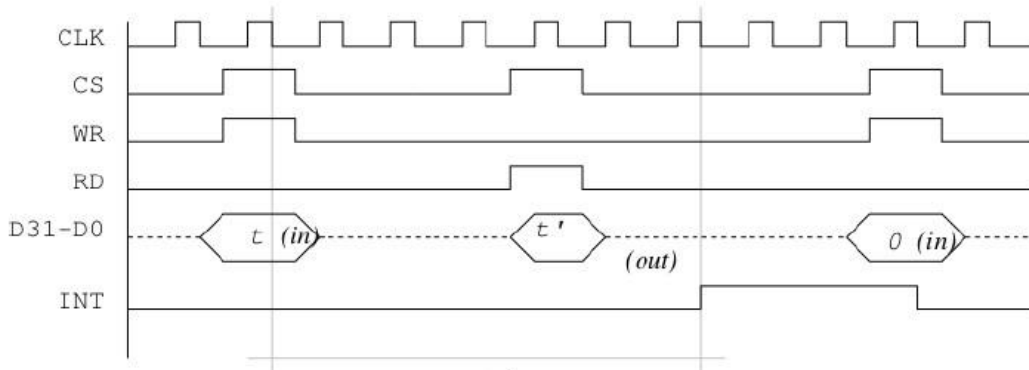


Ilustración 3: Diagrama de tiempo TIMER

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  $t > 0$ , o se desactiva el timer, si  $t = 0$ . 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  $f$ .