

Sistemas complejos

Problema 1

En un sistema de transmisión de datos síncrono se transmiten los datos en tramas de 16 bits. Cada trama de datos va precedido de una cabecera de 4 bits y seguido de un campo checksum de 4 bits (en total la trama tiene 24 bits).

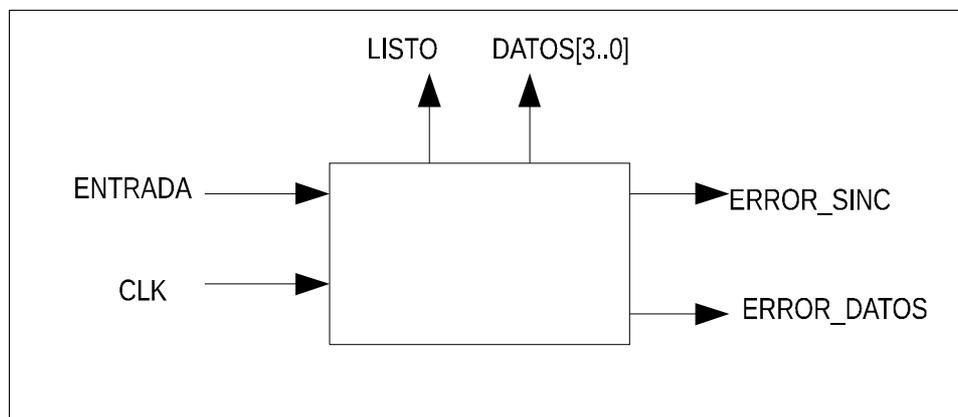
La cabecera será siempre una secuencia 1010 (para recuperar el sincronismo) y el campo checksum será la suma de todos los bits de datos en grupos de 4 bits, despreciando el acarreo. Veamos un ejemplo:

cabecera	datos	checksum
1010	0010 0111 0001 1000	0010

- La cabecera es siempre 1010
- El campo de datos son 16 bits, o visto de otra forma 4 datos de 4 bits, en decimal los números 2, 7, 1 y 8.
- La suma de los 4 datos de 4 bits es $2+7+1+8=18$, si se realiza la suma en 4 bits despreciando el acarreo, el resultado de esta suma es 0010 que es el campo checksum.

El diagrama de bloques del sistema es el que se detalla a continuación, donde el significado de cada una de las líneas es el siguiente:

- ENTRADA: línea por donde entran en serie los datos de la trama al ritmo marcado por la señal de reloj.
- CLK: señal de reloj.
- DATO[3..0]: salida de datos en paralelo.
- LISTO: señal que indica que hay un dato disponible en la salida paralelo.
- ERROR_SINC: señal que indica que se ha producido un error de sincronismo (permanecerá activa un ciclo de reloj).
- ERROR_DATOS: señal que indica que se ha producido un error de datos (permanecerá activa un ciclo de reloj).



El funcionamiento del sistema es el siguiente:

1. Los datos van entrando en serie por la línea ENTRADA, primero entrarán los 4 datos de la cabecera. Se comprobará si ha llegado correcta la secuencia 1010. Si no ha llegado correcta, se activará la señal ERROR_SINC durante un periodo de reloj y se vuelve a empezar a leer la cabecera (ya que el sistema que envía los datos, al avisarle de un error en la cabecera empieza a transmitir de nuevo). Si la cabecera es correcta, pasamos al punto 2.
2. Se leen 4 bits consecutivos. Una vez que se han leído, se sacan por la salida DATO[3..0] y se activa la salida LISTO. Se realiza este paso 4 veces y a la vez se van sumando los 4 datos para calcular el checksum.
3. Se leen 4 bits consecutivos del campo checksum. Una vez que se han leído, se compara el checksum leído con el checksum calculado (en el paso 2). Si no coinciden se activa la salida ERROR_DATOS durante un ciclo de reloj.
4. Se vuelve al paso 1 y se repite indefinidamente.

Se pide realizar el diseño de la arquitectura del sistema y el sistema de control (no intentar hacer todo con un autómata). Para la arquitectura del sistema se recomienda usar (entre otros) registros de desplazamiento, contadores de 4 bits y comparadores.

Problema 2

Se va a realizar un termómetro de máximas y mínimas. Para ello disponemos de un sensor de temperatura que da a su salida una tensión de $333\text{mV}/^\circ\text{C}$. Se utilizará un convertidor analógico/digital para convertir esa tensión en un código digital de 4 bits según la tabla que se adjunta.

El sistema almacenará la temperatura máxima y la mínima. Se tomará una muestra de temperatura cada 20 segundos. En cada caso se comprobará si la temperatura es la máxima o la mínima, almacenando estas en su caso.

El sistema tendrá 3 displays de 7 segmentos en los que mostrará en todo momento la temperatura actual, la máxima y la mínima.

Se pide:

1. Justificar detalladamente cómo se conectará el ADC para conseguir obtener el código digital según la tabla adjunta.
2. Realizar el circuito con los dispositivos secuenciales y combinacionales que se crean necesarios (no puede utilizarse lógica programable). El sistema de control, en caso de que sea necesario, se podrá realizar con un autómata, en cuyo caso se deberá realizar el diagrama de estados y se podrá elegir entre realizarlo con biestables o hacerlo con una PAL programada en ABEL

T(°C)	tensión (V)	código
0	0,0	0000
1	0,3	0001
2	0,7	0010
3	1,0	0011
4	1,3	0100
5	1,7	0101
6	2,0	0110
7	2,3	0111
8	2,7	1000
9	3,0	1001

Problema 3

Para la realización de un sistema de medida de temperaturas remotas, disponemos de un sensor LM35, como el de la figura.

Se quieren medir temperaturas entre 0 y 100°C con una precisión de 8 bits, de forma que el código 00000000 se corresponda con 0°C y el código 11111111 se corresponda con 100°C.

Se realizará una adquisición de temperatura cada dos minutos, se almacenará y se enviará en serie (empezando por el bit más significativo) a una velocidad de 1 bit por segundo.

Typical Applications

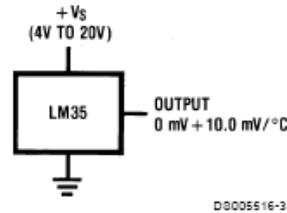


FIGURE 1. Basic Centigrade Temperature Sensor

Se pide:

1. Justificar detalladamente cómo se conectará el ADC para conseguir obtener el código digital según se pide.
2. Explicar detalladamente el circuito que realiza la conversión del dato de paralelo a serie y la temporización necesaria.
3. Realizar el diseño completo de la arquitectura (no puede hacerse con lógica programable) y el sistema de control. El sistema de control, en caso de que sea necesario, se podrá realizar con un autómata, en cuyo caso se deberá realizar el diagrama de estados y se podrá elegir entre realizarlo con biestables o hacerlo con una PAL programada en ABEL

Problema 4

Se quiere realizar un circuito multiplicador de dos números de 4 bits A y B. Para ello tendremos en cuenta lo siguiente:

- Los números A₃..A₀ y B₃..B₀ serán enteros sin signo.
- El resultado de la multiplicación debe tener 8 bits, será M₇..M₀
- La multiplicación de dos números se hará como una serie de sumas consecutivas: multiplicar A x B es sumas A veces B. Cada suma se hará en un ciclo de reloj y se almacenará el resultado parcial (la multiplicación durará A ciclos de reloj)
- El circuito tendrá aparte de las dos entradas A₃..A₀ y B₃..B₀ una entrada INICIO, activa a nivel alto, que indicará (con un pulso de un periodo de reloj de duración) cuando están los datos disponibles para empezar a calcular.
- El circuito tendrá, aparte de sus salidas M₇..M₀, una salida FIN activa a nivel alto, que se activará cuando el dato se haya calculado y esté disponible en las salidas M₇..M₀.

Se pide realizar el circuito con los dispositivos secuenciales y combinacionales que se crean necesarios (no puede utilizarse lógica programable). El sistema de control se podrá hacer con un autómata en cuyo caso se realizará su diagrama de estados y se realizará con biestables y una memoria

Se recomienda usar (entre otros) los siguientes circuitos: 7483, 74373 (o 74374), 74191

Problema 5

Se va a realizar el software de un robot sigue-lineas con un PIC16F84. El hardware del robot consta de 4 sensores CNY70, que detectan líneas negras sobre blanco, y dos motores paso a paso. Las líneas tienen la anchura para que la capten dos sensores a la vez.

Los sensores devuelven un 0 en el puerto cuando detectan negro y un 1 para blanco y van conectados así:

Puerto	sensor
RA0	Sensor derecho
RA1	Sensor central derecho
RA2	Sensor central izdo
RA3	Sensor izquierdo

Los motores van conectados al puerto B, el motor derecho en RB7..RB0 y el izquierdo en RB3..RB0. Para que un motor gire hacia delante hay que activar sus entradas con esta secuencia 1001->1100->0110->0011, y para que gire para atrás la misma secuencia pero en sentido contrario.

I	CI	C	D	Motor izquierdo	Motor derecho
	■	■		Gira delante	Gira delante
■	■			Gira delante	parado
■				Gira delante	Gira detrás
		■	■	parado	Gira delante
			■	Gira detrás	Gira delante
				Gira detrás	Gira detrás

El robot debe seguir una línea negra sobre fondo blanco. Para ello se utilizarán los sensores y los motores según la siguiente tabla, siendo la velocidad de giro, cuando los motores giran, de 20 pasos por minuto.

Se supone que el PIC16F84 va a funcionar con un cristal LP de 32Khz. Sensor CNY70 y montajes típicos.

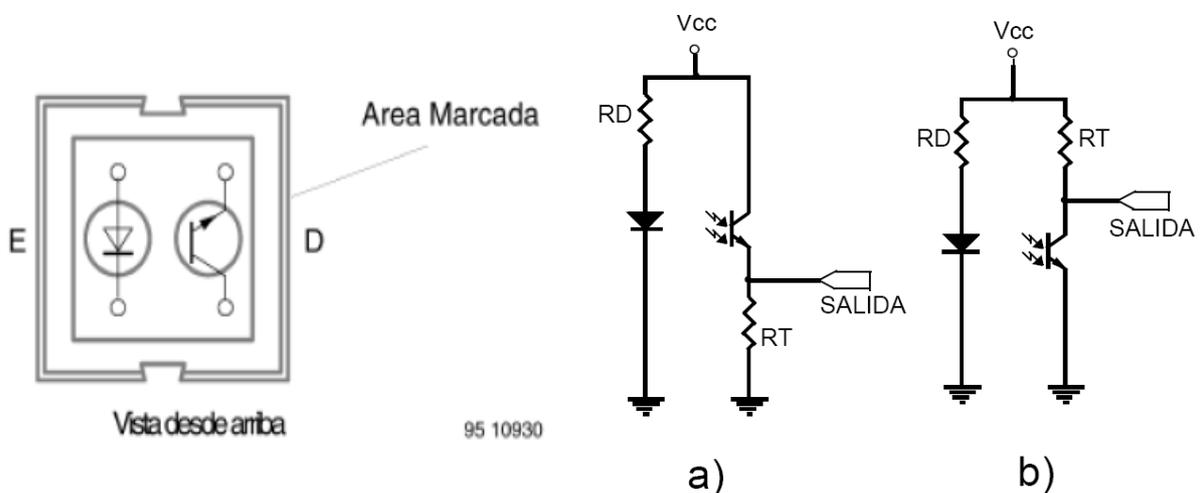


Figura 2.- Circuitos de aplicación

Los CNY70 se polarizarán de forma que por el LED circulen 15mA y que la salida del fotodetector sean niveles TTL válidos y que cumpla con lo que se dice en el enunciado (que den un 0 para negro y un 1 para blanco). Para ello se tendrá en cuenta que para negro (no recibe luz rebotada) el transistor no conduce y para blanco (recibe el 90% de la luz rebotada) conduce según las curvas anteriores.

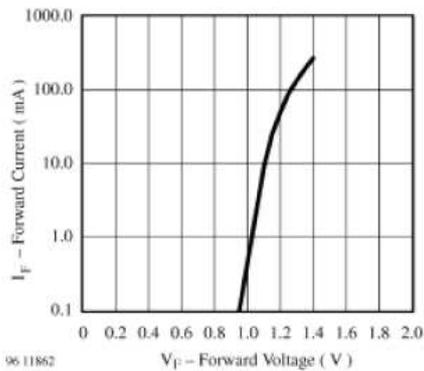


Figura 4.- Corriente directa vs. Tensión directa

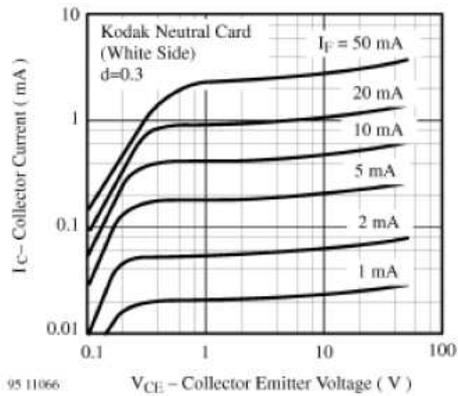


Figura 7.- Corriente de colector vs. Tensión Colector Emisor

Los motores van a requerir hasta 250mA por cada una de sus líneas durante su funcionamiento (cuando

1. Realizar el esquema del circuito, justificando el montaje de los distintos componentes.
2. Realizar el programa de control en pseudocódigo o diagrama de flujo
3. Codificar el programa en ensamblador.