

En este documento se propone una práctica P4 alternativa que hace uso intensivo de las descripciones circuitales ABEL.

El diseño final será funcionalmente equivalente al de la práctica P4 original, pero su funcionamiento interno será sensiblemente distinto.

1. DIAGRAMA DE BLOQUES

La figura 1 muestra el diagrama de bloques del sistema emisor IR propuesto.

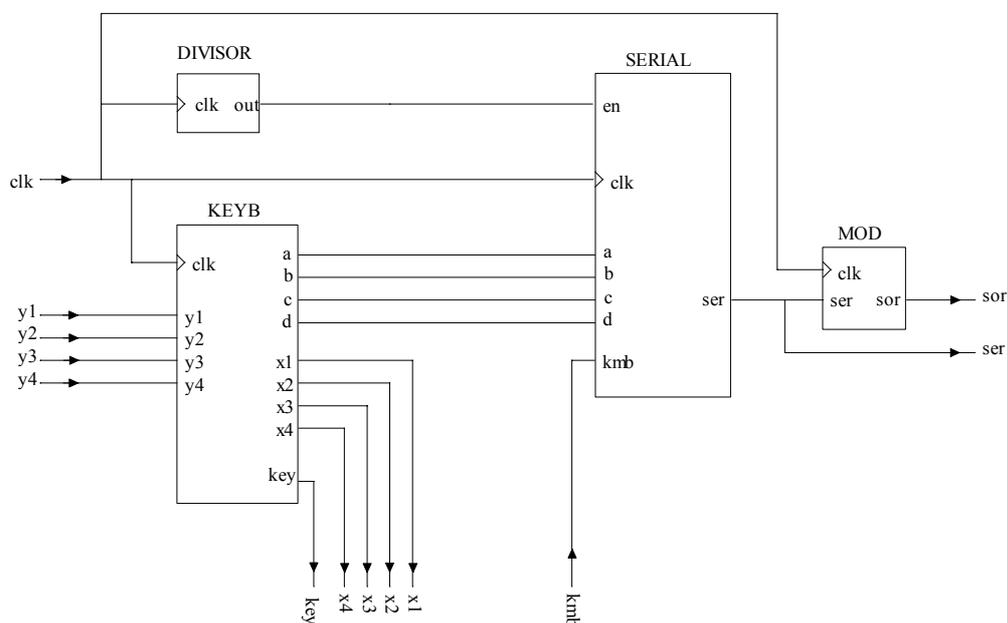


figura 1: Diagrama de bloques

Como puede observarse, el diagrama de bloques del sistema, es el mismo que se describe en la práctica P4. La diferencia estará en la manera de diseñar cada uno de los bloques. Dado que la práctica 4 ya describe suficientemente la funcionalidad del sistema y de cada uno de los bloques, únicamente se describirán los aspectos que los hace distintos internamente en esta versión alternativa de la práctica.

2. MODULADOR

El bloque modulador **MOD** será exactamente el mismo que se describe en la práctica 4, por lo que no hay ninguna diferencia en esta versión alternativa de la práctica.

3. DIVISOR

Las especificaciones del divisor son las mismas que en la práctica 4 original. El divisor de frecuencia deberá dar una señal alta de salida **out** únicamente en uno de cada 64 ciclos de reloj.

Para crear este bloque emplearemos una única descripción ABEL en lugar de encadenar dos divisores 1:8.

4. DECODIFICADOR DE TECLADO

La funcionalidad del decodificador de teclado KEYB será la misma que la descrita en el bloque **conversor de 16 teclas a binario** descrito en la práctica 4. Su construcción interna, no obstante, será distinta. La figura 2 muestra su estructura interna.

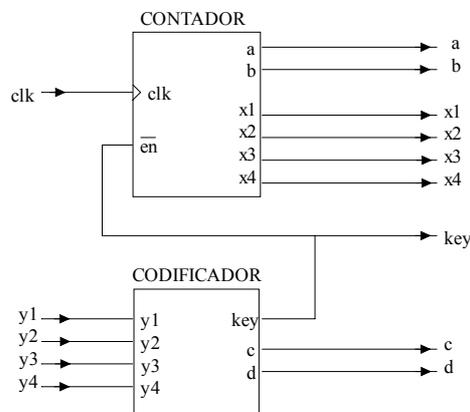


figura 2: Decodificador de teclado

Como puede observarse, el decodificador de teclado tiene únicamente dos bloques. Un contador y un codificador, ambos se generarán empleando descripciones ABEL.

El contador es un contador de 4 estados que, a cada ciclo de reloj, y si la señal de habilitación en (activa baja) lo permite, avanzará un estado. En situación de reset se iniciará en el estado 0 y cada paso avanzará un estado hasta el estado 3. Después del estado 3, el contador volverá al estado 0.

La siguiente tabla muestra el comportamiento del contador en función de las señales de entrada.

clk	en	Acción
	1	Mantener estado
	0	Avanzar estado

El valor de las salidas en función del estado se muestra a continuación:

Estado	b	a	x1	x2	x3	x4
0	0	0	0	1	1	1
1	0	1	1	0	1	1
2	1	0	1	1	0	1
3	1	1	1	1	1	0

El segundo bloque del decodificador de teclado es un codificador que genera las señales key, c y d en función de las señales de entrada y1 a y4. Su funcionalidad es exactamente la misma que la del bloque descrito en la práctica 4.

Su descripción la haremos usando una tabla de la verdad *"truth_table"* en lenguaje ABEL.

5. CONVERTOR PARALELO A SERIE

El último quinto del sistema es el convertor paralelo a serie. En el diseño original de la práctica 4, el convertor serie-paralelo se realizaba empleando un divisor por 10 y un registro de desplazamiento. A parte de la complejidad de su descripción en forma de logigrama, ese diseño tiene el efecto indeseado de generar una señal de salida espuria mientras se vacía el registro de desplazamiento durante el reset del sistema.

La propuesta de esta versión alternativa es realizar todo el convertor paralelo a serie en un único bloque descrito en lenguaje ABEL.

Este bloque contará con las entradas: **clk**, **en**, **kmb**, **a**, **b**, **c**, **d**. Como única salida tendrá la señal **ser**.

El bloque funcionará como una máquina de 10 estados. En el primer estado "0", que es el que se establece después de un reset, la salida se mantendrá a "1" que es el estado normal de reposo de nuestra señal serie.

Si, estando en el estado de reposo "0", las señales **kmb** y **en** se activan al mismo tiempo, se pasará al estado "1".

En el estado "1" se transmitirá el bit de start "0" y se saltará incondicionalmente al estado "2" caso de que se active **en**.

A continuación del bit de start le seguirá el bit A, a este el B, y así sucesivamente hasta el bit de mayor peso D. Le seguirán 3 bits "1" hasta completar los 7 bits de la transmisión serie. Para acabar se enviará un bit de stop "1".

En total se transmitirán 9 bits: Un bit de start, 7 bits de datos y un bit de stop.

Al final del bit de stop se pasará al estado "0" de reset.

Como puede verse, el paso de un estado a otro es incondicional siempre que se de la señal de habilitación en. Eso significa que, una vez empezada a transmitir una palabra, no parará de transmitirse hasta se que se acabe por completo.

Eso significa que si se suelta una tecla, la transmisión continuará pese a ello, por lo que podría darse una transmisión incorrecta caso de que los datos A...D cambien mientras se realiza la transmisión. Es por ello que los valores de A...D se deberán registrar en el momento de iniciarse la transmisión, esto es, cuando se entra en el estado "1" del bit de start.

6. CONJUNTO

Concluido el desarrollo de cada uno de los bloques sólo resta juntarlos entre sí usando las pautas de interconexión descritas en la figura 1.

A continuación se han de crear los pines de conectividad al exterior tal y como se describe en la práctica 4 original y se ha de generar el fichero JEDEC de grabación de PLDs.