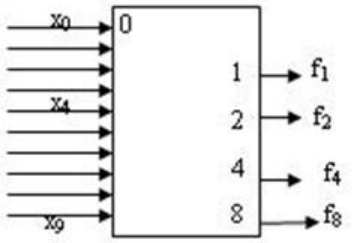
6. Codificadores

Un codificador realiza la función inversa al decodificador, es decir, al activarse una de las entradas, en la salida aparece la combinación binaria correspondiente al número decimal activado entre las entradas.

6.1. Diseño

Codificador sin prioridad. Ejemplo Decimal-BCD

Se denomina codificador sin prioridad a aquellos codificadores en los que no pueden activarse simultáneamente más de una entrada. Supongamos un decodificador de 10 entradas. En ese caso, las únicas posibles combinaciones de entrada serán las representadas en la tabla de la Figura 105. La obtención de las funciones lógicas de salida se obtiene a partir de un análisis de la descripción del funcionamiento del circuito ya que, resulta imposible realizar la tabla de verdad puesto que implicaría 2^{10} combinaciones de entrada. Igualmente, sería imposible realizar la simplificación mediante mapas de Karnaught. Entonces, a partir de la tabla de funcionamiento de la Figura 105 puede concluirse que la salida f_8 se activará si la entrada 8 o la 9 se activan; La salida f_4 se activará si la entrada 4 o la 5 o la 6 o la 7 están activadas; etc..Las ecuaciones lógicas correspondientes a esas descripciones se han representado en la Figura 106 junto con el circuito lógico.



x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	f_8	f_4	f_2	f_1
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

Figura 105

$$f_8 = x_8 + x_9$$
$$f_4 = x_4 + x_5 + x_6 + x_7$$
$$f_2 = x_2 + x_3 + x_6 + x_7$$
$$f_1 = x_1 + x_3 + x_5 + x_7 + x_9$$

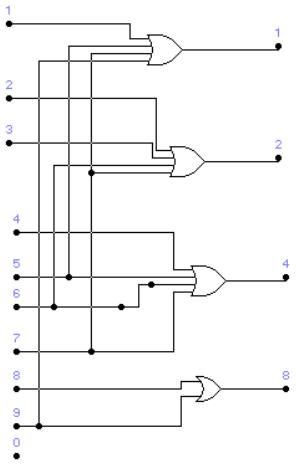


Figura 106

Del diseño presentado se puede extraer que si se activan dos o más entradas simultaneamente, la respuesta en las salidas no tienen significado alguno. Por ello, es necesario diseñar lo que se denomina codificador con prioridad.

Codificador con prioridad. Ejemplo Decimal-BCD

En un codificador con prioridad, con la activación simultanea de varias entradas, en la salida aparecerá el código de la entrada de mayor prioridad (normalmente entrada de peso más significativo), solucionando así el problema planteado en el apartado anterior.
A la hora de describir el funcionamiento mediante la Tabla de Verdad, deben incluirse todas las posibles combinaciones de las variables de entrada. En este caso 2^{10} . Por ello, las funciones lógicas también habrá que obtenerlas a partir de un análisis de la descripción del funcionamiento del circuito. Para facilitar el análisis en la tabla de la izquierda se han representado algunas de las posibles combinaciones de entrada junto con la salida correspondiente. En la tabla de la derecha se representan las únicas posibles combinaciones de salida para todas las combinaciones de entrada.

9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**f8**	**f4**	**f2**	**f1**
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	1	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	1	0	0	1	0
0	0	0	0	0	0	0	1	1	0	0	0	1	0
0	0	0	0	0	0	0	1	1	1	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	0	1	0	0	1	1
0	0	0	0	0	0	1	0	1	0	0	0	1	1
0	0	0	0	0	0	1	0	1	1	0	0	1	1
0	0	0	0	0	0	1	1	0	0	0	0	1	1

f8	**f4**	**f2**	**f1**
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

Tabla 29

La salida f_1 se activa cuando es prioritaria la entrada 1, 3, 5, 7 o 9. Sin embargo, también debe cumplirse que siendo prioritaria la entrada 1, la 2,4,6 y 8 deben ser cero. Por lo tanto, la función que describe dicho comportamiento sería:

$$f_1 = \left(1.\overline{2}.\overline{4}.\overline{6}.\overline{8}\right) + \left(3.\overline{4}.\overline{6}.\overline{8}\right) + \left(5.\overline{6}.\overline{8}\right) + \left(7.\overline{8}\right) + 9$$

Siguiendo la misma lógica, obtenemos las expresiones para las salidas restantes.

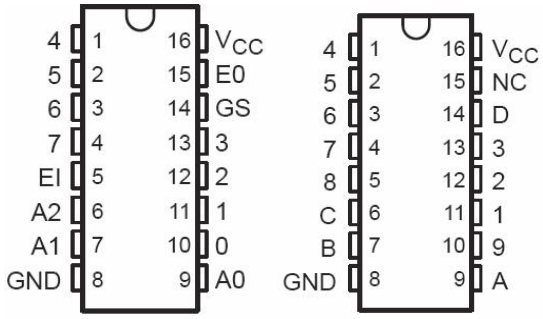
$$f_2 = \left(2.\overline{4}.\overline{5}.\overline{8}.\overline{9}\right) + \left(3.\overline{4}.\overline{5}.\overline{8}.\overline{9}\right) + \left(6.\overline{8}.\overline{9}\right) + \left(7.\overline{8}.\overline{9}\right)$$

$$f_4 = \left(4.\overline{8}.\overline{9}\right) + \left(5.\overline{8}.\overline{9}\right) + \left(6.\overline{8}.\overline{9}\right) + \left(7.\overline{8}.\overline{9}\right)$$

$$f_8 = 8 + 9$$

6.2. Circuitos comerciales: 74148 y 74147

Estos dos codificadores (encapsulado en la Figura 107) tienen tanto las entradas como las salidas activas a nivel bajo. El 74148, dispone además de una entrada de habilitación (EI) y dos salidas cuya funcionalidad puede obtenerse a partir de la Tabla de Funcioanmieto en la Hoja de Caracteristicas correspondiente (Figura 108).



74148 74147

Figura 107

INPUTS									OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

Figura 108

6.3. Aplicación

El ejemplo que se ilustra en la Figura 109 se corresponde con un codificador de teclado. Las teclas se representan mediante diez pulsadores, conectados cada uno de ellos a una resistencia con conexión a la alimentación. Así, si el pulsador está sin pulsar se asegura un estado de nivel alto en las entradas, es decir, desactivada. Al pulsar, la línea se conecta a tierra y entonces se aplica un nivel bajo en la entrada del codificador. Si no se pulsa ninguna tecla, la salida será cero. El codificador utilizado es un codificador con prioridad.

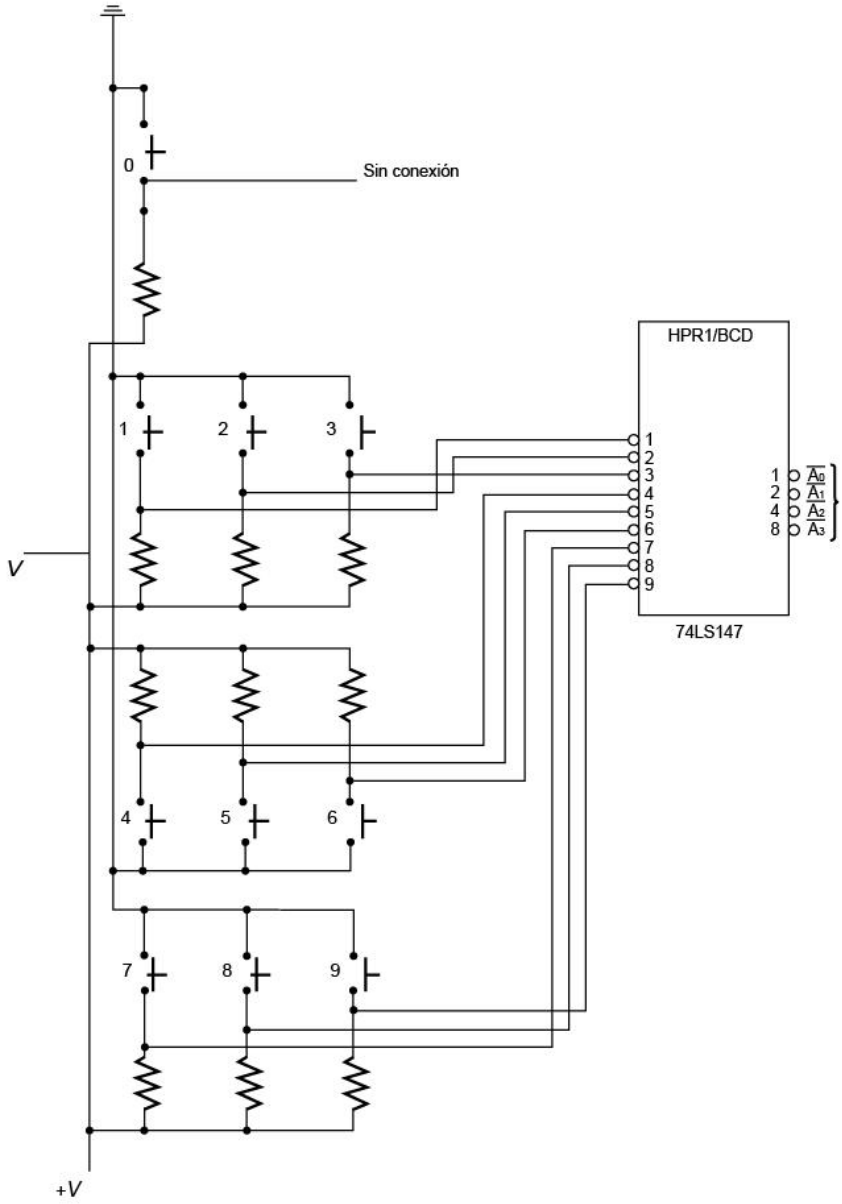


Figura 109

6.4. Descripción VHDL: Codificador 8/3

```
library ieee;
use ieee.std_logic_1164.all;
use work.std_arith.all;
entity coder is port(
dentro: in  std_logic_vector(7 downto 0);
fuera : out std_logic_vector(2 downto 0);
eo    : out bit
);
end coder;

1   architecture archicoder of coder is
2   begin
3   process (dentro)
4   begin
5   if std_match(dentro,"1-------") then fuera<="111";
6   eo<='0'; end if;
7   if std_match(dentro,"01------") then fuera<="110";
8   eo<='0'; end if;
9   if std_match(dentro,"001-----") then fuera<="101";
10  eo<='0'; end if;
11  if std_match(dentro,"0001----") then fuera<="100";
12  eo<='0'; end if;
13  if std_match(dentro,"00001---") then fuera<="011";
14  eo<='0'; end if;
15  if std_match(dentro,"000001--") then fuera<="010";
    eo<='0'; end if;
    if std_match(dentro,"0000001-") then fuera<="001";
    eo<='0'; end if;
    if std_match(dentro,"00000001") then fuera<="000";
    eo<='0'; end if;
    if std_match(dentro,"00000000") then fuera<="000";
    eo<='1'; end if;
    end process;
    end coder;
```