## 0.1 Codificador y Decodificador

Es el proceso de asignar a cada entrada una combinación única de bits.

### 0.1.1 Encoder

Son circuitos combinacionales con $2^n$ entradas máximas y $n$ salidas, en donde las filas de las entradas van a tener **solo un dato que cambia** y en la salida aparece un código asignado a esas entradas.

#### Procedimiento

1. Determine cual es el único valor que cambia en la entrada.
2. Las salidas se leen por columnas tomando cada dato como un término completo.

- **Caso 1: En cada fila hay un dato que cambia.** En un teclado se utilizan teclas del 0 al 3 y se requiere que en las salidas entregue los números codificados en binario en las entradas. La

| IN | | | | OUT | |
|---|---|---|---|---|---|
| $T_0$ | $T_1$ | $T_2$ | $T_3$ | $B_0$ | $B_1$ |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |

Cuadro 1: Tabla de verdad codificador 3 botones.

realización de esta tabla cumple con: $2^2 = 4$ entradas y 2 salidas. El dato que cambia es la entrada 1, si lo escribimos como POS y SOP:

$$B_1 = T_1 + T_3$$
$$B_2 = T_2 + T_3$$
$$B_1 = \overline{T_0} \cdot \overline{T_2}$$
$$B_2 = \overline{T_0} \cdot \overline{T_1}$$

- **Caso 2: Hay una fila donde ningún dato cambia.**

| IN | | | OUT | |
|---|---|---|---|---|
| $D_0$ | $D_1$ | $D_2$ | B | A |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Cuadro 2: Codificador: una fila no cambia.

$$A = \overline{D_1} + (D_0 \cdot D_1 \cdot D_2)$$
$$B = \overline{D_2} + (D_0 \cdot D_1 \cdot D_2)$$
$$A = D_0 \cdot D_2$$
$$B = D_0 \cdot D_1$$

| IN | | | OUT | |
|---|---|---|---|---|
| $D_0$ | $D_1$ | $D_2$ | B | A |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

- **Hay una fila donde todos los datos cambian.**

$$A = \overline{D_1} + (\overline{D_0} \cdot \overline{D_1} \cdot \overline{D_2})$$
$$B = \overline{D_1} + \overline{D_2}$$
$$A = D_0 \cdot D_2$$
$$B = D_0 \cdot (D_0 + D_1 + D_2)$$

Tipos:

- **Sin prioridad**: Solamente una entrada puede ser activada en cada instante.

| IN | | | | | | | | OUT | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $A_0$ | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $Y_2$ | $Y_1$ | $Y_0$ |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

$$Y_0 = \overline{A_1} + \overline{A_3} + \overline{A_5} + \overline{A_7} \therefore Y_0 = \overline{A_1 \times A_3 \times A_5 \times A_7}$$
$$Y_1 = \overline{A_2} + \overline{A_3} + \overline{A_6} + \overline{A_7} \therefore Y_1 = \overline{A_2 \times A_3 \times A_6 \times A_7}$$
$$Y_2 = \overline{A_4} + \overline{A_5} + \overline{A_6} + \overline{A_7} \therefore Y_0 = \overline{A_4 \times A_5 \times A_6 \times A_7}$$

  Las entradas se activan con 0 lógico.
- **Con prioridad**: Codifica la entrada activa de mayor valor decimal son tener en cuenta los demás. Un ejemplo con prioridad es el IC74147, donde todas las entradas se activan con 0 lógico y las todas las salidas están negadas, es decir, se activan con 0 lógico.

### 0.1.2 Decoder

Al contrario del encoder, el decoder es lo contrario: de $n$ entradas se puede manipular $2^n$ entradas. El número en binario se escribe empezando por el A1 (que es más significativo). Los decodificadores tienen su propia tabla de la verdad que representa los estado posibles de entrada y los respectivos valores de las salidas para cada uno de esos estados.
De esta tabla de la verdad podemos sacar el circuito lógico combinacional con puertas lógicas, circuito lógico para construir nuestro decodificador.

**Ejercicio 0.1** Diseñe 2 circuitos que al activarse cada uno de los pulsadores en la salida se visualicen los siguientes números en binario:

1. Pulsador 1: 14
2. Pulsador 2: 50
3. Pulsador 3: 23
4. Pulsador 4: 72

**Solución:**

Primero, vamos a separar decenas y unidades de cada número y diseñar una tabla de verdad para cada uno. Solo poseemos 4 pulsadores, por lo tanto, vamos a necesitar 3 salidas puesto que los números de las decenas(1, 5, 2, 7) expresadas en binario necesitan 3 dígitos; entonces para n=3 salidas nos podemos permitir $2^3 = 8$ entradas, pero solo haremos uso de 4 entradas. Nuestro circuito constará de dos sub-circuitos: uno se encargará de mostrar los dígitos de las decenas y el otro mostrará los dígitos de las unidades. Desarrollamos la tabla de verdad para las decenas.

| INPUT | | | | OUTPUT | | | |
|---|---|---|---|---|---|---|---|
| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $D_2$ | $D_1$ | $D_0$ | $N_0$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 5 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 7 |

Una vez que hayamos desarrollado nuestra tabla de verdad las simplificamos usando POS o SOP, es preciso decir que ambos son posibles pero como hablamos de **SIMPLIFICAR** usaremos el que menos términos genere, en otras palabras: la salida $D_0$ posee 3 salidas en 1(SOP), esa misma salida posee 1 salida en 0(POS), entonces nos combiene POS pues solo será un término comprado a los 3 que nos generaría la otra forma. Aplicando POS a la tabla de las decenas:

$$D_0 = \overline{P_3}$$
$$D_1 = \overline{P_1} \times \overline{P_2}$$
$$D_2 = \overline{P_1} \times \overline{P_3}$$

Lo mismo haremos con las unidades, sin embargo en esta tabla notamos que nos conviene expresarlo como SOP pues en cada salida hay menos salidas 1 que salidas 0, por lo tanto, simplificando mediante SOP la tabla de las unidades:

| INPUT | | | | OUTPUT | | | |
|---|---|---|---|---|---|---|---|
| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $D_2$ | $D_1$ | $D_0$ | $N_0$ |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 3 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 2 |

$$U_0 = P_3$$
$$U_1 = P_3 + P_4$$
$$U_2 = P_1$$

Ya tenemos las expresiones, antes de realizar el circuito esta de más aclarar que el bit más significativo es en el caso de las decenas $D_2$ y en el caso de las unidades es $U_2$. Tenemos las expresiones booleanas de cada tabla así que solo queda realizar su circuito: