



## Informe Tarea 1

### Arquitectura y Organización de Computadores

#### Codificación por '2'

Para la realización del circuito se comenzó por hacer la codificación en orden correspondiente de los caracteres hexadecimales, para así poder seguir trabajando en base a ésta, en ella se dividió cada carácter hexadecimal en sus dígitos  $D_0$  a  $D_3$  en binario y se codificó cada carácter según la clave, en el orden de ésta:

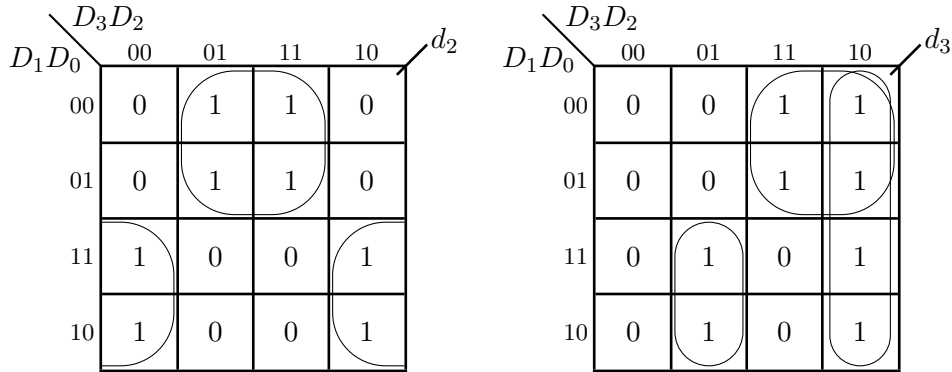
Tabla de Codificación por '2'

Caracter	Término	$D_3$	$D_2$	$D_1$	$D_0$	$d_3$	$d_2$	$d_1$	$d_0$
0	0	0	0	0	0	0	0	1	0
1	1	0	0	0	1	0	0	1	1
2	2	0	0	1	0	0	1	0	0
3	3	0	0	1	1	0	1	0	1
4	4	0	1	0	0	0	1	1	0
5	5	0	1	0	1	0	1	1	1
6	6	0	1	1	0	1	0	0	0
7	7	0	1	1	1	1	0	0	1
8	8	1	0	0	0	1	0	1	0
9	9	1	0	0	1	1	0	1	1
A	10	1	0	1	0	1	1	0	0
B	11	1	0	1	1	1	1	0	1
C	12	1	1	0	0	1	1	1	0
D	13	1	1	0	1	1	1	1	1
E	14	1	1	1	0	0	0	0	0
F	15	1	1	1	1	0	0	0	1

En primer lugar se ve inmediatamente que el primer dígito  $D_0$  no varía, por lo que la función sale inmediatamente ( $d_0 = D_0$ ). De forma análoga sucede para  $D_1$ , pero con la negación:  $d_1 = \overline{D_1}$ . Por lo que la salida para ambos dígitos estaría lista y faltaría trabajar con  $D_2$  y  $D_3$ . Si nos fijamos, no es tan simple visualizar el circuito para  $D_2$ , por lo que recurrimos al mapa de Karnaugh para obtener  $d_2$  y  $d_3$ .



### Mapas de Karnaugh para $d_2$ y $d_3$



Entonces las funciones son respectivamente:  $d_2 = D_2 \overline{D_1} + \overline{D_2} D_1$  y  $d_3 = D_3 \overline{D_1} + \overline{D_3} D_2 D_1 + D_3 \overline{D_2}$ .

### Codificación por '4'

Luego seguimos con la tabla para la codificación para la clave '4'.

Tabla de Codificación por '4'

Caracter	Término	$D_3$	$D_2$	$D_1$	$D_0$	$d_3$	$d_2$	$d_1$	$d_0$
0	0	0	0	0	0	0	1	0	0
1	1	0	0	0	1	0	1	0	1
2	2	0	0	1	0	0	1	1	0
3	3	0	0	1	1	0	1	1	1
4	4	0	1	0	0	1	0	0	0
5	5	0	1	0	1	1	0	0	1
6	6	0	1	1	0	1	0	1	0
7	7	0	1	1	1	1	0	1	1
8	8	1	0	0	0	1	1	0	0
9	9	1	0	0	1	1	1	0	1
A	10	1	0	1	0	1	1	1	0
B	11	1	0	1	1	1	1	1	1
C	12	1	1	0	0	0	0	0	0
D	13	1	1	0	1	0	0	0	1
E	14	1	1	1	0	0	0	1	0
F	15	1	1	1	1	0	0	1	1

Concluimos inmediatamente:

- $d_0 = D_0$



- $d_1 = D_1$
- $d_2 = \overline{D_2}$

Y para  $d_3$  realizamos el mapa Karnaugh.

**Mapa de Karnaugh para  $d_3$**

		$D_3D_2$				
		00	01	11	10	
$D_1D_0$	00	0	1	0	1	$d_3$
	01	0	1	0	1	
	11	0	1	0	1	
	10	0	1	0	1	

Entonces obtenemos la última función:  $d_3 = \overline{D_3}D_2 + D_3\overline{D_2}$ .

## Codificación por '5'

Finalmente, para terminar la parte de codificación hacemos el mismo proceso para cuando la clave sea '5'.

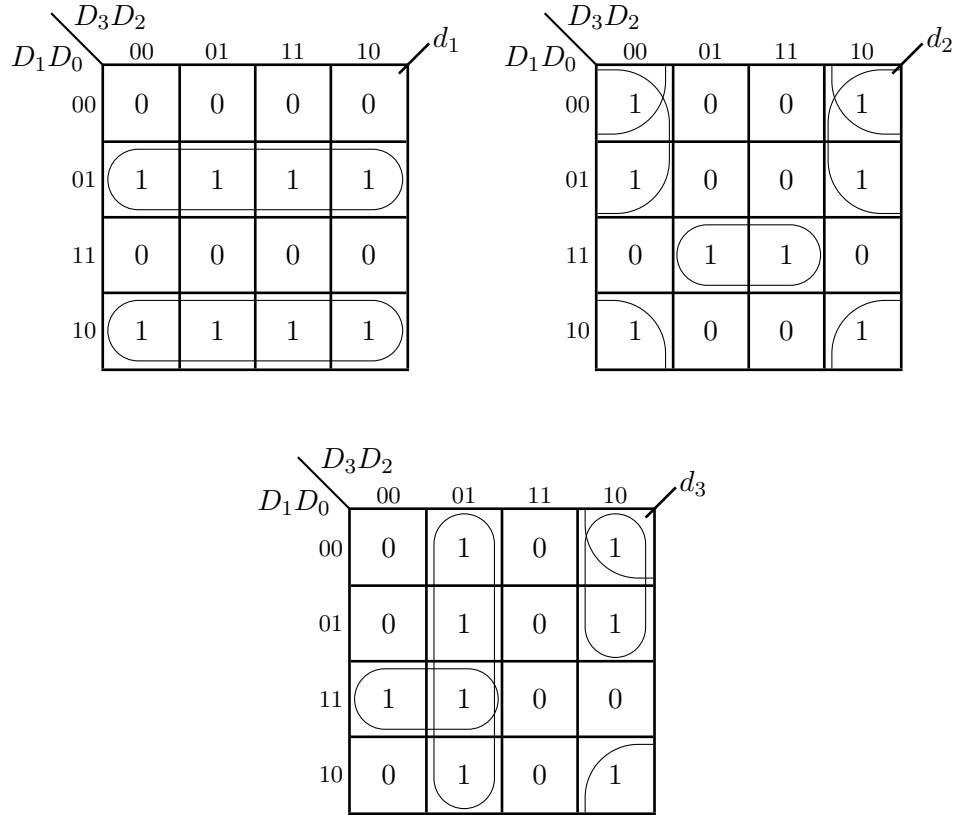
**Tabla de Codificación por '5'**

Caracter	Término	$D_3$	$D_2$	$D_1$	$D_0$	$d_3$	$d_2$	$d_1$	$d_0$
0	0	0	0	0	0	0	1	0	1
1	1	0	0	0	1	0	1	1	0
2	2	0	0	1	0	0	1	1	1
3	3	0	0	1	1	1	0	0	0
4	4	0	1	0	0	1	0	0	1
5	5	0	1	0	1	1	0	1	0
6	6	0	1	1	0	1	0	1	1
7	7	0	1	1	1	1	1	0	0
8	8	1	0	0	0	1	1	0	1
9	9	1	0	0	1	1	1	1	0
A	10	1	0	1	0	1	1	1	1
B	11	1	0	1	1	0	0	0	0
C	12	1	1	0	0	0	0	0	1
D	13	1	1	0	1	0	0	1	0
E	14	1	1	1	0	0	0	1	1
F	15	1	1	1	1	0	1	0	0



Luego  $d_0 = \overline{D_0}$  y usamos Karnaugh para minimizar el resto:

### Mapas de Karnaugh para $d_1$ , $d_2$ y $d_3$



De donde obtenemos las funciones:

- $d_1 = \overline{D_1}D_0 + D_1\overline{D_0}$
- $d_2 = \overline{D_2}\overline{D_1} + D_2D_1D_0 + \overline{D_2}\overline{D_0}$
- $d_3 = \overline{D_3}D_2 + \overline{D_3}D_1D_0 + D_3\overline{D_2}\overline{D_0} + D_3\overline{D_2}D_1$



## Construcción del Circuito

Primero para construir el circuito recapitularemos las funciones que tenemos de forma ordenada:

### Cifrado por '2'

- $d_0 = D_0$
- $d_1 = \overline{D_1}$
- $d_2 = D_2\overline{D_1} + \overline{D_2}D_1$
- $d_3 = D_3\overline{D_1} + \overline{D_3}D_2D_1 + D_3\overline{D_2}$

### Cifrado por '4'

- $d_0 = D_0$
- $d_1 = D_1$
- $d_2 = \overline{D_2}$
- $d_3 = \overline{D_3}D_2 + D_3\overline{D_2}$

### Cifrado por '5'

- $d_0 = \overline{D_0}$
- $d_1 = \overline{D_1}D_0 + D_1\overline{D_0}$
- $d_2 = \overline{D_2}\overline{D_1} + D_2D_1D_0 + \overline{D_2}\overline{D_0}$
- $d_3 = \overline{D_3}D_2 + \overline{D_3}D_1D_0 + D_3\overline{D_2}\overline{D_0} + D_3\overline{D_2}\overline{D_1}$

Notamos que las funciones de la forma:  $\overline{AB} + \overline{A}B$  equivalen simplemente a la representación de un *XOR*.

En segundo lugar, como la clave se distribuye de forma cíclica sobre la palabra y se tienen palabras de cuatro y ocho caracteres, la construcción de los circuitos corresponde a un subcircuito para cada clave (2, 4 y 5) y la repetición de éstos según sea necesario.

Por otro lado como el input y output deben ser una palabra de cuatro y ocho dígitos hexadecimales, se utilizó un PIN de 16 y 32 bits respectivamente para representar dicha palabra en binario, donde cada 4 bits se representa un carácter ordenadamente de izquierda a derecha y de arriba hacia abajo. Luego se utilizó un splitter para distribuir los bits correspondientes a cada carácter sobre los subcircuitos adecuados, de forma ordenada, de tal modo que los caracteres también están distribuidos de izquierda a derecha sobre los subcircuitos. Análogamente se aplica el proceso para el output. Además, hacia las entradas y salidas de los subcircuitos se incluyó un Display de 7 Segmentos, para facilitar la visualización en hexadecimal de cada carácter a codificar o ya codificados.



## Extra

Para ello se toman ambos inputs (la palabra y la clave), de ser necesario se deben transformar a binario, y se hacen pasar por un circuito sumador para que así el resultado sea el caracter codificado, es importante tener en cuenta que el Carry Out es lo que se debe ignorar para simplemente para emular el módulo que se realiza en el cifrado. Finalmente, como el resultado está en binario, bastaría transformar a hexadecimal de vuelta si así se requiere.