

Informe de Laboratorio: El santuario de los circuitos digitales

Vicente Rodríguez Rogers 202273503-1
Nicolás Muñoz 202273641-0

Octubre 2025

1. Introducción

Durante el transcurso del informe se documentará el desarrollo del decodificador de mensajes encriptados para la revelación de los códigos súper secretos que ByteMaster ha interceptado. Para cumplir con este propósito, se realizará un análisis de inteligencia que permitirá entender los códigos y al mismo tiempo diseñar un modelo minimizado para la creación de un circuito encargado de des-encriptar los secretos de la organización criminal.

2. Análisis Teórico

2.1. Análisis de situación

Gracias a lo otorgado por ByteMaster, sabemos que mediante 4 bit, los ladrones han estado comunicándose entre ellos mediante códigos con un display de 7 segmentos, estableciendo 16 símbolos cifrados distintos como se muestra en la siguiente tabla:

Código (4 bits)	Decimal	Símbolo	Significado Criminal
0000	0	0	Operación en espera
0001	1	1	Agente en posición
0010	2	2	Dos vehículos listos
0011	3	3	Tres puntos de acceso
0100	4	4	Cuatro minutos para iniciar
0101	5	5	Cinco millones transferidos
0110	6	6	Seis guardias neutralizados
0111	7	7	Siete pisos del edificio
1000	8	8	Ocho cajas fuertes
1001	9	9	Nueve rehenes asegurados
1010	10	A	Alerta máxima
1011	11	b	Base comprometida
1100	12	C	Código de evacuación
1101	13	d	Destruir evidencia
1110	14	E	Emergencia total
1111	15	F	Fin de la operación

Cuadro 1: Tabla de Símbolos Cifrados y Significados Criminales.

Ademas de esto, para el display de 7 segmentos que representan los símbolos de los criminales, se estableció la siguiente caracterización y codificación:

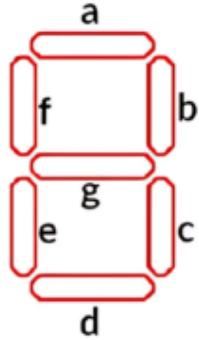


Figura 1: *
Representación de un display
de 7 segmentos

Símbolo	Segmentos Activos	abcdefg
0	a,b,c,d,e,f	1111110
1	b,c	0110000
2	a,b,g,e,d	1101101
3	a,b,g,c,d	1111001
4	f,g,b,c	0110011
5	a,f,g,c,d	1011011
6	a,f,g,e,d,c	1011111
7	a,b,c	1110000
8	a,b,c,d,e,f,g	1111111
9	a,b,c,d,f,g	1111011
A	a,b,c,e,f,g	1110111
b	f,e,d,c,g	0011111
C	a,f,e,d	1001110
d	b,c,d,e,g	0111101
E	a,f,g,e,d	1001111
F	a,f,g,e	1000111

Figura 2: *
Tabla de Segmentos y Códigos Binarios

2.2. Tablas de verdad y mapa de Karnaugh

Para poder fabricar un circuito capaz de des-criptar los códigos, se necesitara poder establecer distintas tablas de verdad que permitan transformar los 4 bits de entrada en los símbolos del display, es por eso que se confecciono la siguiente tabla de verdad.

Entrada (4-bit)				Salida (Segmentos)						
A3	A2	A1	A0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	0	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	1	1	1	1	0	1	1
1	0	1	1	0	0	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1	0
1	1	0	1	0	1	1	1	1	0	1
1	1	1	0	1	0	0	0	1	1	1
1	1	1	1	1	0	0	0	0	1	1

Cuadro 2: Tabla de Verdad del Decodificador

Y ahora se analizará particularmente cada entrada y se buscara minimizar la expresión característica mediante tablas de Karnaugh.

1. Segmento a:

a) Tabla de verdad:

A3	A2	A1	A0	a
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

b) Mapa de Karnaugh:

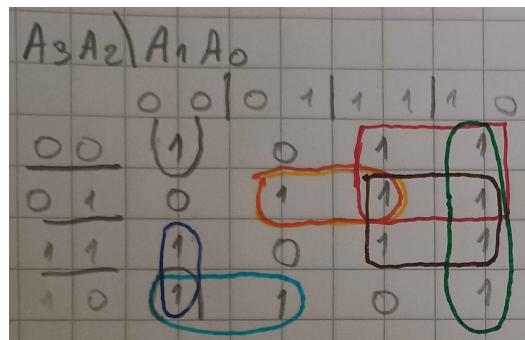


Figura 3: $f(a) = \overline{A_2}\overline{A_1}\overline{A_0} + A_3\overline{A_1}\overline{A_0} + A_3\overline{A_2}\overline{A_1} + \overline{A_3}A_2A_0 + \overline{A_3}A_1 + A_2A_1 + A_1\overline{A_0}$

2. Segmento b:

a) Tabla de verdad:

A3	A2	A1	A0	b
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

b) Mapa de Karnaugh:

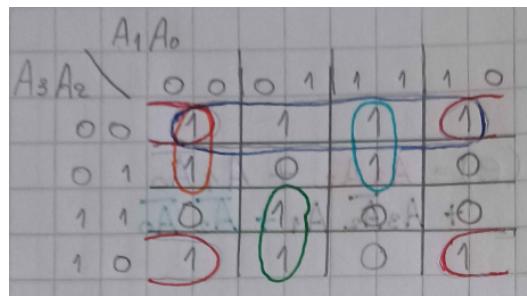


Figura 4: $f(b) = A_3\bar{A}_1A_0 + \bar{A}_3A_1A_0 + \bar{A}_3\bar{A}_2 + \bar{A}_3\bar{A}_1\bar{A}_0 + \bar{A}_2\bar{A}_0$

3. Segmento c:

a) Tabla de verdad:

A3	A2	A1	A0	c
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

b) Mapa de Karnaugh:

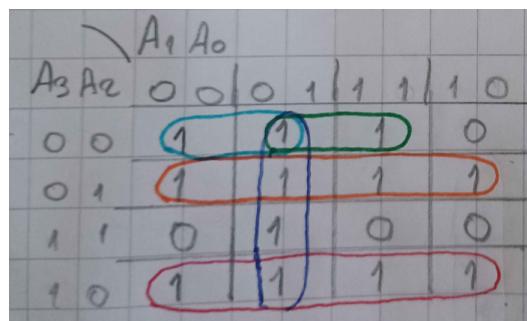


Figura 5: $f(c) = \overline{A}_3\overline{A}_2A_0 + \overline{A}_3\overline{A}_2\overline{A}_1 + \overline{A}_3A_2 + \overline{A}_1A_0 + A_3\overline{A}_2$

4. Segmento d:

a) Tabla de verdad:

A3	A2	A1	A0	d
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

b) Mapa de Karnaugh:

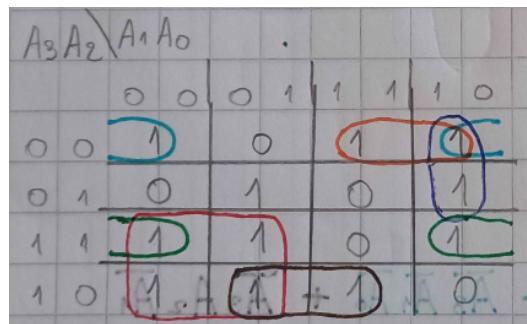


Figura 6: $f(d) = A_3\bar{A}_2A_0 + \bar{A}_3\bar{A}_2A_1 + \bar{A}_3\bar{A}_2\bar{A}_0 + A_3\bar{A}_1 + A_3A_2\bar{A}_0 + \bar{A}_3A_1\bar{A}_0$

5. Segmento e:

a) Tabla de verdad:

A3	A2	A1	A0	e
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

b) Mapa de Karnaugh:

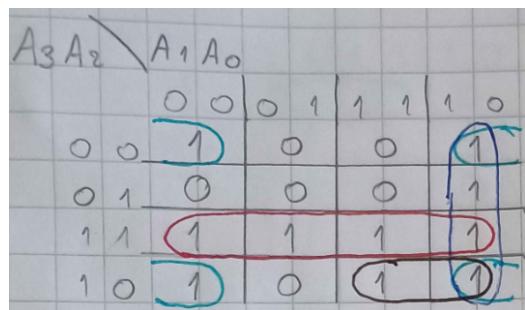


Figura 7: $f(e) = A_3A_2 + A_1\bar{A}_0 + A_3\bar{A}_2A_1 + \bar{A}_2\bar{A}_0$

6. Segmento f:

a) Tabla de verdad:

A3	A2	A1	A0	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

b) Mapa de Karnaugh:

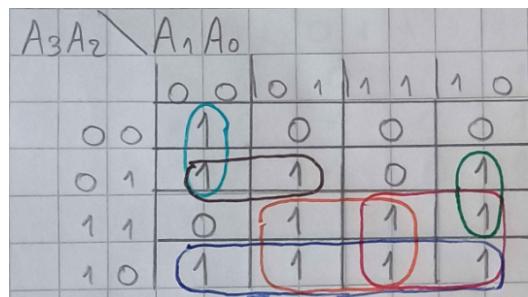


Figura 8: $f(f) = A_3A_0 + A_3A_1 + A_3\bar{A}_2 + A_2A_1\bar{A}_0 + \bar{A}_3\bar{A}_1\bar{A}_0 + \bar{A}_3A_2\bar{A}_1$

7. Segmento g:

a) Tabla de verdad:

A3	A2	A1	A0	g
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

b) Mapa de Karnaugh:

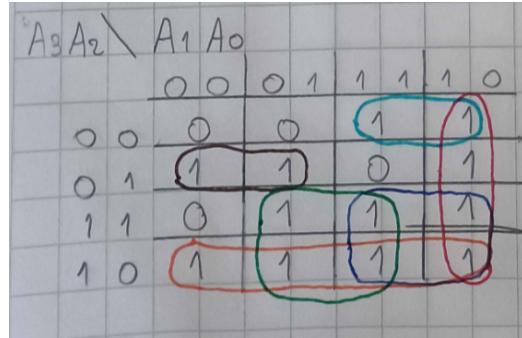


Figura 9: $f(g) = \overline{A}_3\overline{A}_2A_1 + A_1\overline{A}_0 + A_3A_1 + A_3A_0 + A_3\overline{A}_2 + \overline{A}_3A_2\overline{A}_1$

Con esto, ya tendríamos una expresión para cada uno de los segmentos del display, expresiones que gracias a haberlas obtenido desde un mapa de Karnaugh se encontraría optimamente minimizadas al menos en términos de álgebra booleana clásica. En resumen las expresiones de cada segmento serían:

- $f(a) = \overline{A}_2\overline{A}_1\overline{A}_0 + A_3\overline{A}_1\overline{A}_0 + A_3\overline{A}_2\overline{A}_1 + \overline{A}_3A_2A_0 + \overline{A}_3A_1 + A_2A_1 + A_1\overline{A}_0$

Nota: Luego de analizar las expresiones, se notó que esta podría haber sido minimizada más en el mapa de Karnaugh tomando mejor las 4

esquinas y así ahorrar una expresión. generando lo siguiente:

$$f'(a) = A'_2 A'_0 + A'_3 A_1 + A'_3 A_2 A_0 + A_2 A_1 + A_3 A'_2 A'_1 + A_3 A'_0$$

- $f(b) = A_3 \bar{A}_1 A_0 + \bar{A}_3 A_1 A_0 + \bar{A}_3 \bar{A}_2 + \bar{A}_3 \bar{A}_1 \bar{A}_0 + \bar{A}_2 \bar{A}_0$
- $f(c) = \bar{A}_3 \bar{A}_2 A_0 + \bar{A}_3 \bar{A}_2 \bar{A}_1 + \bar{A}_3 A_2 + \bar{A}_1 A_0 + A_3 \bar{A}_2$
- $f(d) = A_3 \bar{A}_2 A_0 + \bar{A}_3 \bar{A}_2 A_1 + \bar{A}_3 \bar{A}_2 \bar{A}_0 + A_3 \bar{A}_1 + A_3 A_2 \bar{A}_0 + \bar{A}_3 A_1 \bar{A}_0$
- $f(e) = A_3 A_2 + A_1 \bar{A}_0 + A_3 \bar{A}_2 A_1 + \bar{A}_2 \bar{A}_0$
- $f(f) = A_3 A_0 + A_3 A_1 + A_3 \bar{A}_2 + A_2 A_1 \bar{A}_0 + \bar{A}_3 \bar{A}_1 \bar{A}_0 + \bar{A}_3 A_2 \bar{A}_1$
- $f(g) = \bar{A}_3 \bar{A}_2 A_1 + A_1 \bar{A}_0 + A_3 A_1 + A_3 A_0 + A_3 \bar{A}_2 + \bar{A}_3 A_2 \bar{A}_1$

3. Implementación

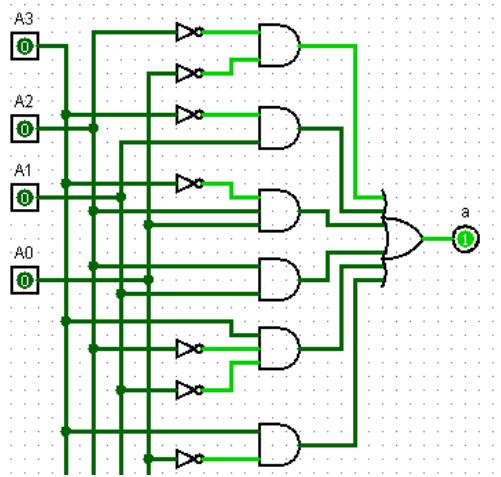
Ahora que tenemos las expresiones de cada segmento, estamos capacitados para fabricar el circuito para el display decodificador. Para esto haremos uso del software **Logism** que nos facilitara tanto la creación gráfica como la del modelo del circuito.

El circuito consta de 7 secciones, cada una dedicada a el segmento del display específico y otorgando una salida correspondiente al estado de encendido del segmento, todas ellas comparten las 4 entradas booleanas donde se ingresaran los códigos de los criminales. estos fueron creados a partir de compuertas **OR**, **AND** y **NOT** por sus cualidades directas y completas para la álgebra booleana.

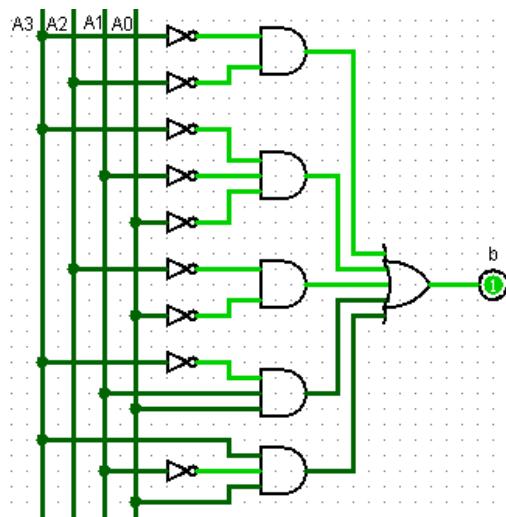
3.1. Circuito segmentado

Para mayor comprensión del circuito, se muestra a continuación cada segmento por separado:

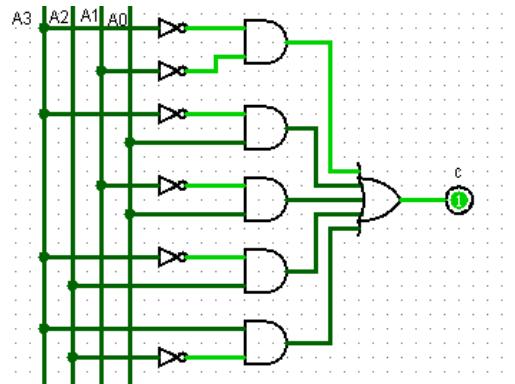
- **Segmento a:**



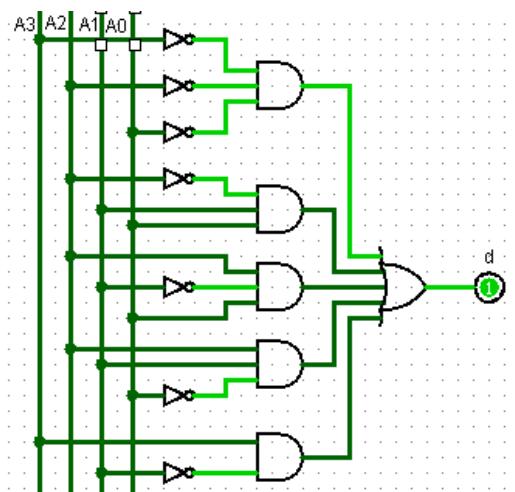
- Segmento b:



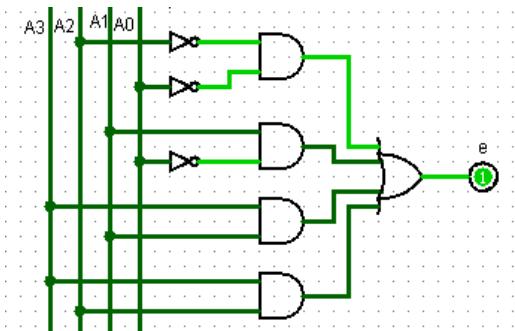
- Segmento c:



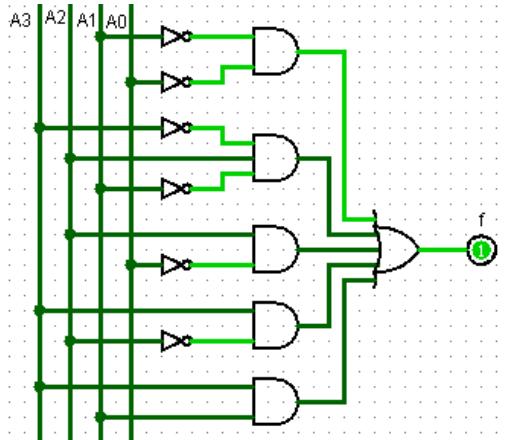
■ Segmento d:



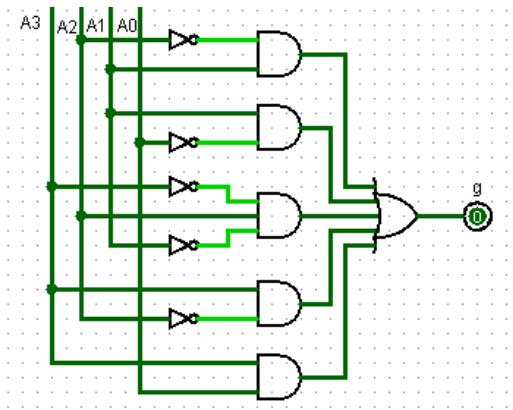
■ Segmento e:



- Segmento f:



- Segmento g:



3.2. Arquitectura, Explicación y Fundamentos

La Arquitectura del decodificador se basa en el diseño de un **circuito combinacional**. Esto debido a que la salida, que son los símbolos entregados por el display, depende **únicamente** de la combinación actual de las 4 entradas. Y como no se requiere **memoria** o un **reloj**, no es necesario utilizar una **lógica secuencial** para el desarrollo del circuito (Flip-Flop, etc.). Por lo tanto, la utilización exclusiva de **compuertas básicas** (AND, OR, NOT) pareció una opción ideal para el desarrollo del sistema.

¿Por qué solo compuertas AND, OR y NOT?

Utilizamos una estrategia de **minimización óptima** para el sistema de cuatro variables que nos garantizaba obtener una expresión mínima de **Suma de Productos (SOP)**, dándonos un **álgebra booleana** sin términos redundantes ordenados por bloques fácilmente integrables al sistema con estos tipos de compuertas. Las expresiones nos otorgaban una **guía directa y óptima** para la creación del circuito, la cual se hubiera vuelto más compleja de utilizar otras compuertas como XOR o NAND.

El **diseño del circuito** fue planeado de manera **segmentada** (modular), y cada segmento se materializó en **tres niveles de compuertas** gracias al formato **SOP**.

- **Primer nivel:** Este nivel se compone de compuertas **NOT**, **invirtiendo** todas las entradas que deban ser negadas (\bar{A}_x) antes de pasar al siguiente nivel.
- **Segundo nivel:** Este nivel se compone de compuertas **AND**, obtenida de las multiplicaciones booleanas que nos otorga el K-map.
- **Tercer nivel:** Por último, este y último nivel toma todos los valores obtenidos de los niveles anteriores y les aplica una compuerta **OR**. Esta es la **compuerta final** de cada segmento.

3.3. Circuito Integrado

Al completar el circuito, lo conectamos a un **display de 7 segmentos**, y se logra comprobar la **funcionalidad del decodificador** al ingresar los códigos de 4 bits y obtener la figura esperada como se muestra en las siguientes figuras.

Figura 10: Cuadrícula de 16 símbolos de 7 segmentos (4x4).

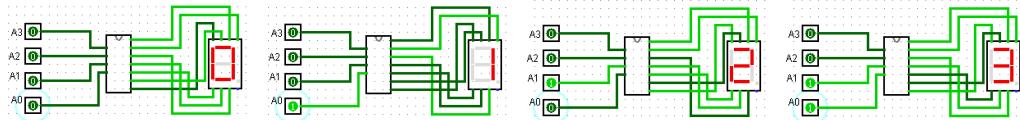


Figura 11: *
0000

Figura 12: *
0001

Figura 13: *
0010

Figura 14: *
0011

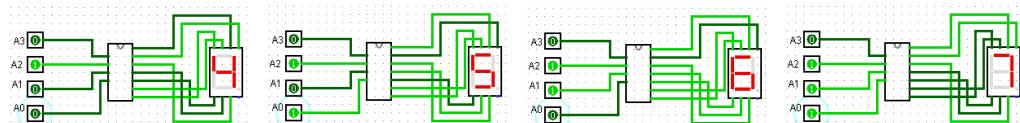


Figura 15: *
0100

Figura 16: *
0101

Figura 17: *
0110

Figura 18: *
0111

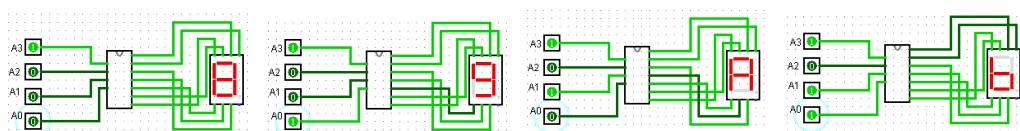


Figura 19: *
1000

Figura 20: *
1001

Figura 21: *
1010 (A)

Figura 22: *
1011 (b)

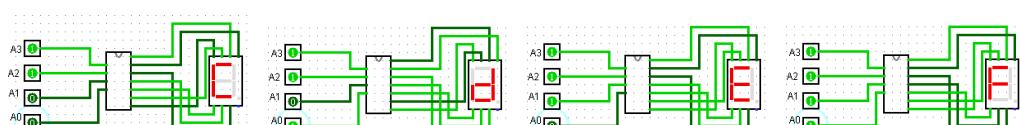


Figura 23: *
1100 (C)

Figura 24: *
1101 (d)

Figura 25: *
1110 (E)

Figura 26: *
1111 (F)

Nota: Para un efecto visual mas claro, se utilizo el circuito en formato modulo de 4 bit de entrada y 7 de salida. el circuito interior de este modulo es exactamente igual que el mostrado con anterioridad.