

Circuitos y Sistemas Digitales

Proyecto final

Clase 16



La excelencia no se improvisa



1. INTRODUCCIÓN

En este capítulo veremos la última fase de nuestro proyecto que Integrará los elementos desarrollados a lo largo del curso; reemplazando con una memoria Ram/Rom la lógica de la máquina secuencial que presenta un mensaje personalizado usando circuitos combinacionales, los cuales fueron implementados con MUX como circuitos de salida múltiple que alimentaba al display de siete segmentos, mientras que el reloj será reemplazado por uno de diseño propio.

En cada uno de los retos se han ido incorporando nuevos elementos que reemplazaban ciertas etapas del circuito, optimizando el diseño y poniendo en práctica los conocimientos adquiridos. En esta última etapa se incorporará la memoria RAM, en la cual se escribirá la tabla de verdad del mensaje individual presentado previamente por cada estudiante y será leída para ser mostrada en un display.

Clase 16: Proyecto final.

Resultado o resultados de aprendizaje que será abordado con el contenido de la clase. **(No agregue los criterios)**

Realizar pruebas y depuración de circuitos digitales, para llegar a la identificación y solución de problemas de manera eficiente.

6. Proyecto final.

El proyecto final del curso contempla el implementar la simulación de una máquina secuencial que muestre un mensaje de 13 a 15 caracteres usando circuitos combinacionales, divisores de frecuencia, un contador de diseño propio y almacene el mensaje en una memoria para presentarlo en un display.

Esta máquina debe operar dentro del entorno de simulación SimulIDE, con el propósito de conocer y entender cómo estos componentes interaccionan para procesar entradas secuenciales y producir las salidas correspondientes, lo que representa una experiencia práctica para adquirir habilidades en la comprensión y diseño de los Circuitos y Sistemas Digitales.

Previamente se había implementado:

- La simulación de un Circuito Combinacional que sirve como interface entre números binarios de 4 dígitos y letras a presentarse en un display de 7 segmentos que presenta un mensaje que contiene entre 13 y 15 caracteres.
- La simulación de una máquina secuencial que cuenta en hexadecimal del cero a la F usando un contador de diseño propio con divisores de frecuencia.

- Se integraron los distintos módulos desarrollados como una máquina secuencial que presenta un mensaje hasta 15 caracteres

En esta etapa, como reemplazo a la lógica binaria construida con MUX, se usará una memoria Ram/Rom para almacenar el mensaje, se reemplazará el reloj por uno diseñado con el circuito integrado 555 y luego se incorporarán los otros módulos para alcanzar el producto final.

Mensaje a presentarse.

Cada estudiante definió su propio mensaje, el cual es único y deberá mantenerlo en esta etapa del proyecto, por otro lado y como ejemplo; el mensaje de referencia que hemos usado como ejemplo en este curso es:

d	E	J	A	-	t	U	-	H	U	E	L	L	A	-	-
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Una vez definido el mensaje, se construyó la tabla de verdad para las combinaciones de la entrada ABCD y la salida para el display de 7 segmentos, lo que permitió diseñar el circuito combinacional, de donde se extrajeron las funciones a, b, c, d, e, f y g que manejan el display de 7 segmentos, para lo cual previamente usamos MUX 4-1 como implementación de la lógica binaria combinacional del circuito.

[En este sitio puede conocer mas detalles del display de 7 segmentos y su uso.](#)

Sin embargo en este punto y para grabar el mensaje en la memoria, será necesario agregar tres columnas nuevas llamadas “HEX”, “DEC” y “ASCII”, tal como se muestra en la Tabla 1.

Dado que ya conocemos el funcionamiento y uso de una memoria; podemos reemplazar los MUX por una memoria, y en ella almacenaremos el mensaje que queremos presentar. Es decir necesitaríamos una memoria con un bus de direcciones de 4 bits para las 16 posiciones y una palabra de tamaño de 7 bits para las 7 salidas a, b, c, d, e, f y g. Dado que comercialmente existen memorias con palabras de 8, 16, 32... bits, usaremos una memoria con una palabra 8 bits, ignorando el más significativo.

Estas tres nuevas columnas contendrán en este orden:

- **HEX:** Contiene el número hexadecimal de la salida de 7 segmentos agregado un cero a la izquierda. Por ejemplo para las entradas ABCD; 0000, las salidas abc-defg son; 011-1101, lo cual puede ser expresado en Hexadecimal como 3D, agregando el cero a la izquierda, como se mencionó previamente.
- **DEC:** Muestra equivalente en decimal del número hexadecimal previamente descrito
- **ASCII:** Presenta el código ASCII correspondiente a la salida descrita.

Estas tres columnas se aprecian a la derecha en la Tabla 1:

Tabla 1

Tabla de verdad ABCD a 7 segmentos con el mensaje "deJA-tU-HUELLA- -"

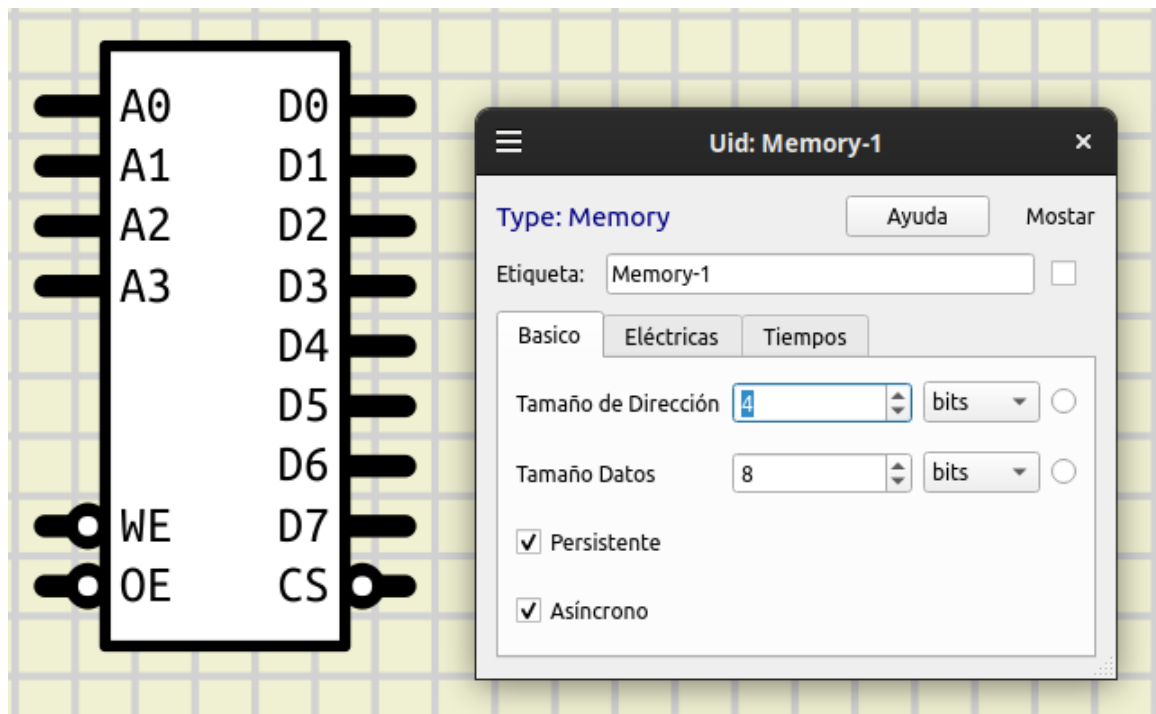
Cel-da	A	B	C	D	Mensaje	a	b	c	d	e	f	g	HE X	DEC	AS-CII
						D 1	D 2	D 3	D 4	D 5	D 6	D 7			
0	0	0	0	0	d	0	1	1	1	1	0	1	3D	61	=
1	0	0	0	1	E	1	0	0	1	1	1	1	4F	79	O
2	0	0	1	0	J	0	1	1	1	0	0	0	38	56	8
3	0	0	1	1	A	1	1	1	0	1	1	1	77	119	w
4	0	1	0	0	-	0	0	0	0	0	0	1	01	1	SOH
5	0	1	0	1	t	0	0	0	1	1	1	1	0F	15	SI
6	0	1	1	0	U	0	1	1	1	1	1	0	3E	62	>
7	0	1	1	1	-	0	0	0	0	0	0	1	01	1	SOH
8	1	0	0	0	H	0	1	1	0	1	1	1	37	55	7
9	1	0	0	1	U	0	1	1	1	1	1	0	3E	62	>
10	1	0	1	0	E	1	0	0	1	1	1	1	4F	79	O
11	1	0	1	1	L	0	0	0	1	1	1	0	0E	14	SO
12	1	1	0	0	L	0	0	0	1	1	1	0	0E	14	SO
13	1	1	0	1	A	1	1	1	0	1	1	1	77	119	w
14	1	1	1	0	-	0	0	0	0	0	0	1	01	1	SOH
15	1	1	1	1	-	0	0	0	0	0	0	1	01	1	SOH

Nota: Elaborado por (Escobar C, 2025)

Para simular la memoria con un bus de direcciones de 4 bits y una palabra de 8 bits en SimulIde (González et al., 2025), se puede agregar una memoria Ram/Rom y luego, sobre ella, con “botón derecho” —> “propiedades” de su menú contextual definirlos parámetros que se muestran en la Figura 1.

Figura 1

Memoria RAM simulada en SimulIDE con 4 bits de direcciones y palabra de 8 bits.



Nota: Elaborado por (Escobar C, 2025)

Una vez que se ha introducido y configurado la memoria Rom/Ram se puede escribir el contenido en las direcciones de memoria, para ello de manera similar; con el botón derecho sobre la memoria seleccione mostrar tabla de memoria, esto nos mostrará la tabla de la Figura 2

Figura 2
Tabla de memoria de la Ram/Rom

Nota: Elaborado por (Escobar C, 2025)

En el lado izquierdo de la tabla de memoria se tiene las direcciones de memoria y su contenido, mientras que en el lado derecho, las direcciones de memoria con el mismo contenido pero expresado en código ASCII

La tabla puede ser llenada en el lado izquierdo, escribiendo el código en decimal, el cual automáticamente, una vez ingresados los dígitos decimales, estos se transformarán en hexadecimal y se mostrará a la derecha el código ASCII equivalente.

En el lado derecho de la tabla también se puede ingresar su contenido en código ASCII, sin embargo debido a que la columna ASCII tiene caracteres no convencionales, se dificulta el ingreso de estos, razón por la cual es mejor realizar el ingreso como se propone previamente.

Así para la dirección de memoria 0000 (Columnas A, B, C, D) el contenido en decimal es 61 (Columna DEC), tal como se muestra en la tabla 2.

Tabla 2Contenido de la dirección de memoria 0000

Cel-da	A	B	C	D	Mensaje	a	b	c	d	e	f	g	HE X	DEC	AS-CII
						D 1	D 2	D 3	D 4	D 5	D 6	D 7			
0	0	0	0	0	d	0	1	1	1	1	0	1	3D	61	=

Nota: Elaborado por (Escobar C, 2025)

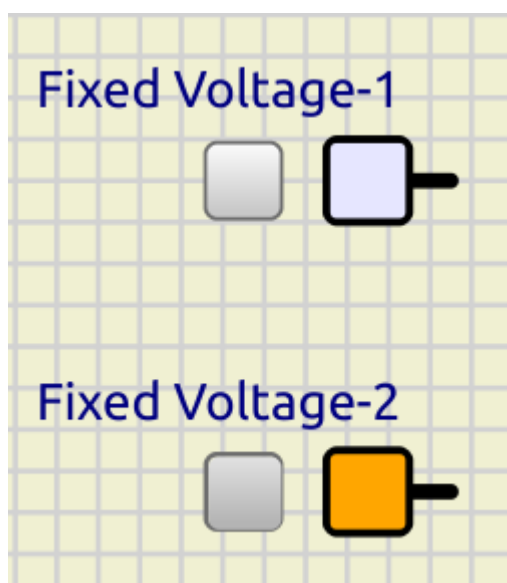
En este caso al ingresar “61” en el casillero de la izquierda, correspondiente a la dirección de memoria “00”, al dar enter o hacer clic fuera de la celda, este se convierte automáticamente en su equivalente hexadecimal, es decir “3D” y se proyecta a la derecha en esa dirección de memoria el código ASCII correspondiente, en este caso; “=”

De esta forma procedemos a llenar toda la tabla de memoria conforme la columna DEC de la tabla con lo que se obtiene el resultado mostrado en la tabla de memoria de la parte inferior de la Figura 4.

Diseñar la interfaz de switchs electrónicos.

En capítulos previos vimos que los switch electrónicos o conmutadores se pueden implementar usando transistores BJT o CMOS, sin embargo, en este caso se usarán los switchs de voltaje fijo del simulador SimulIDE (González et al., 2025), mismos que se muestran en la Figura 3 y que serán usados en las distintas fases del proyecto de acuerdo a como se necesite y que nos permitirán conectar las entradas de nuestros diseños o entradas de componentes digitales a tierra; como cero lógico o a voltaje; como uno lógico.

Figura 3 Interruptores de voltaje fijo del simulador SimulIDE



Nota: Elaborado por (Escobar C, 2025)

Simulación de la implementación

En esta última etapa y para simular el circuito que proyecte el mensaje almacenado en memoria a un display de siete segmentos, vamos a usar inicialmente un contador 74HC191, el mismo que será reemplazado posteriormente con el contador asíncrono diseñado previamente, esto con el fin de minimizar la posibilidad de errores en esta etapa, de tal manera que una vez que funcione correctamente el circuito procederemos con el reemplazo descrito.

Adicionalmente vamos a incorporar una etapa a la salida del contador con un decodificador BDC a 7 segmentos (Bcd a 7s en SimulIDE), conectado a un display de 7 segmentos a través de un banco de resistencias, esto nos permitirá monitorear la salida del contador y que a la vez es la entrada a nuestro circuito decodificador de binario de 4 bits al mensaje a presentarse secuencialmente en el display de 7 segmentos.

Este decodificador BCD a 7 segmentos es un circuito digital que convierte un número binario codificado en decimal (BCD, Binary-Coded Decimal) de 4 bits en las señales necesarias para activar los 7 segmentos de un display de 7 segmentos, permitiendo mostrar el número decimal correspondiente (del 0 al 9) y en el caso de simulIDE, este decodificador usa una tabla extendida que también permite presentar las combinaciones 1010 a 1111 correspondiente a los hexadecimales A a F.

Mientras que a la salida de la memoria pondremos dos bancos de 4 LEDs, con su banco de resistencias para monitorear el contenido en hexadecimal de la memoria, la cual a su vez alimenta el display de 7 segmentos con su respectivo banco de resistencias. Esto se puede apreciar con detalle en la Figura 4.

Esta misma implementación puede ser útil para presentar cualquier mensaje almacenado en la memoria Rom/Ram, lo que quiere decir que el estudiante pudiera basarse en este diseño y actualizar el contenido de la memoria para presentar su propio mensaje.

Para realizar la implementación de la simulación, arrastre los componentes mostrados en la Figura 4 al lienzo de trabajo y realice las conexiones tal como se muestra en la figura, reserve los espacios para los componentes y las conexiones.

Tenga cuidado con las conexiones de los túneles de conexión que permiten ser etiquetados para evitar un cableado excesivo pero que si no se identifican apropiadamente, generará problemas en el funcionamiento del circuito.

En SimulIDE, los túneles de conexión (en inglés Tunnel Tool) son una herramienta que permite conectar puntos sin necesidad de cables visibles, esto es ideal para mantener el esquema limpio y organizado, especialmente en circuitos grandes o complejos.

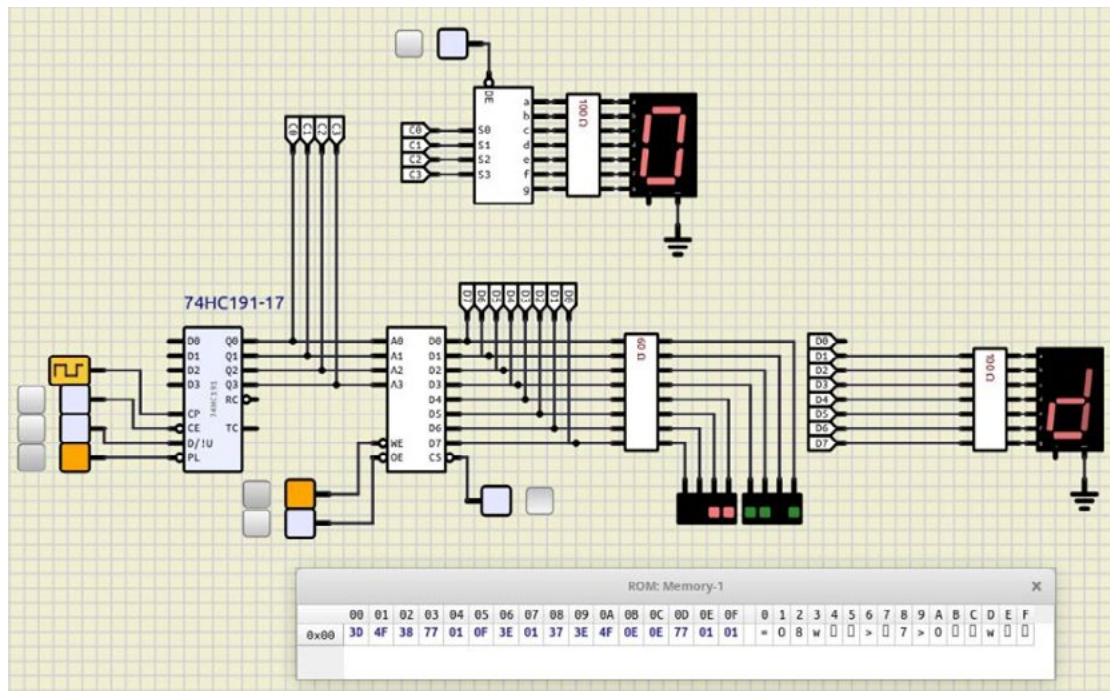
También identifique con claridad cuáles son las salidas y entradas más significativa y menos significativas, ya que si se las conecta en sentido inverso entre distintas etapas se tendrá un funcionamiento no esperado.

De manera ilustrativa implementaremos una parte del mensaje usando el contador 74HC191 y la segunda parte del mensaje se mostrará reemplazando este contador por el contador asíncrono de diseño propio, tal como se detalla a continuación.

Las Figura 5 a la Figura 9 presentan la salida del circuito decodificador de mensaje, usando el contador 74HC191, para las combinaciones del contador; 0001 - 0010 - 0011 - 0100 - 0101 y 0110, correspondientes a; “d E J A – t U”

Figura 4

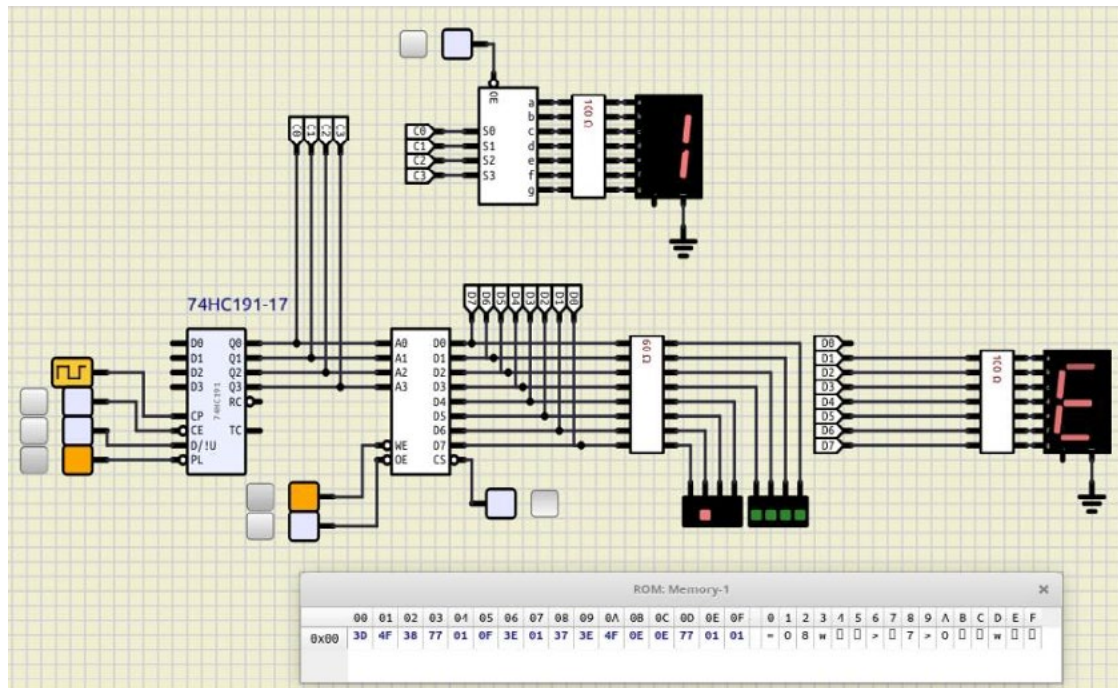
Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 0000 del contador



Nota: Elaborado por (Escobar C, 2025)

Figura 5

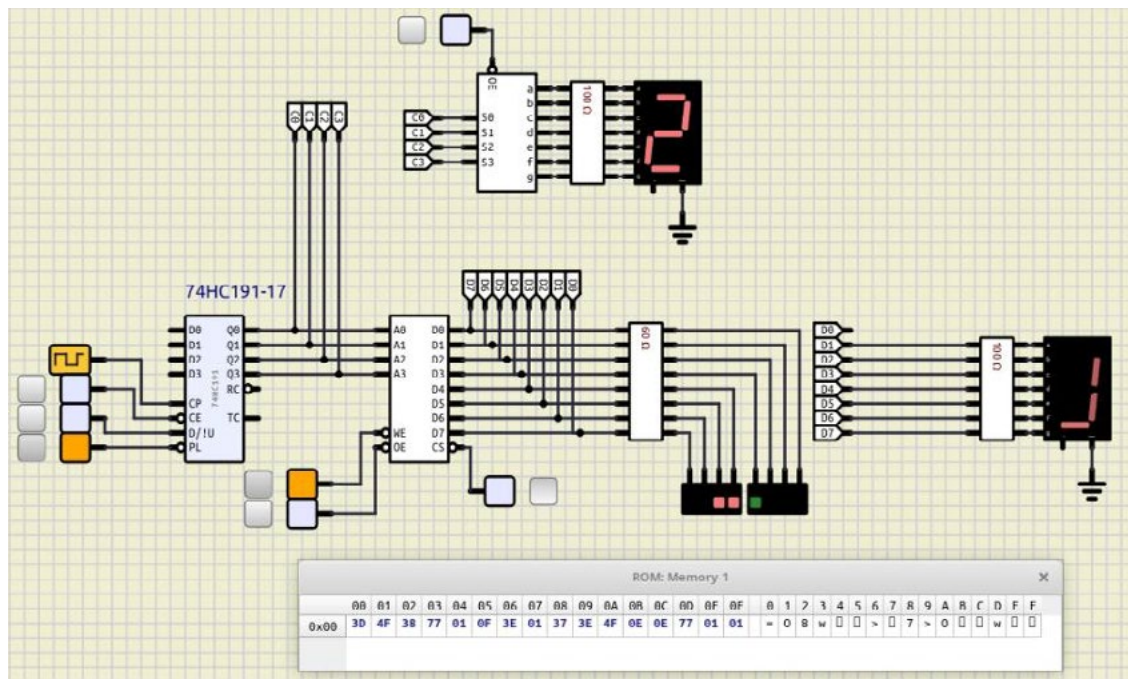
Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 0001 del contador



Nota: Elaborado por (Escobar C, 2025)

Figura 6

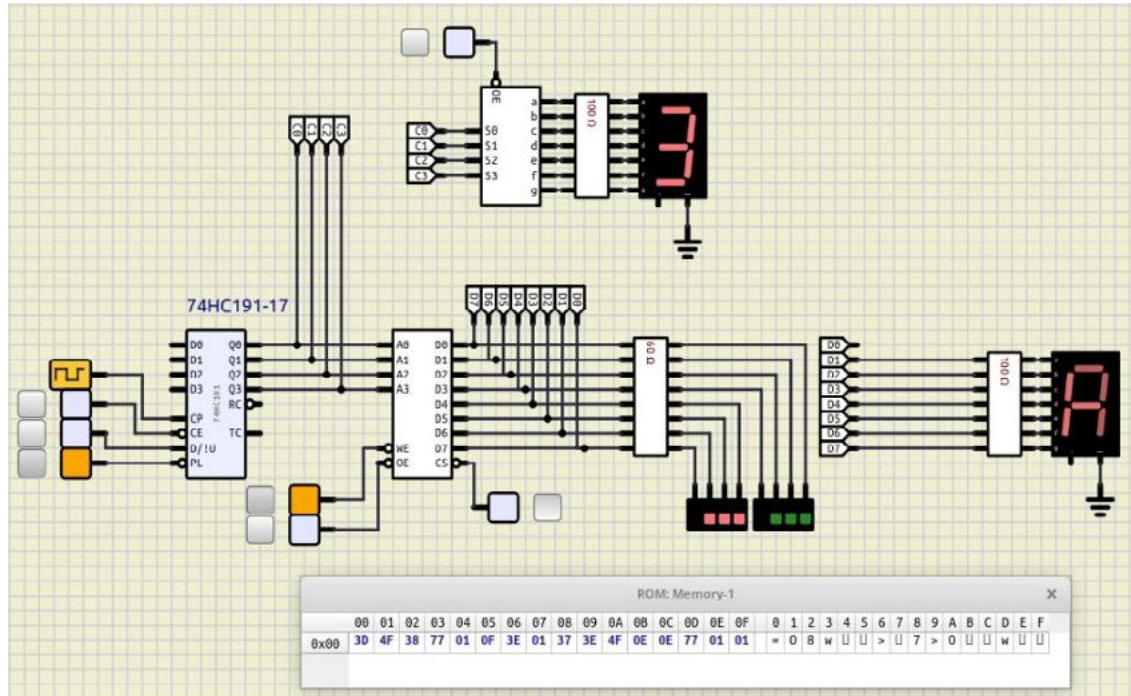
Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 0010 del contador



Nota: Elaborado por (Escobar C, 2025)

Figura 7

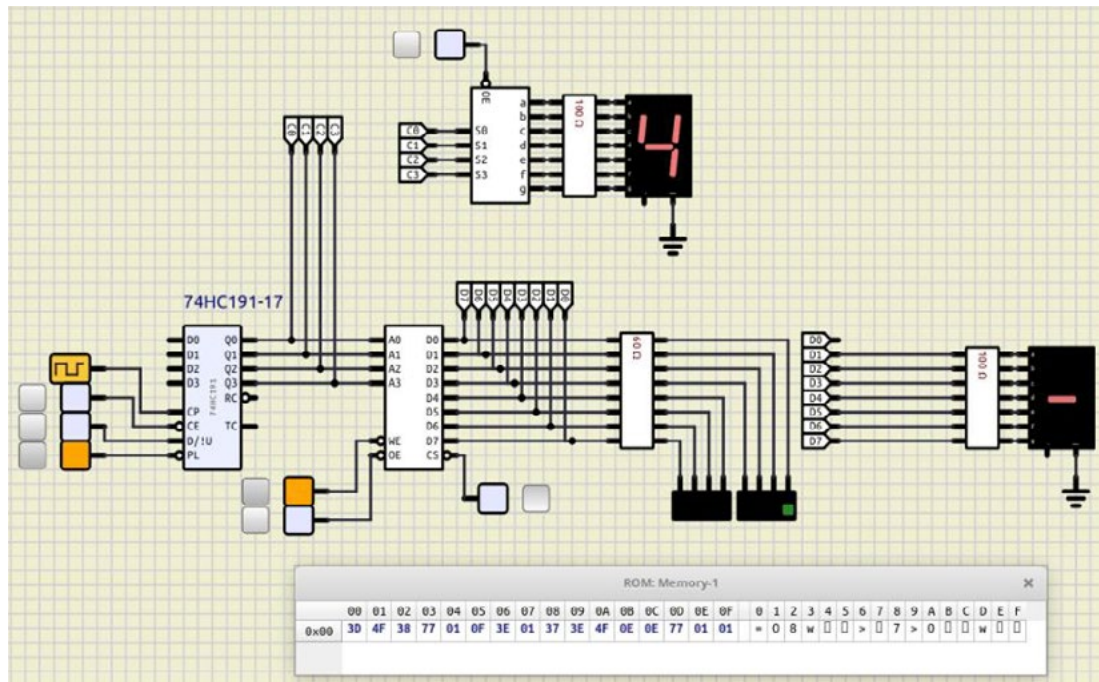
Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 0011 del contador



Nota: Elaborado por (Escobar C, 2025)

Figura 8

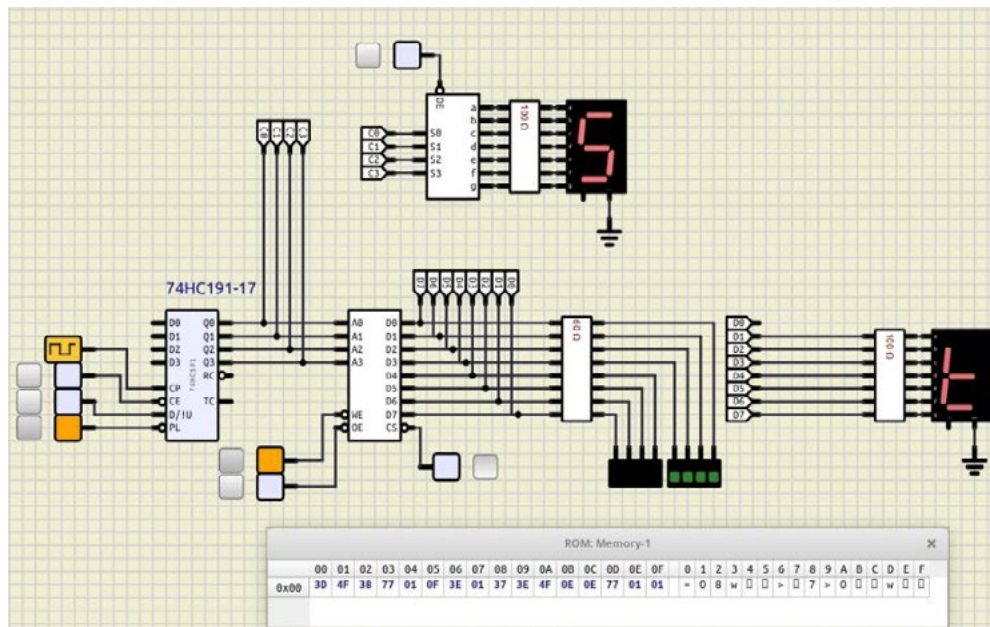
Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 0100 del contador



Nota: Elaborado por (Escobar C, 2025)

Figura 9

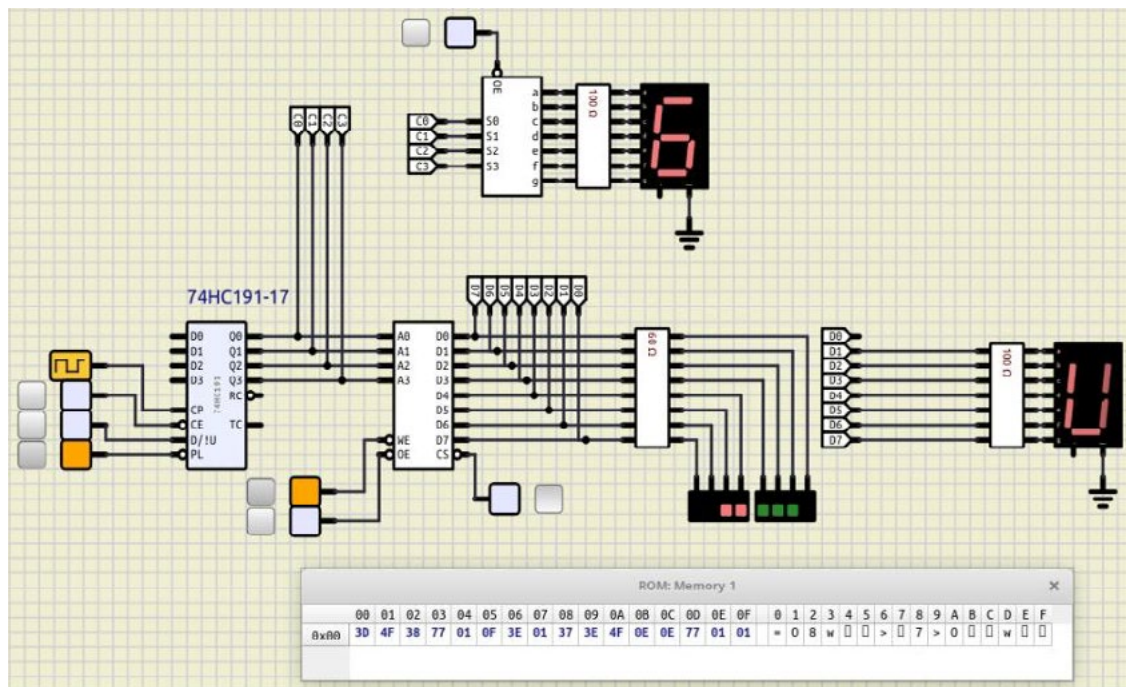
Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 0101 del contador



Nota: Elaborado por (Escobar C, 2025)

Figura 10

Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 0110 del contador



Nota: Elaborado por (Escobar C, 2025)

Una vez que hemos comprobado el funcionamiento del circuito con el contador, ya podemos reemplazar este contador 74HC191 por nuestro contador asíncrono creado previamente con flip-flops JK conectados en cascada.

En este punto podemos diseñar nuestro propio circuito generador de la señal de reloj, ya que si bien es cierto, nuestra simulación usa el reloj de los componentes de SimulIDE, este podría ser reemplazado con un circuito integrado 555 configurado como un reloj en la frecuencia deseada, de 1 Hz.

[En este enlace puede conocer mas detalles del circuito integrado 555 y sus aplicaciones.](#)

Para el diseño del reloj con el circuito integrado 555 en modo astable se puede generar una onda cuadrada en su pin de salida (pin 3) donde la frecuencia de esta señal depende de los valores de las resistencias R_A y R_B y del condensador C .

La frecuencia de salida está dada por la Ecuación 1:

Ecuación 1

Fórmula para el cálculo de frecuencia en modo astable para el circuito integrado 555

$$f = \frac{1.44}{(R_A + 2R_B) \cdot C} \{ (R_A + 2R_B) \cdot C \}$$

Nota: Elaborado por (Escobar C, 2025)

En esta ecuación podemos determinar los valores para R_A , R_B y C , para lo cual elegimos primero los valores para el condensador y una de las resistencias, así el condensador será de 10 μF , mientras que la resistencia R_A será de 10 Kohms, con esto podemos calcular el valor de R_B para lo cual, haciendo los respectivos reemplazos en la Ecuación 1, encontramos el valor de R_B de 68 Kohms. En nuestro caso se han seleccionado valores que existen comercialmente, sin embargo, si los valores calculados no se encuentran comercialmente disponibles se puede usar un potenciómetro como resistencia variable y ajustarla al valor requerido. En el caso de SimulIDE, estos valores pueden ser definidos nominalmente sin problema en todos los elementos.

La tabla 3 muestra el mapa descriptivo de conexiones para que circuito integrado 555 funcione como un reloj oscilador de 1 Hz.

Tabla 3

Mapa descriptivo de conexiones del circuito integrado 555

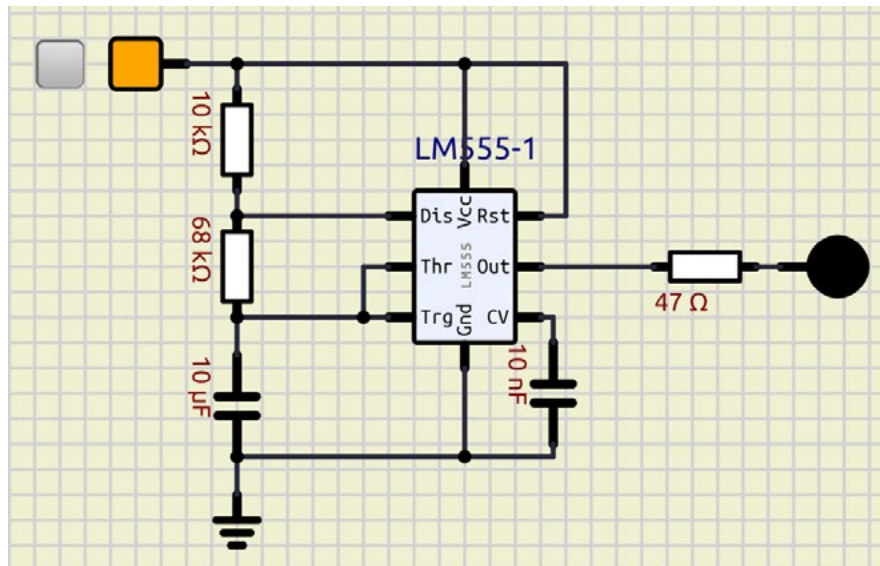
Pin	Función	Conexión
1	GND	Tierra
2	Disparo	Entre R_B y C
3	Salida	LED o osciloscopio
4	Reset	Conectado a +5V
5	Control	10nF a tierra (opcional)
6	Umbral	Conectado al pin 2
7	Disparo	Entre R_A y R_B
8	Vcc	+5V

Nota: Elaborado por (Escobar C, 2025)

La Figura 11 muestran el circuito de reloj con el circuito integrado 555 configurado como un oscilador astable ajustado a la frecuencia de 1 Hz.

Figura 11

Circuito de reloj de 1 Hz elaborado con el circuito integrado 555



Nota: Elaborado por (Escobar C, 2025)

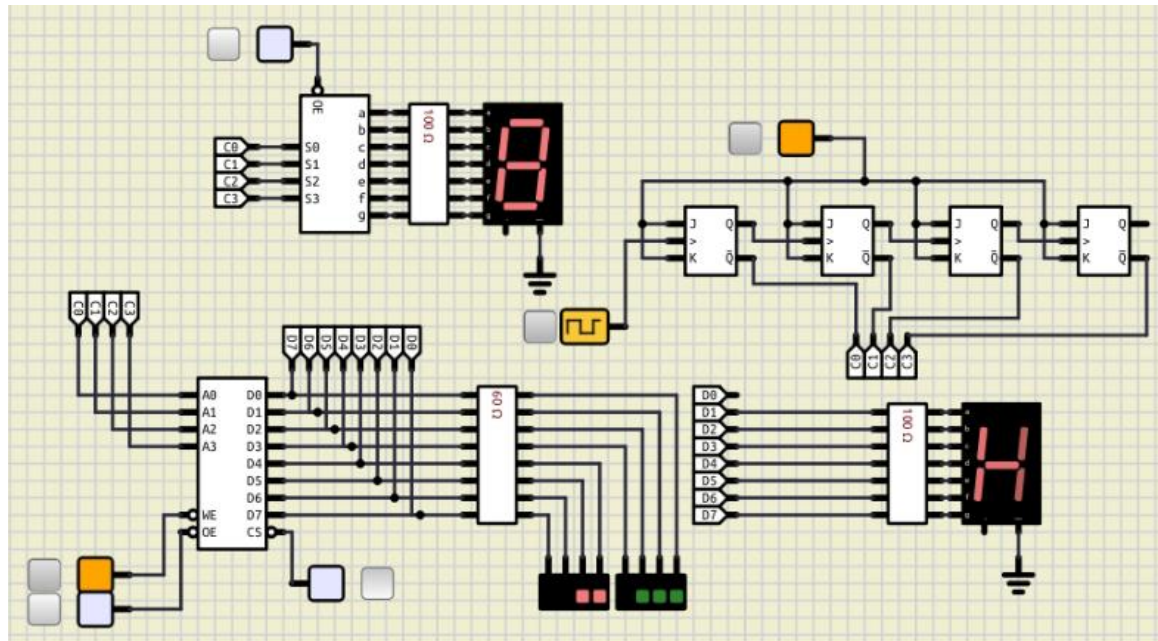
Las Figura 12 a la Figura 15 muestran el circuito decodificador de mensajes donde se ha reemplazado el contador 74HC191 por el contador asíncrono de flip-flops JK para las combinaciones 0111 - 1000 - 1001 - 1010 con la salida “- H U E “, mientras que el resto del mensaje se presentará ya con el reemplazo del reloj por nuestro oscilador diseñado con el 555.

Figura 12Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 0111 del contador asíncrono elaborado con flip-flops

Nota: Elaborado por (Escobar C, 2025)

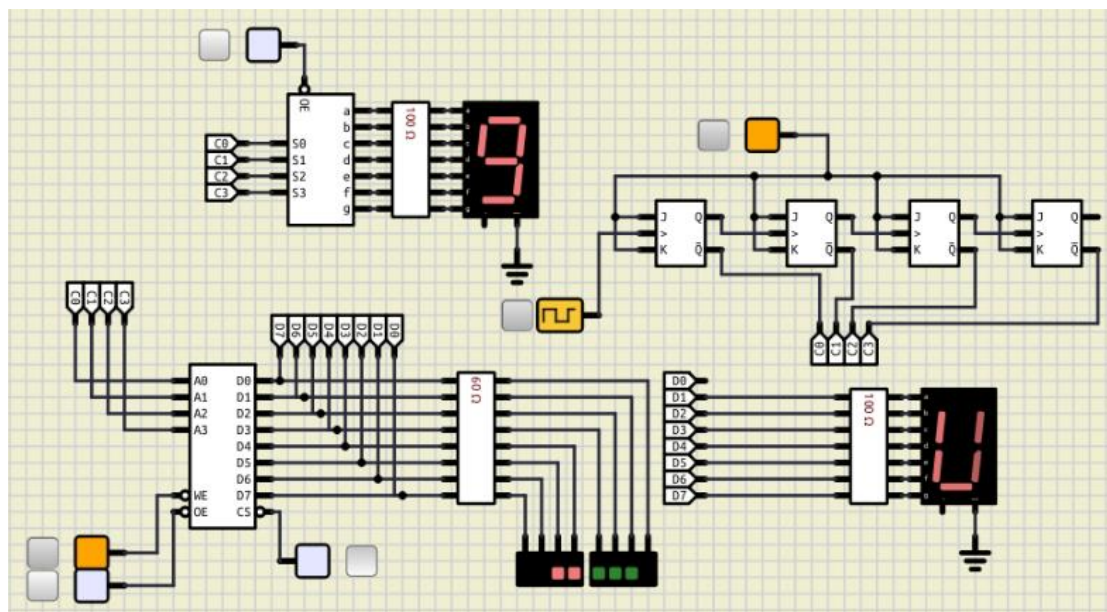
Figura 13

Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 1000 del contador asíncrono elaborado con flip-flops



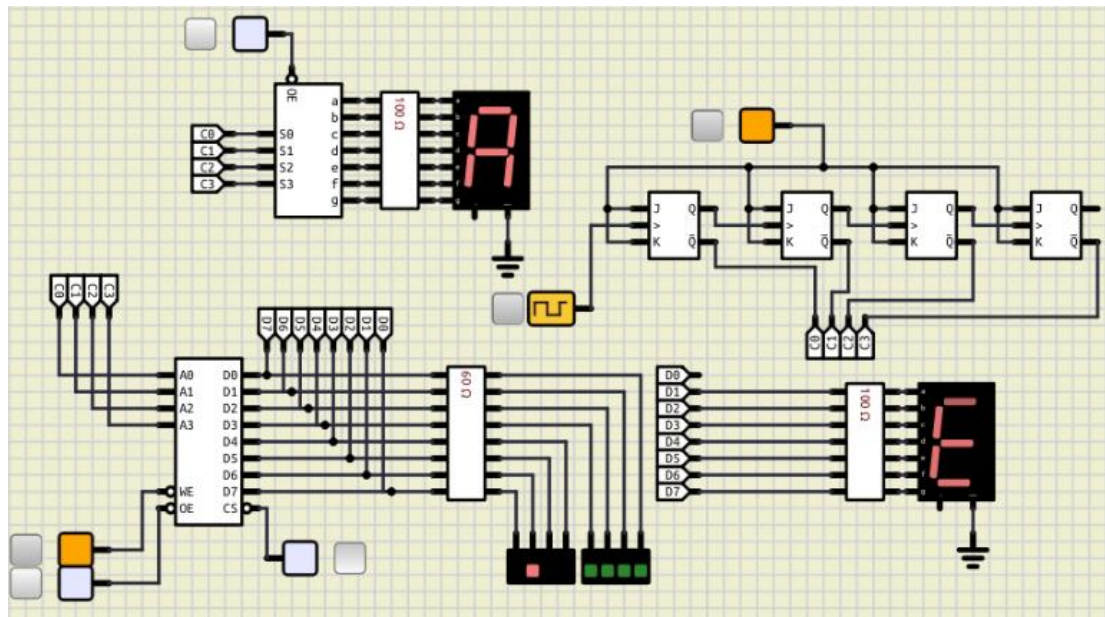
Nota: Elaborado por (Escobar C, 2025)

Figura 14 Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 1001 del contador asíncrono elaborado con flip-flops



Nota: Elaborado por (Escobar C, 2025)

Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 1010 del contador asíncrono elaborado con flip-flops

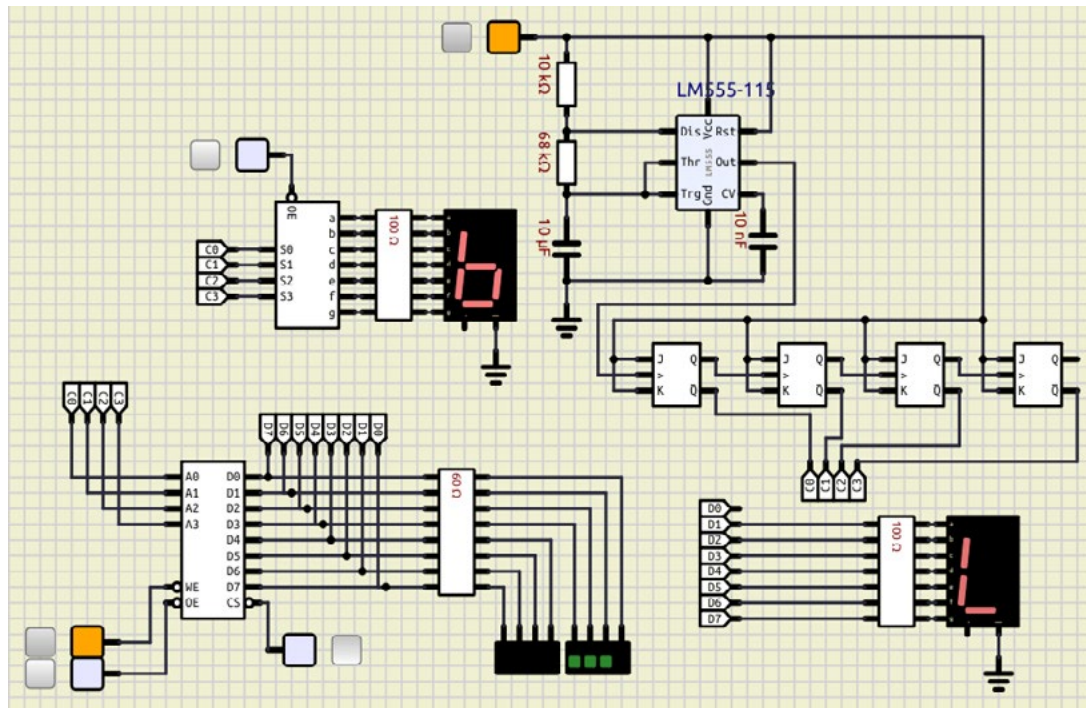


Nota: Elaborado por (Escobar C, 2025)

Las Figura 16 a la Figura 20 muestran el circuito decodificador de mensajes donde se ha reemplazado el reloj con el 555 para las combinaciones 1011 - 1100 - 1101 - 1110 - 1111 con la salida “L L A - -“, con lo que el mensaje está completo y también nuestro producto final.

Figura 16

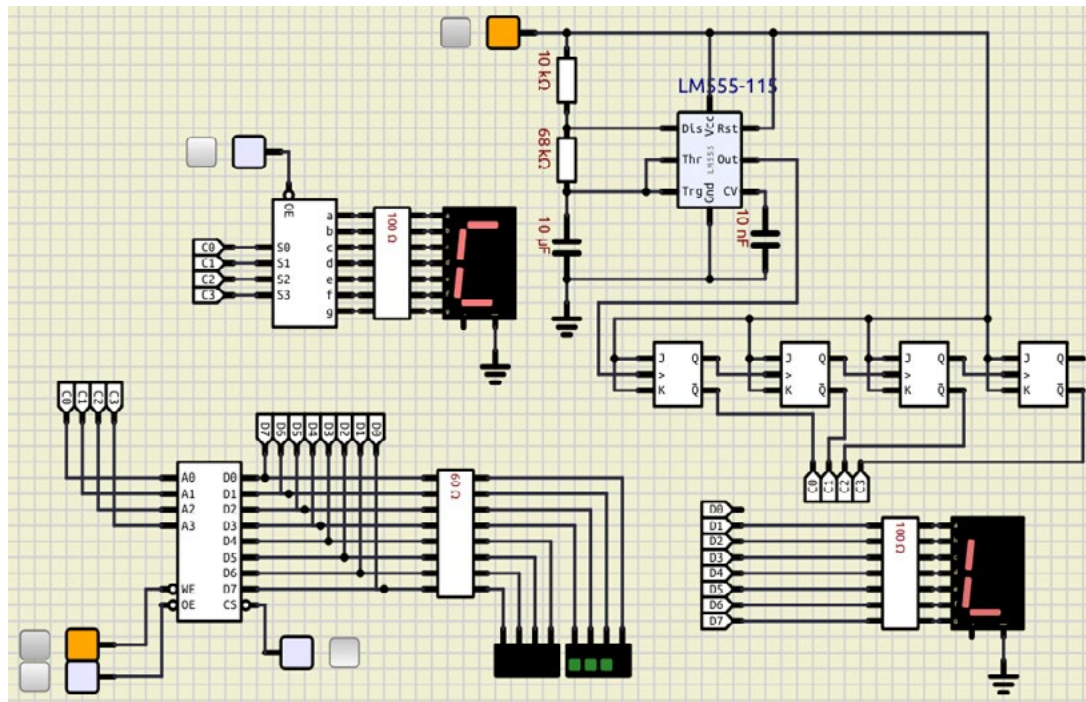
Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 1011 del contador asíncrono elaborado con flip-flops



Nota: Elaborado por (Escobar C, 2025)

Figura 17

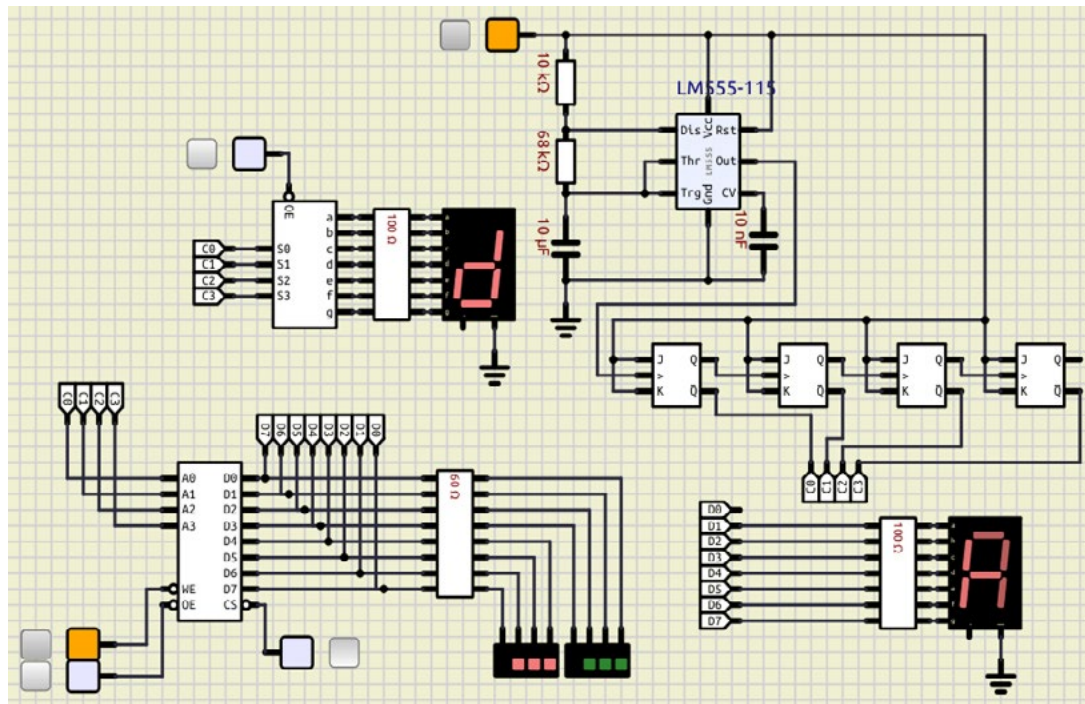
Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 1100 del contador asíncrono elaborado con flip-flops



Nota: Elaborado por (Escobar C, 2025)

Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 1101 del contador asíncrono elaborado con flip-flops

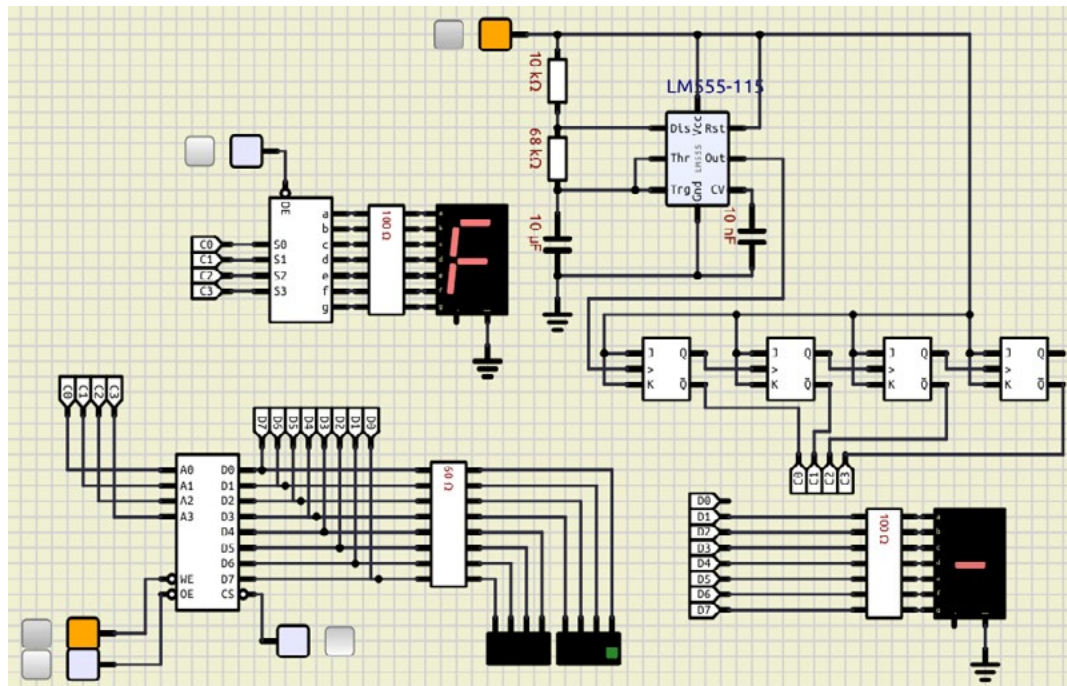
Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 1101 del contador asíncrono elaborado con flip-flops



Nota: Elaborado por (Escobar C, 2025)

Figura 20

Circuito decodificador de mensajes usando una memoria Ram/Rom que presenta la salida de la combinación 1111 del contador asíncrono elaborado con flip-flops



Nota: Elaborado por (Escobar C, 2025)

Finalmente es importante resaltar que en esta implementación y como elemento de control del funcionamiento del circuito se tienen los siguientes módulos:

- El decodificador BCD a 7 segmentos a la salida del contador asíncrono que muestran el número hexadecimal de la secuencia y que es a la vez la dirección de memoria leída.
- Los bancos de 4 LEDs de color rojo y verde, donde se proyecta el número hexadecimal almacenado en la memoria.
- El display de 7 segmentos que muestra el mensaje individual de cada estudiante.

Referencias citadas en la Clase 16.

González, S., De Francisco, P., Padrah, Z., Vrana, J., Kellner, K., Salishhchev, A., & Otros. (2025). Simulador de circuitos SimulIDE. Simulador de Circuitos SimulIDE. <https://simulide.com/p/>

Definición de los términos citados en la Clase 16.

Decodificador BCD a 7 segmentos: es un circuito digital que convierte un número binario codificado en decimal (BCD, Binary-Coded Decimal) de 4 bits en las señales necesarias para activar los 7 segmentos de un display de 7 segmentos, permitiendo mostrar el número decimal correspondiente (del 0 al 9).

Túneles de conexión (Tunnel Tool): son una herramienta que en SimulIDE, permiten conectar puntos sin necesidad de cables visibles, ideal para mantener el esquema limpio y organizado, especialmente en circuitos grandes o complejos.



La excelencia no se improvisa

síguenos

