

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización Computacional

Ing. Juan Carlos Maeda Juárez

Aux: Javier Gutierrez



PRÁCTICA #1

SIMULACIÓN DE UN VISUALIZADOR DE 7 SEGMENTOS (DISPLAY)

INTRODUCCIÓN

En la actualidad, la mayoría de los sistemas están compuestos por circuitos combinacionales. Estos circuitos son fundamentales en el diseño de sistemas digitales y se utilizan para realizar diversas operaciones lógicas. Los circuitos combinacionales consisten en una serie de compuertas lógicas, que al ser combinadas transforman un conjunto de entradas en una única salida. Las compuertas lógicas, como AND, OR, NOT, NAND, NOR, XOR y XNOR, son los elementos básicos utilizados en los circuitos combinacionales. Cada una de estas compuertas realiza una operación lógica específica y se conectan entre sí para implementar la función lógica deseada.

OBJETIVOS

General

Aplicar los conocimientos teóricos aprendidos en clase magistral y laboratorio para la construcción de circuitos combinacionales.

Específicos

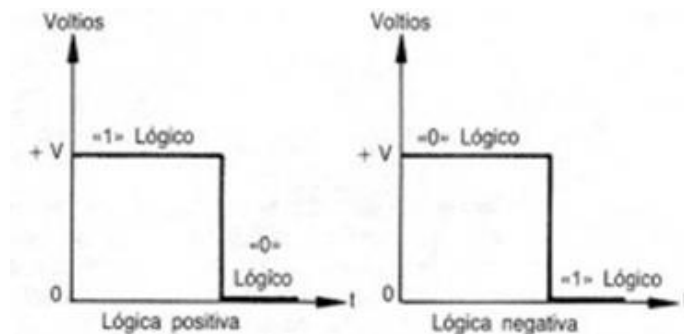
1. Poner en práctica los conocimientos de Lógica Combinacional y Mapas de Karnaugh.
2. Conocer el funcionamiento de transistores y realización de compuertas lógicas transistorizadas.
3. Crear un dispositivo de visualización a mayor escala (Display).
4. Utilizar lógica negativa y positiva durante el desarrollo de la práctica.

MARCO TEÓRICO

Lógica Negativa y Positiva

La señal binaria a la entrada o salida de cualquier circuito puede tener uno de dos valores, excepto durante la transición. Un valor debe ser mayor que el otro ya que tienen que ser diferentes para poder distinguirlos. Designese el nivel alto como H (High) y el nivel bajo como L (Low). Hay dos alternativas para la asignación de la lógica.

Figura I: Ejemplo gráfico de lógica negativa y positiva



Fuente: C. Amaury, tomado de:

<https://circuitos-logicos-aarc.webnode.mx/blog/logica-negativa/>

Términos mínimos y máximos

Una Función Lógica que está compuesta por operadores lógicos puede ser expresada en forma canónica usando los conceptos de minterm y maxterm. Todas las funciones lógicas son expresables en forma canónica, tanto como una “suma de minterms” como “producto de maxterms”. Esto permite un mejor análisis para la simplificación de dichas funciones, lo que es de gran importancia para la minimización de circuitos digitales.

Figura II: Uso de mintérminos y maxtérminos

x	y	z	Mini-términos		Maxi-términos	
			Término	Designación	Término	Designación
0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

Fuente: P. Juan, tomado de:

<https://circuitosdigitalesudblog.wordpress.com/2020/06/03/practica-5/>

Familias Lógica

Las familias lógicas son conjuntos de circuitos integrados (chips) que implementan funciones lógicas, como AND, OR, NOT, etc. Estos chips se utilizan en la electrónica digital para procesar y manipular señales binarias (1s y 0s), que son la base de la computación y los sistemas digitales. Cada familia lógica tiene sus propias características de rendimiento, consumo de energía, velocidad, costo y niveles de voltaje de operación.

Familia Lógica TTL

La familia lógica TTL es una de las familias lógicas más comunes y ampliamente utilizadas en la electrónica digital. "TTL" significa "Transistor-Transistor Logic", lo que indica que estos chips están contruidos con transistores bipolares.

Características TTL

- Velocidad de conmutación rápida: Los chips TTL pueden cambiar de estado (de 0 a 1 o viceversa) muy rápidamente, lo que los hace adecuados para aplicaciones que requieren una alta velocidad de procesamiento.
- Consumo de energía moderado: Aunque no es la familia lógica más eficiente en términos de consumo de energía, tiene un consumo de energía aceptable para muchas aplicaciones.
- Compatibilidad universal: Los niveles de voltaje lógico de entrada y salida son compatibles con una amplia gama de dispositivos y sistemas, lo que facilita su integración en diseños más grandes.

Voltaje Umbral en TTL

Una característica crítica de los chips TTL es el "voltaje umbral" o "nivel de disparo" (threshold voltage en inglés). Los transistores bipolares en un chip TTL requieren una cierta cantidad de voltaje para cambiar su estado. A este valor específico se le llama "voltaje umbral". Por lo general, para la familia TTL, el voltaje umbral es de aproximadamente 0.8 V para un nivel lógico bajo (0) y 2.0 V para un nivel lógico alto (1).

Importancia de una Fuente de Poder Estable

Dado que los chips TTL tienen un voltaje umbral definido para reconocer niveles lógicos, es crucial proporcionar una fuente de alimentación con un voltaje estable y bien regulado a +5 voltios (5V). Si el voltaje de la fuente no es lo suficientemente alto para superar el voltaje umbral requerido, los transistores no cambiarán de estado adecuadamente, lo que podría llevar a errores o comportamientos inesperados en el circuito.

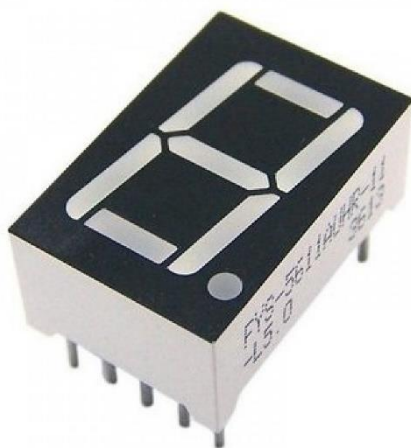
Una fuente de alimentación con un voltaje estable de +5V garantiza que los niveles lógicos se interpreten de manera confiable y precisa, lo que es esencial para un funcionamiento correcto del circuito TTL y para evitar daños en los componentes electrónicos.

En resumen, la familia lógica TTL es ampliamente utilizada en electrónica digital debido a su velocidad de conmutación rápida y su compatibilidad universal. Sin embargo, es crucial suministrar una fuente de alimentación estable y bien regulada a +5V para garantizar un funcionamiento confiable, tomando en cuenta el voltaje umbral necesario para el cambio de estado de los transistores en los chips TTL. Esto es especialmente importante al realizar simulaciones o prácticas con protoboard o fibra de vidrio, donde la precisión y estabilidad del voltaje son fundamentales para un correcto funcionamiento de los circuitos.

DESCRIPCIÓN DE LA PRÁCTICA

Una empresa de publicidad contrató a su grupo de laboratorio de Organización Computacional para diseñar e implementar un circuito combinacional de 4 bits en Proteus para las entradas W, X, Y y Z, con el objetivo de generar una salida A. Esta salida deberá mostrar una palabra de 16 letras y/o números seleccionada por cada grupo, utilizando un display de 7 segmentos.

Figura III: Display de 7 segmentos.



Fuente: <https://laelectronica.com.gt/display-color-rojo-de-7-segmentos-catodo-comun>

Como la empresa no puede tener a un operando realizando las combinaciones ya que sería un trabajo muy desgastante, por lo que utilizando Arduino se requiere realizar un contador ascendente de 4 bits que será el encargado de realizarlas estando conectado a la entrada del circuito combinacional.

METODOLOGÍA DE TRABAJO

- La palabra por implementar será la combinación de las primeras 4 letras del nombre de cada integrante (en dado caso el nombre tenga menos tiene que incluir letras de su apellido), ejemplo:

- Juan
- María
- Pedro
- Álvaro

La palabra será: JUAN-MARI-PEDR-ALVA

- Una de las funciones deberá ser implementada en placa, queda a discreción del grupo cual función realizar, se permite el uso de placa perforada.

LISTADO DE COMPONENTES PERMITIDOS

Nombre	Código
AND	74ls08
OR	74ls32
NOT	74ls04
NAND	74ls00
NOR	74ls02
XOR	74ls86
XNOR	74ls266
RESISTENCIAS	
DISPSWITCH	
LED	

DOCUMENTACIÓN

Se deberá documentar cada parte del circuito, agregando capturas del diseño del circuito con una breve descripción de su funcionalidad. Agregado a esto, la documentación debe contener:

1. Carátula.
2. Introducción.
3. Objetivos.
4. Contenido.
 - a. Funciones booleanas.
 - b. Mapas de Karnaugh.
 - c. Diagramas del diseño del circuito.
 - d. Equipo utilizado.
 - e. Presupuesto (utilizado para la parte física)
 - Gastos totales
5. Aporte individual de cada integrante.
6. Conclusiones.
7. Anexos.
 - a. Diagrama del circuito impreso.
 - b. Fotografías de los circuitos físicos (protoboard y placa).
 - c. Enlace al video grupal de los circuitos físicos.

NOTA:

Para tener derecho a calificación de documentación, deberán entregar simulación completa del circuito en Proteus.

RESTRICCIONES

1. Las copias totales o parciales tendrán una nota de 0 y serán reportadas a la escuela.
2. El simulador para utilizar debe de ser Proteus.
3. La entrega de la documentación debe ser con el siguiente formato:
P1_G#.pdf
4. No se puede utilizar logicstate para dar voltaje en Proteus.
5. Se calificará un único archivo de Proteus, es decir que todos los módulos deben ir unidos en este mismo, de lo contrario se penalizará.
6. El circuito físico deberá funcionar en su totalidad, de lo contrario se penalizará.
7. Usuario de GitHub al cual se debe de añadir como colaborador:
 - PikaGuty

.

ENTREGABLES

1. Un archivo con extensión .pdsprj que contenga el circuito combinacional.
2. Documentación.
3. Se requiere entregar el enlace al repositorio grupal de GitHub con la siguiente nomenclatura: P1_ORGA_G# el cual contendrá todo lo mencionado anteriormente.

FORMA DE ENTREGA

Mediante UEDI, subiendo el enlace del repositorio. Solamente una persona del grupo debe realizar la entrega.

Nota: La fecha límite de entrega es el domingo 9 de junio de 2024, antes de las 23:59. Se calificará el lunes 10 de junio.

ANEXOS

Abecedario en display de 7 segmentos:

