

870116

UNIVERSIDAD AUTONOMA DE GUADALAJARA

Incorporada a la Universidad Nacional Autónoma de México

ESCUELA DE INGENIERIA EN COMPUTACION

9²
Gen.



TESIS CON
FALLA DE ORIGEN

PRACTICAS PARA EL LABORATORIO DE DISEÑO
LOGICO BASADAS EN UNA CALCULADORA

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

PRESENTA:

SONIA VILLA SOTO

GUADALAJARA, JAL

JUNIO DE 1989



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

TESIS CON FALLA DE ORIGEN

INDICE

	INTRODUCCION	1
	ANTECEDENTES	2
CAPITULO I	CIRCUITO DE ENTRADA	3
	1.1 Circuito Codificador	4
	1.2 Práctica No. 1	5
	1.3 Registro de almacenamiento	10
	1.4 Práctica No. 2	12
	1.5 Circuito para controlar la transferencia de datos	17
	1.6 Práctica No. 3	19
	1.7 Práctica No. 4	23
CAPITULO II	CIRCUITO ARITMETICO	25
	2.1 Sumador BCD	26
	2.2 Práctica No. 5	28
	2.3 Generador de complemento a nueve	32
	2.4 Práctica No. 6	34
	2.5 Circuito controlador de operaciones	38
	2.6 Práctica No. 7	41
	2.7 Práctica No. 8	45
CAPITULO III	CIRCUITO DE SALIDA	47
	3.1 Circuito multiplexor	48
	3.2 Práctica No. 9	49
	3.3 Circuito decodificador	53
	3.4 Práctica No. 10	54
	3.5 Desplazado de información	58
	3.6 Práctica No. 11	59
	3.7 Práctica No. 12	60
	CONCLUSIONES	62
	ANEXOS	63
	BIBLIOGRAFIA	74

INTRODUCCION

El Diseño Lógico es un campo muy amplio y sumamente interesante dentro de la electrónica, al que conocemos como Electrónica Digital.

La electrónica digital ha conocido, en los últimos años un notable y gran desarrollo, observando que tiene bastantes campos de aplicación como son las comunicaciones digitales, el control de sistemas, diseño de calculadoras y computadoras.

Es por eso que cada día es más necesario adentrarse en el estudio de esta materia.

La presente tesis consiste en el diseño de varias prácticas para el laboratorio de Diseño Lógico, estas prácticas consisten en ir armando por partes un sencillo calculador que realizará -- las operaciones de suma y resta para un dígito, obviamente este dispositivo no es con fines prácticos sino educativos, lógicamente sólo se utilizarán dispositivos básicos para el diseño de sistemas digitales por tratarse de la materia de Diseño Lógico.

El objetivo principal que se pretende lograr al realizar este trabajo es que el estudiante reafirme sus conocimientos adquiridos en clase mediante la realización de una aplicación específica de los diferentes dispositivos estudiados.

El tener los conocimientos teórico de una materia adquiridos mediante la consulta de libros y manuales, la enseñanza de un maestro hace que en un momento el enseñando esté preparado -- teóricamente, pero al llevar a cabo las prácticas de laboratorio le proporciona la destreza necesaria para aplicar sus conocimientos teóricos al desarrollo de sistemas digitales.

ANTECEDENTES:

La materia de Diseño Lógico abarca un campo muy amplio dentro de la electrónica, por lo que surge la necesidad no sólo de estudiarla en teoría sino de practicarla, esto con el fin de conocer más ampliamente cada dispositivo visto en clase.

Encontramos que desde que inicia el laboratorio de diseño lógico existen diversos manuales de consulta que nos hablan de diferentes prácticas.

En un principio el trabajo desarrollado en el laboratorio era la realización de prácticas recopiladas de manuales considerándose éstas las de mayor importancia y utilidad para el alumno.

Cada vez crecen las personas interesadas en ésta área de electrónica por lo mismo las necesidades del laboratorio también aumentan para proporcionar una práctica más completa que vaya de acuerdo a la teoría que ha adquirido el alumno, por lo tanto se han diseñado manuales de laboratorio cada vez más específicos y enfocados a las necesidades que se van presentando.

De ahí surge la idea de crear este trabajo para integrar en una sola práctica la publicación de los diferentes componentes básicos de diseño.

C A P I T U L O I.
"Circuito de Entrada".

Los dispositivos de entrada proporcionan la información --- necesaria para su procesamiento a cualquier sistema de cómputo, --- ya sea una computadora o una calculadora, etc.

El dispositivo de entrada más común es el teclado, el cual nos proporciona una serie de señales lógicas que van a ser transformadas en algún código determinado antes de ser procesadas por los circuitos digitales.

En nuestro caso necesitamos de convertir nuestra información en código BCD.

El circuito de entrada que vamos a utilizar lo podemos dividir en tres partes para su explicación.

- 1.- Circuito codificador.
- 2.- Registro de almacenamiento.
- 3.- Circuito para controlar la transferencia de datos a los registros.

El circuito de entrada completo primeramente convierte la información decimal a código BCD, después almacena el dígito en su registro correspondiente para posteriormente enviarlo al circuito de salida, así como al circuito aritmético para realizar las operaciones correspondientes.

Para seleccionar la operación y los signos de los dígitos --- tenemos dos switches, cuando el switch de operación está en 0, --- realiza la operación de suma y cuando está en 1 la de resta, pero el signo un 1 significará un signo negativo y un 0 un signo positivo.

1.1 CIRCUITO CODIFICADOR.

Los circuitos codificadores son sistemas combinatoriales de 2^n entradas y n salidas, realizados de tal forma que cuando una sola de las entradas asume un estado lógico determinada 0 o 1 a la salida aparece la combinación binaria correspondiente a el número decimal asignado a dicha entrada.

El circuito codificador que vamos a utilizar es el de la figura 1.1.1.

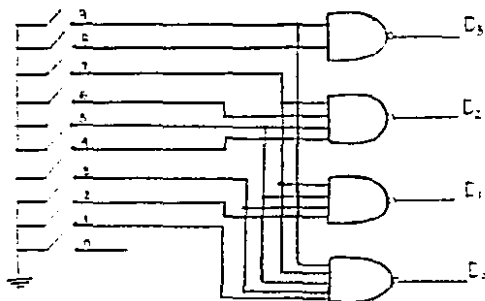


Figura 1.1.1

Como los swiches están produciendo entradas altas a las compuertas NAND teniendo como salida un 0, cuando presionar un switch envía un bajo a las compuertas NAND produciendo así que el código correcto BCD de salida se produce.

Para el sistema del codificador podemos haber utilizado compuertas OR o un circuito integrado que realice esta función, pero los CI que venden en el mercado son codificadores con prioridad y en nuestro caso no es necesario.

1.2 PRACTICA No. 1

OBJETIVO:

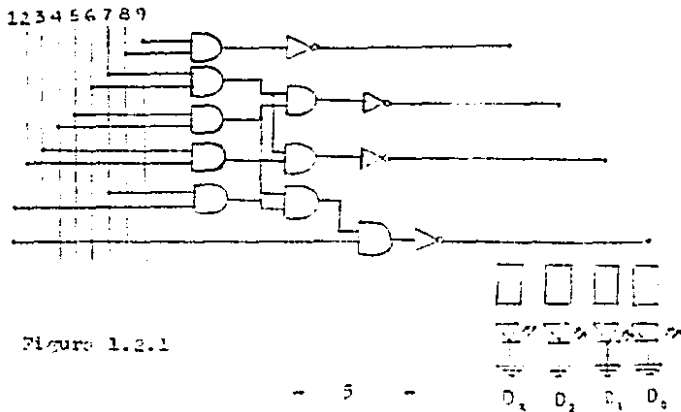
- Comprender como se construye un codificador.
- Mostrar cómo un codificador puede servir como dispositivo de entrada.

LISTA DE MATERIAL:

- Proto.
- Fuente de alimentacion de 5V dc.
- Multímetro.
- 3 IC TTL SN7408.
- 1 IC TTL SN7404.
- Alambres.
- 3 resistencias de 330 Ohms.
- 4 Led's.

PROCEDIMIENTO.

- 1.- Arma el circuito de la figura 1.2.1



2.- Obtén la tabla 1.2.1

Entradas.	Salidas.
0 1 2 3 4 5 6 7 8 9	D ₃ D ₂ D ₁ D ₀
0 1 1 1 1 1 1 1 1 1	
1 0 1 1 1 1 1 1 1 1	
1 1 0 1 1 1 1 1 1 1	
1 1 1 0 1 1 1 1 1 1	
1 1 1 1 0 1 1 1 1 1	
1 1 1 1 1 0 1 1 1 1	
1 1 1 1 1 1 0 1 1 1	
1 1 1 1 1 1 1 0 1 1	
1 1 1 1 1 1 1 1 0 1	
1 1 1 1 1 1 1 1 1 0	

Tabla 1.2.1

PREGUNTAS.

- 1.- Qué otra aplicación tiene un codificador ?
- 2.- Diseña un codificador de octal a binario con compuertas OR.
- 3.- Investiga qué función realiza el IC DM74147.
- 4.- Dibuja cómo harías las conexiones del IC DM74147, arma el --
circuito y obtén su tabla de verdad.

RESPUESTAS.

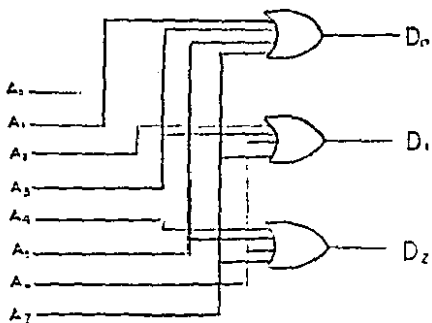
1.- Además de utilizarse como dispositivo de entrada se utiliza para convertir información de un código a otro código conecctífico.

2.-	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	D_2	D_1	D_0
	1	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	1
	0	0	1	0	0	0	0	0	0	1	0
	0	0	0	1	0	0	0	0	0	1	1
	0	0	0	0	1	0	0	0	1	0	0
	0	0	0	0	0	1	0	0	1	0	1
	0	0	0	0	0	0	1	0	1	1	0
	0	0	0	0	0	0	0	1	1	1	1

$$D_0 = A_8 + A_5 + A_3 + A_1$$

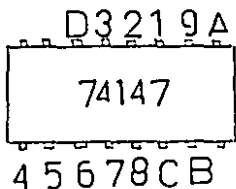
$$D_1 = A_7 + A_6 + A_4 + A_2$$

$$D_2 = A_8 + A_5 + A_4 + A_1$$



3.- Es un codificador de prioridad, codifica nueve líneas de ---
 datos a cuatro, o sea que es un codificador de decimal a BCD.
 La condición de cero no es necesaria, se tendrá una salida -
 para ésta condición cuando las 9 líneas de entrada tengan un
 nivel lógico alto.

4.- 1 2 3 4 5 6 7 8 9	D C B A
1 1 1 1 1 1 1 1 1	1 1 1 1
X X X X X X X X 0	0 1 1 0
X X X X X X X 0 1	0 1 1 1
X X X X X X 0 1 1	1 0 0 0
X X X X X 0 1 1 1	1 0 0 1
X X X X 0 1 1 1 1	1 0 1 0
X X X 0 1 1 1 1 1	1 0 1 1
X X 0 1 1 1 1 1 1	1 1 0 0
X 0 1 1 1 1 1 1 1	1 1 0 1
0 1 1 1 1 1 1 1 1	1 1 1 0



1.3 REGISTRO DE ALMACENAMIENTO.

Una vez que se tiene la salida del codificador, debemos almacenar éste código que corresponde a la tecla presionada, en un registro para poder trabajar con él.

El elemento de memoria más utilizado es el Flip Flop (FF).- Estos circuitos son celdas binarias capaces de almacenar un bit de información. Un circuito flip-flop tiene dos salidas una para el valor normal y una para el valor complemento del bit almacenado en él. La información binaria puede entrar a un flip-flop en una variedad de formas, hecho éste, que da termina diferentes tipos de flip-flops.

Un circuito flip-flop puede mantener un estado binario indefinidamente hasta que se cambie por una señal de entrada para com cambiar estados. La principal diferencia entre varios tipos de flip-flops es el número de entradas que poseen y la manera en la cual las entradas afectan el estado binario.

El flip-flop tipo D es el que se usa principalmente en la transferencia de datos binarios, ya que la salida Q del flip-flop irá al mismo estado que está presente en la entrada D cada que oc ocurre un pulso de reloj.

Para nuestro diseño ocupamos dos registros de 4 bits, cada uno para almacenar los dos dígitos con los cuales se harán las operaciones, también ocupamos dos registros más para almacenar el signo de cada dígito.

Para formar un registro de 4 bits ocupamos 4 flip-flops tipo D, el CI 74175 es un circuito que contiene 4 flip-flops tipo D con una entrada de reloj común a los 4, por lo que se adapta a nuestra necesidad.

Para almacenar los signos utilizamos el CI 7474 que es un circuito que contiene dos flip-flops tipo D con entradas de reloj independientes.

El circuito que vamos a utilizar para nuestro registro de almacenamiento lo construimos en la figura 1.3.1.

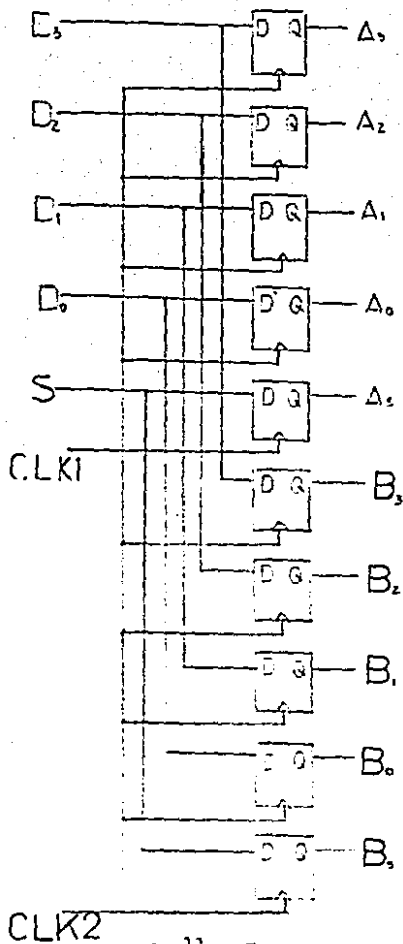


Figura 1.3.1

1.4 PRACTICA No. 2

OBJETIVO:

- Mostrar que un flip-flop es un elemento de memoria.
- Construir dos registros para almacenar los dígitos de 4 bits.
- Construir dos registros de un bit para almacenar el signo de los dígitos.

LISTA DE MATERIAL:

- Proto.
- Fuente de alimentación de 5 V dc.
- Multímetro.
- 2 IC TTL SN74175.
- 1 IC TTL SN7474.
- 10 resistencias de 330 Ohms.
- 10 Led's.
- Alambres de conexión.

PROCEDIMIENTO:

- 1.- Arme el circuito de la figura 1.4.1.

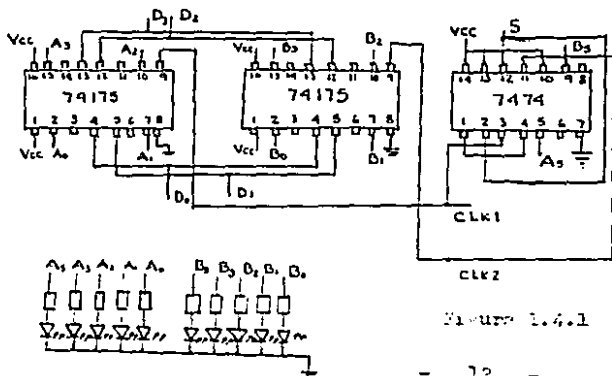


Figura 1.4.1

2.- Aplique la siguiente entrada a los registros.

D₃ D₂ D₁ D₀ S
0 0 0 1 1

3.- Aplique un pulso al registro A y visualice en los Led's esta entrada, en el registro B la salida no debe ser afectada.

4.- Aplique otra entrada a los registros.

D₃ D₂ D₁ D₀ S
0 0 1 1 0

5.- Aplique un pulso al registro B y de nuevo observe que la salida del reg. B cambi6 más no la del reg. A.

6.- Repita los casos 2,3,4 y 5 para las siguientes entradas y llene la tabla.

<u>D₃</u> <u>D₂</u> <u>D₁</u> <u>D₀</u> <u>S</u>	<u>A₃</u> <u>A₂</u> <u>A₁</u> <u>A₀</u> <u>S</u>	<u>B₃</u> <u>B₂</u> <u>B₁</u> <u>B₀</u> <u>S</u>
1 1 1 1 1		
0 0 0 0 0		
1 1 0 0 0		
1 1 1 0 1		
0 0 1 1 1		
0 1 0 1 1		

PREGUNTAS.

- 1.- Explica qué es un flip-flop tipo D.
- 2.- Explica el funcionamiento del IC SN74175 y el IC SN7474.
- 3.- Menciona algunas de las aplicaciones de los flip-flops.
- 4.- Que tipo de transferencia de información es la que realiza el circuito de la figura 1.4.1 y por que se llama así ?

RESPUESTAS.

- 1.- De aquel en el cual la salida del flip-flop irá al mismo estado que está presente en la entrada D, cuando quiera que ocurra una transición positiva o negativa según sea en la entrada CLK.
- 2.- El 74175 tiene una entrada sincrónica CLEAR y otra entrada de reloj que son comunes a los 4 flip-flops tipo D que contiene este IC, tiene ambas salidas Q y Q'. Cuando ocurre una transición positiva en la entrada de reloj lo que hay en D pasa a la salida Q,

Tabla de verdad.

Entradas			Salidas.	
CLR	CLK	D	Q	Q'
0	X	X	0	1
1	↑	1	1	0
1	↑	0	0	1
1	0	X	Q	Q'

El 7474 tiene 2 flip-flops tipo D con entradas individuales de: D, CLEAR, CLOCK, PRESET y ambas salidas de Q y Q'.

La información de la entrada D será transferida a Q en el borde de subida del pulso de reloj.

Tabla de verdad.

Entradas				Salidas.	
PR	CLR	CLK	D	Q	Q'
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	No estable.	
1	1	↑	1	1	0
1	1	↑	0	0	1
1	1	0	X	Q	Q'

3.- En operaciones de transferencia.

En la construcción de contadores, registros de desplazamiento etc.

4.- Es una transferencia en paralelo puesto que la información presente en las entradas es transferida simultáneamente al registro.

CIRCUITO PARA CONTROLAR LA TRANSFERENCIA DE DATOS.

Una operación que ocurre muy frecuentemente en sistemas digitales es la transferencia de información de un lugar a otro.

Bajo la ocurrencia de un pulso de reloj lo vamos realizar esta operación.

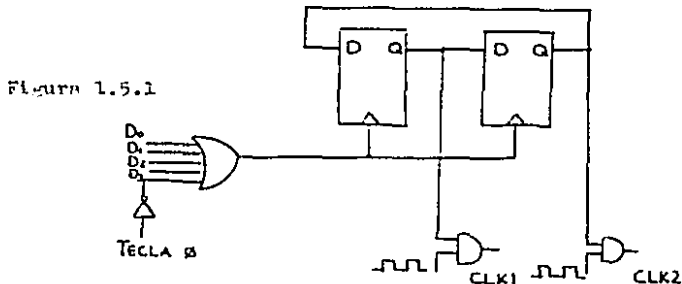
En nuestro caso lo que deseamos es que al primer dígito sea almacenado en un registro y el segundo en otro registro, así como sus respectivos signos.

Las entradas de los dos registros están conectadas entre sí cuando se presiona una tecla tenemos a la salida del codificador el dígito que se va a almacenar, lo que queremos es guardarlo en el primer registro (REG A) por lo que hay que aplicarle un pulso, mientras que el segundo registro (REG B) debe estar inhabilitado para recibir los datos.

Para almacenar el segundo dígito se revierte lo anterior pero con el registro B.

Esta operación de transferencia la podemos lograr con un contador de anillo, que mientras que nos proporciona un pulso a un registro el otro está inhabilitado y viceversa.

La figura 1.5.1 nos muestra el circuito.



Cada vez que se presiona una tecla, la salida de la compuerta OR se hace alta, enviando un pulso al contador de anillo, poniéndolo en el estado 10. La transición de 10 hace que las salidas de el codificador sean transferidas al registro A, lo mismo ocurre con el signo.

La misma operación ocurre para almacenar el segundo dígito.

Las compuertas AND que nos proporcionan las entradas de reloj a los registros se pusieron únicamente para asegurarnos que la transferencia ocurre en la transición positiva, para evitar un posible error.

PRACTICA No. 3

OBJETIVO:

Mostrar como un contador de anillo no puede servir para controlar la transferencia de datos entre la salida del codificador y los registros de almacenamiento.

LISTA DE MATERIALES:

- Proto.
- Fuente de alimentación de 5 V d.c.
- Multímetro.
- 1 IC SN7474.
- 1 IC TTL SN7432.
- 2 Resistencias de 330 ohms.
- 2 Led's.
- Alambres de conexión.

PROCEDIMIENTO:

- 1.- Arma el circuito de la Figura 1.6.1 y obtén su tabla de verdad.
- 2.- Al iniciar colocar el pin de CLR del primer flip-flop y el pin de PR del segundo flip-flop a tierra, después se los dejas en voltaje, esto hace que inicie el contador en 01.

7474

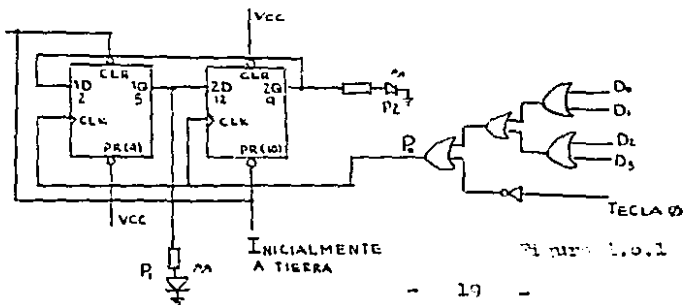


Figura 1.6.1

Tabla de verdad.

Entradas				Salidas.	
D ₃	D ₂	D ₁	D ₀	P ₁	P ₂
0	0	0	0		
0	0	1	1		
1	1	0	0		
1	1	1	1		

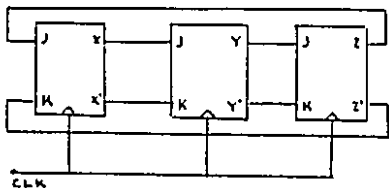
3.- Los valores de P₁ y P₂ no deben de estar en el mismo nivel --
lógico.

PRGUNTAS.

- 1.- Explica qué es un contador de anillo ?
- 2.- Menciona una aplicación del contador de anillo.
- 3.- Respecto al circuito que se armó explica por qué hay que invertir la señal que se produce al presionar la tecla 0.
- 4.- Diseña un contador de anillo con flip-flops JK.

RESPUESTAS:

- 1.- Es un tipo de registro de desplazamiento circulante conectado de tal modo que el último flip-flop deslaza su valor en el primer flip-flop.
- 2.- Para controlar la secuencia de operaciones en un sistema.
- 3.- Cada vez que se presiona una tecla produce una salida alta - en la compuerta OR enviándole un pulso al contador de anillo, cuando se presiona la tecla 0 produce una salida baja por lo que se requiere se invierta para enviar el pulso al contador.
- 4.-



1.7 PRACTICA No. 4.

OBJETIVO:

Integrar las tres primeras prácticas para formar el circuito de entrada completo, este circuito nos va a proporcionar como salida el código correspondiente al dígito al ser procesado, así como su signo correspondiente.

LISTA DE MATERIAL:

- Fuente de alimentación de 5 V dc.
- Multímetro.
- Prácticas No. 1, 2 y 3
- Alambres para conexión.

PROCEDIMIENTO:

- 1.- En el diagrama de la figura 1.7.1 se muestra en bloques las tres prácticas así como la manera de conectarlas entre sí.
- 2.- Desconecte los led's visualizadores de cada práctica y haga las conexiones necesarias de acuerdo a la figura 1.7.1
- 3.- No olvides inicializar el contador de anillo.
- 4.- Complete la tabla 1.7.1.

Entradas					Salidas.																		
S	D ₃	D ₂	D ₁	D ₀	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	B ₈	B ₇	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀	
1	0	0	0	0																			
0	1	1	1	1																			
1	0	0	1	1																			
1	1	1	0	0																			
0	1	0	1	0																			

Tabla 1.7.1

CIRCUITO DE ENTRADA

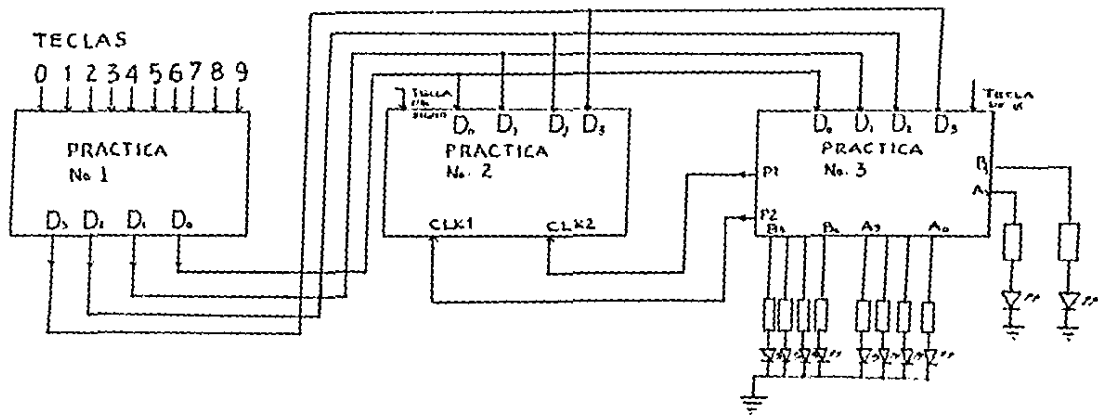


FIGURA 17.1

C A P I T U L O I I .
"Circuito Aritmético ".

La función esencial de la mayoría de las computadoras y calculadoras es la ejecución de operaciones aritméticas.

El componente básico de la sección aritmética es un sumador paralelo; que se construye con un número de circuitos sumadores completos conectados en cascada. Controlando la entrada de datos al sumador es posible obtener diferentes tipos de operaciones aritméticas.

El circuito aritmético que se presenta realiza las operaciones de suma y resta en código decimal codificado en binario (BCD), esto con el fin de facilitar el diseño, ya que las conversiones entre decimal y binario pueden ser largas y complicadas, por esta razón se utiliza dicho código ya que combina algunas características de los sistemas decimal y binario.

El circuito aritmético que se analizará es el circuito sumador/restador BCD natural para una sola etapa.

Podemos dividirlo en tres partes que son:

- Sumador BCD.
- Generador de complemento a nueve.
- Controlador de operaciones.

El circuito completo realiza las operaciones de suma y resta para números positivos y negativos es por eso la necesidad del circuito controlador de operaciones ya que es independiente la operación que desea el usuario o la que realiza el circuito aritmético.

Enseguida se analizará cada uno de los circuitos que forman la parte aritmética del calculador que se está construyendo.

2.1 SUMADOR BCD.

El proceso de suma se puede resumir en los siguientes pasos:

- 1.- Para aquellas posiciones donde la suma es de nueve o menos, la suma está en la forma apropiada y no necesita corrección.
- 2.- Cuando la suma de los dígitos es mayor que nueve, debe añadirse una corrección de 0110 para producir el resultado apropiado. Esto produce un acarreo al ser añadido a la siguiente posición decimal.

Esto lo podemos observar en la tabla 2.1.1

Suma Binaria					Suma BCD				Decimal
K	Z	Z	Z	Z	C	S	S	S	S
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	0	0	1	0
0	0	0	1	1	0	0	0	1	1
0	0	1	0	0	0	0	1	0	0
0	0	1	0	1	0	0	1	0	1
0	0	1	1	0	0	0	1	1	0
0	0	1	1	1	0	0	1	1	1
0	1	0	0	0	0	1	0	0	0
0	1	0	0	1	0	1	0	0	1
0	1	0	1	0	1	0	0	0	0
0	1	0	1	1	1	0	0	0	1
0	1	1	0	0	1	0	0	1	0
0	1	1	0	1	1	0	0	1	1
0	1	1	1	0	1	0	1	0	0
0	1	1	1	1	1	0	1	0	1
1	0	0	0	0	1	0	1	1	0
1	0	0	0	1	1	0	1	1	1
1	0	0	1	0	1	1	0	0	0
1	0	0	1	1	1	1	0	0	1

Tabla 2.1.1

El circuito logico que detecta la correccion puede derivarse de la tabla.

Se necesita una correccion cuando $K=1$, las otras seis combinaciones desde 1010-1111 que necesitan correccion tienen un 1 en Z_1 , para distinguirlos del numero 1000 y 1001 que tambien tienen un 1 en Z_3 , deben tener un 1 en Z_2 o en Z_4 .

Por lo que la condicion para que exista una correccion debe ser expresada por medio de una funcion de Boole:

$$C = K + Z_3Z_2 + Z_1Z_4.$$

Cuando $C=1$ es necesario corregir 0110 a la suma binaria y suministrar un arrastre de salida a lo siguiente etcetera.

El circuito sumador se muestra en la figura 2.1.1

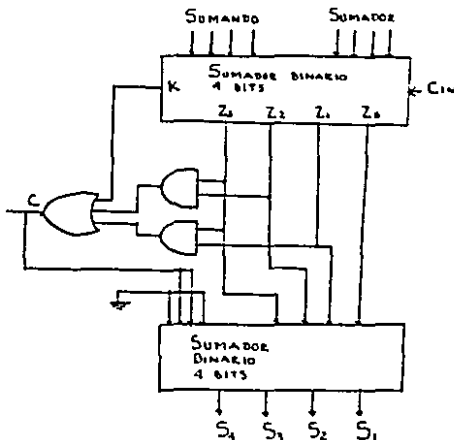


Figura 2.1.1

2.2 PRACTICA No. 5

OBJETIVO:

Mostrar que teniendo un sumador paralelo binario y una serie de compuertas lógicas construir un circuito sumador BCD.

LISTA DE MATERIAL:

- Proto.
- Fuente de alimentación de 5 V dc.
- Multímetro.
- 2 IC TTL SN7483.
- 1 IC TTL SN7408.
- 1 IC TTL SN7432.
- 5 Resistencias de 330 Ohms.
- 5 Led's.
- Alambres de conexión.

PROCEDIMIENTO:

- 1.- Arma el circuito de la figura 2.2.1
- 2.- Complete la tabla de verdad.

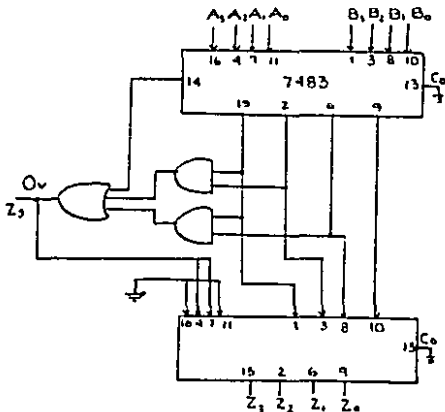
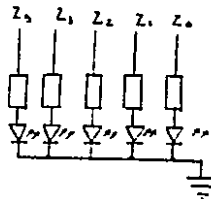


Figura 2.2.1



Entradas:

A ₁	A ₂	A ₃	A ₀	B ₁	B ₂	B ₃	B ₀
0	0	0	1	0	1	0	0
0	0	1	1	0	1	1	1
1	0	0	1	0	1	1	0
0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	1

Salidas:

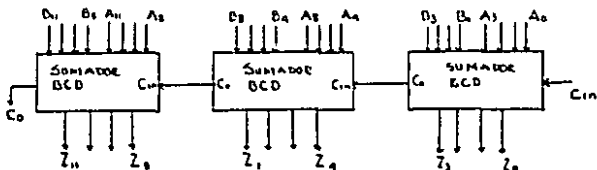
S ₀	S ₁	S ₂	S ₃
----------------	----------------	----------------	----------------

PREGUNTAS.

- 1.- Explica cuándo hay que hacer una corrección a la suma binaria para tenerla en código BCD.
- 2.- Menciona una ventaja de usar el sumador BCD y no el binario en el diseño del calculador.
- 3.- El circuito que se realizó es para una etapa, dibuje las conexiones que harías para tener un sumador de 3 dígitos.
- 4.- Qué otra aplicación le das a un circuito sumador binario?

REQUISITOS:

- 1.- Cuando la suma es mayor que nueve, hay que sumar 0110 para tener el resultado apropiado. Se suma 0110 para tomar en cuenta los 6 grupos codificados en vélices.
- 2.- La información procesada por cualquier sistema debe convertirse al decimal para que pueda ser interpretada fácilmente y el código BCD por combinar características de los sistemas binario y decimal hace más fácil las conversiones.
- 3.-



- 4.- Un circuito sumador binario tiene muchas aplicaciones ya que en la base para realizar las operaciones primitivas como un restador o un multiplicador.

2.3 GENERADOR DE COMPLEMENTO A NUEVE

La resta en el sistema decimal se puede realizar de una forma similar al sistema binario, sumando al minuendo el complemento a 10 o a 9 del sustraendo.

En los sistemas BCD se utiliza el complemento a 9 porque permite reducir la complejidad del sistema.

Puesto que el código BCD no es autocomplementario, el complemento a 9 no puede obtenerse complementando cada bit en el código.

Para diseñar el circuito generador de complemento a 9 podemos analizar la tabla 2.3.1

$A_3 A_2 A_1 A_0$	Complemento a nueve $B_3 B_2 B_1 B_0$
0 0 0 0	1 0 0 1
0 0 0 1	1 0 0 0
0 0 1 0	0 1 1 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 0 1
0 1 0 1	0 1 0 0
0 1 1 0	0 0 1 1
0 1 1 1	0 0 1 0
1 0 0 0	0 0 0 1
1 0 0 1	0 0 0 0

Tabla 2.3.1

Observamos que se trata de una multifunción que puede ser sintetizada por los métodos de simplificación. Pero por otra parte el complemento a nueve de un dígito se puede calcular -- invirtiendo el número y sumándole 1010, produciéndose un acarreo que no es tenido en cuenta.

Cuando este circuito se junta con el sumador BCD obtenemos un sumador/restador BCD.

En la figura 2.3.1 se muestra el circuito que genera el complemento a nueve de un dígito.

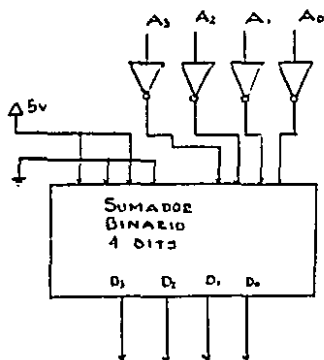


Figura 2.3.1

2.4 PRACTICA No.6

OBJETIVO:

Diseñar un circuito que genere el complemento a nueve de un número.

LISTA DE MATERIAL

- Proto.
- Fuente de alimentación de 5V dc.
- Multímetro.
- 1 IC TTL SN7483.
- 1 IC TTL SN7404.
- 4 Resistencias de 330 Ohms.
- 4 Led's
- Alambres para conexión.

PROCEDIMIENTO

- 1.- Arma el circuito de la figura 2.4.1
- 2.- Coloca los valores de la tabla 2.4.1 en las entradas E_0, E_1, E_2, E_3 y obtén sus salidas.

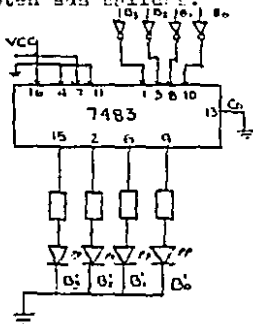


Figura 2.4.1

Entradas	Salidas
$B_3 B_2 B_1 B_0$	$B'_3 B'_2 B'_1 B'_0$
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	
1 0 0 0	
1 0 0 1	

Tabla 2.4.1

PREGUNTAS

1.- Mencione dos formas de realizar el complemento a nueve de un dígito en BCD.

2.- Obten el complemento a nueve de los siguientes números.

24 _____
50 _____
37 _____

3.- Diseñe un circuito que genere el complemento a nueve de un número que sea diferente al de la práctica.

4.- Porqué no complementamos bit por bit?

RESPUESTAS

- 1.- a) Se complementa el dígito y se suma 1010, el acarreo final se descarta.
 b) Se suma el binario 0110 al dígito y después se complementa bit a bit, el acarreo final también se descarta.

2.- 24 0111 0101
 50 0100 1001
 37 0110 0010

24., 0010 0100
 1101 1011₊
 1010 1010

 10111 10101

50., 0101 0300
 1010 1111₊
 1010 1010

 10101 11001

37., 0011 0111
 1100 1000₊
 1010 1010

 10110 10010

- 3.- Analizando la tabla de verdad para el generador de complemento a nueve y con un bit de modo C que: cuando C=0; X=B, cuando C=1; X= Complemento a nueve de B.

$$X = D_0 C' + B_0' C$$

$$X = B,$$

$$X = B_1 C' + (B_1' B_1 + D_1 B_1') C$$

$$X = B_1 C' + B_1 B_1' B_1' C$$

- 4.- Porque el código BCD no es un código autocomplementario.

2.5 CIRCUITO CONTROLADOR DE OPERACIONES

Hasta ahora se ha visto el circuito sumador y el generador de complemento; también sabemos que si los juntamos vamos a obtener un sumador/restador y podemos controlar las operaciones de suma y resta con un bit de modo Op que cuando $Op=0$, los dígitos se suman y cuando $Op=1$ los dígitos se restan, pero en nuestro caso las operaciones son para números con signo por lo que hay que hacer un análisis más detallado para determinar los algoritmos de suma y resta.

Designaremos la magnitud de dos números A y B. Cuando los números con signo se suman o restan encontramos que hay ocho -- condiciones diferentes para considerar, dependiendo del signo y de la operación que se realice.

Estas consideraciones se enumeran en la tabla 2.5.1, las otras columnas en la tabla muestran la operación actual a realizar con las magnitudes de los números. La última columna es para prevenir un cero negativo.

Operación	Suma las magnitudes	Resta las magnitudes			
		Cuando:	A > B	A < B	A = B
$(+A)+(+B)$	$+(A+B)$				
$(+A)+(-B)$			$+(A-B)$	$-(B-A)$	$+(A-B)$
$(-A)+(+B)$			$-(A-B)$	$+(B-A)$	$+(A-B)$
$(-A)+(-B)$	$-(A+B)$				
$(+A)-(+B)$			$+(A-B)$	$-(B-A)$	$+(A-B)$
$(+A)-(-B)$	$+(A+B)$				
$(-A)-(+B)$	$-(A+B)$				
$(-A)-(-B)$			$-(A-B)$	$+(B-A)$	$+(A-B)$

Los algoritmos de suma y resta se derivan de la tabla 2.5.1 y pueden enunciarse como sigue.

Algoritmo de suma: Cuando los signos de A y B son idénticos se suman las magnitudes y se pone el signo de A al resultado, -- cuando los signos de A y B son diferentes se comparan las magnitudes y se resta el número menor del mayor, el signo del resultado será el mismo que el de A si $A > B$ o el complemento del signo de A si $A < B$. Si las magnitudes son iguales, se resta B de A y -- el signo del resultado es positivo.

El algoritmo para la resta es similar que el de la suma -- excepto por la comparación del signo, el procedimiento para signos idénticos en suma es el mismo que el procedimiento para signos diferentes en resta y viceversa.

Para implementar lo anterior en hardware tenemos:
Primariamente se definirán las variables a utilizar

As signo de A
Bs signo de B
Op operación Op=0 : suma, Op=1: resta.
C control de complemento.
Ov sobreflujo.
Z suma Z=0: resultado=0, Z=1: resultado \neq 0.

Los dos signos As y Bs son comparados por una compuerta OR-exclusiva, si la salida de la compuerta es 0 los signos son idénticos; si es 1 los signos son diferentes.

Las magnitudes se suman si los signos son idénticos en la operación suma y diferentes en la operación resta.

Las magnitudes se restan si los signos son diferentes para suma o idénticos para resta.

Podemos recurrir a la siguiente tabla:

As	Bs	As \oplus Bs	
0	0	0	signos idénticos.
0	1	1	signos diferentes.
1	0	1	signos diferentes.
1	1	0	signos idénticos.

El signo positivo se representa con un 0 y el negativo con un 1.

As0E2s	Op	As0E2s00 = B	Signo	Operación	Complemento
0	0	0	iguales	suma	no necesite
0	1	1	"	resta	si "
1	0	1	dif.	suma	si "
1	1	0	"	resta	no "

Ya hemos controlado si se realiza una operación de suma que implica que el número B no será complementado, o una operación de resta que indica que el número B será complementado. Para esto existen otros factores que debemos tomar en cuenta para obtener un resultado correcto, por ejemplo si el resultado es negativo habrá que sacar su complemento e indicar que es un número negativo, por lo que debemos de tener las siguientes consideraciones:

En la operación resta no debe ocurrir un sobreflujo de tal manera que si esto sucede deberá de ignorarse, ésta operación es la que realiza el circuito no la que desea el usuario; de lo anterior obtenemos la siguiente ecuación:

$$\text{Sobreflujo} = 7'0V$$

Obtendremos un resultado sin necesidad de corrección cuando:

- 1.- $B \neq 0$, haya sobreflujo y la operación sea resta con signos diferentes o suma con signos iguales.
- 2.- La operación sea suma y signos iguales.
- 3.- La operación sea resta y signos diferentes.

De lo anterior se genera la señal S1, el signo será el de A

$$S1 = B' + 0VCE$$

Bajo las siguientes condiciones el resultado se complementa
tal cual como el signo de A.

1.- No hay sobreflujo y la operación es resta y signos di-
ferentes.

Lo anterior genera la señal S2, su ecuación es:

$$S2 = Ov \cdot E$$

El circuito para obtener éstas señales es el de la figura
2.5.2.

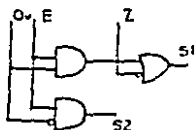


Figura 2.5.2

Para determinar el signo del resultado tenemos el circuito
de la figura 2.5.3

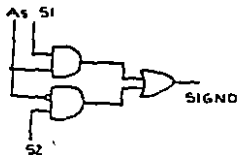


Figura 2.5.3

2.6 PRACTICA No. 7

OBJETIVO:

Que se comprenda porque hay que generar las seña-
les S1 y S2 para obtener el resultado correcto de las operaciones
así como el tipo de operación que deberá de realizar el circuito.

LISTA DE MATERIAL

- Proto
- Fuente de alimentación de 5V dc.
- Multímetro.
- 1 IC TTL SN 7408
- 1 IC TTL SN7404
- 1 IC TTL SN7432
- 1 IC TTL SN7486
- 3 Resistencias de 330 Ohms
- 2 Led's
- Alambres para conexión

PROCEDIMIENTOS

- 1.- Arma el circuito de la figura 2.6.1
- 2.- Completa la tabla de verdad de la figura 2.6.1

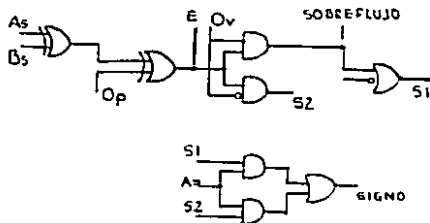


Figura 2.6.1

Entradas

Salidas

As Bs Cs Dv E

D E1 E2 Sigma Sobreflujo

0	0	0	0	1
0	1	0	1	1
0	1	0	0	1
0	1	0	1	0
1	0	0	1	1
1	0	0	0	1
1	0	0	1	0
1	1	0	0	1
0	0	1	1	1
0	0	1	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	1	1
1	1	1	1	1
1	1	1	0	1
1	1	1	1	0

Tabla 2.6.1

PREGUNTAS

1.- Observe la tabla que obtuviste y explica que información nos proporciona.

2.- Explica para que nos sirven las señales E, S1, S2 y Sobreflujo.

3.- Realiza las siguientes operaciones para que puedas comprender las respuestas anteriores.

1) $(A + B)$ $A = 0101$ $B = 0101$

2) $(+A) + (-B)$ $A = 0101$ $B = 0100$

3) $(-A) + (+B)$ $A = 0100$ $B = 0101$

4) $(A - B)$ $A = 0101$ $B = 0100$

5) $(+A) - (-B)$ $A = 0010$ $B = 0011$

6) $(-A) - (+B)$ $A = 0110$ $B = 0110$

7) $(-A) - (-B)$ $A = 0001$ $B = 0001$

8) $(-A) + (-B)$ $A = 0011$ $B = 0010$

RESPUESTAS

- 1.- La tabla que se obtuvo corresponde a las 8 condiciones que se presentarán cuando vamos a realizar las operaciones de suma y resta con signo, generándose 16 operaciones con características diferentes. Las señales que se generan nos van a proporcionar la información necesaria para saber que operación va a realizar el circuito así como el signo del resultado.
- 2.- E -- Determina si se realiza una suma y resta.
- S1-- El resultado está correcto y el signo corresponde al de As.
- S2-- El resultado deberá de complementarse así como el signo de As que va a ser el del resultado.
- Sobreflujo-- Cuando es 1 lo tomamos en cuenta.
Cuando es 0 no lo tomamos en cuenta.

3.-

Operación	Resultado	Señal que se genera
1	1 0000	S1
2	1 0001	Sobreflujo=0 y S1
3	1001	S2
4	0001	S1
5	0101	S1
6	1 0010	S1
7	1 0000	Sobreflujo=0
8	0100	S1

2.7 PRACTICA No. 8

OBJETIVO:

Armar el circuito aritmético completo.

Este circuito procesa los datos según se determine por las señales generadas, se tendrá a la salida el resultado en código BCD.

LISTA DE MATERIAL

- Proto
- Fuente de alimentación de 5V dc.
- Multímetro.
- 1 IC TTL SN7432
- 2 IC TTL SN74157
- 1 IC TTL SN7483
- 1 IC TTL SN7404
- Práctica No. 5, 6 y 7
- 5 Resistencias de 330 Ohms.
- 5 Led's
- Alambres para conexión.

PROCEDIMIENTO

- 1.- Arma el circuito de la figura 2.7.1 y completa la tabla 2.7.1

Entradas									Salidas					
A ₈	A ₇	A ₆	A ₅	A ₄	B ₃	B ₂	B ₁	B ₀	Or	Sigmo	R ₁	R ₂	R ₃	R ₄
0	0	1	0	1	0	0	1	0	1	0				
0	0	1	0	1	1	0	1	0	0	0				
1	0	1	0	0	0	0	1	0	1	0				
1	0	0	1	1	1	0	0	1	0	0				
0	0	1	0	1	0	0	1	0	0	1				
1	0	1	1	0	0	0	1	1	0	1				
0	0	0	1	0	1	0	0	1	1	1				
1	0	0	0	1	1	0	0	0	1	1				

Tabla 2.7.1

CIRCUITO ARITMETICO

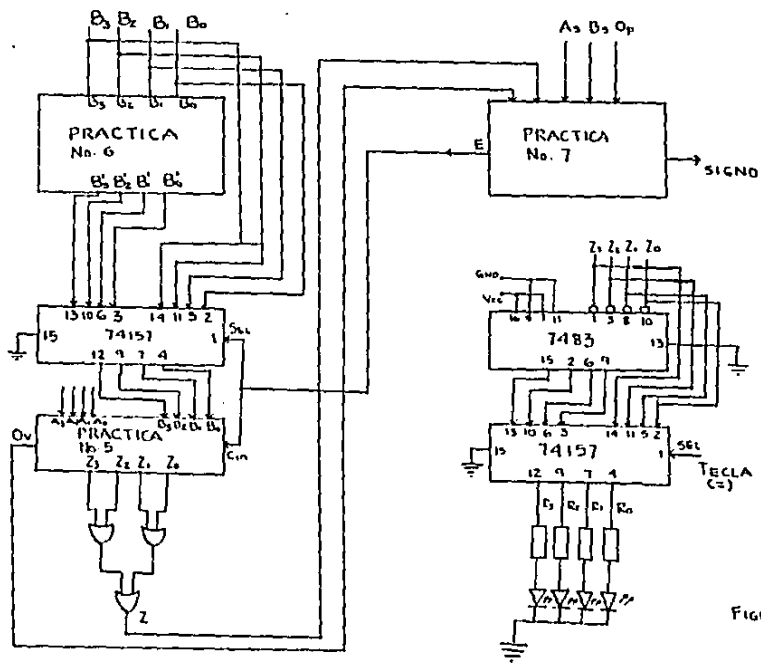


FIGURA 2.7.1

CAPITULO III
"Circuito de Salida".

Los sistemas digitales necesitan de comunicarse con el mundo exterior es por eso que hay dispositivos de entrada y dispositivos de salida, como se mencionó en el primer capítulo los dispositivos de entrada introducen la información que el usuario proporciona para su procesamiento, mientras que los dispositivos de salida nos muestran los resultados obtenidos al procesar dicha información.

Los dispositivos de salida mas comunes son un monitor, una impresora, etcétera.

Para nuestro diseño utilizaremos como dispositivo de salida tres display de cátodo común.

El circuito de salida lo vamos dividir en tres partes para su análisis.

- Circuito multiplexor.
- Circuito decodificador.
- Despliegado de información.

El circuito de salida tiene como función mostrar en los display el dígito correspondiente a la tecla que se presionó así como su respectivo signo.

También nos mostrará el resultado de la operación que realizó. El control de que es lo que está desplegando será llevado con la tecla de igual.

A continuación se explica cada uno de los circuitos que forman en conjunto el circuito de salida.

43

3.1 CIRCUITO MULTIFLEXOR

Los multiplexores generalmente enrutan datos desde una de varias fuentes a un destino.

Un multiplexor es un circuito lógico que acepta varias entradas de datos y permite que sólo una de ellas pase a un tiempo a la salida. El enrutamiento de la entrada deseada de datos hacia la salida está controlada por las entradas de selección.

Para nuestro circuito vamos a ocupar de un multiplexor ya que necesitamos de mostrar tanto los dígitos para los operaciones así como el resultado, así que un multiplexor de dos canales de cuatro bits nos sirve para lograr nuestro objetivo.

El circuito que se va a ocupar es el de la figura 3.1.1

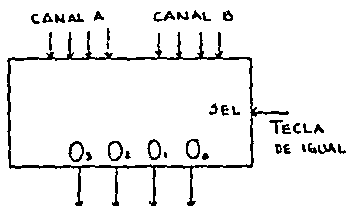


Fig. 3.1.1

En el canal A colocamos las salidas del codificador y en el canal B el resultado de las operaciones.

Cuando la línea de selección está en 0, la información del canal A pasa a la salida del multiplexor que corresponde a los dígitos a ser procesados.

Cuando la línea de selección está en 1 vamos a tener a la salida del multiplexor el resultado de la operación que se realizó.

La línea de selección del multiplexor lo vamos a controlar con la tecla del signo igual.

3.2 PRACTICA No. 9

OBJETIVO:

Comprender el uso de multiplexores en el manejo de información.

LISTA DE MATERIAL

- Proto
- Fuente de alimentación de 5V dc.
- Multímetro.
- 1 IC TTL SN74157
- 4 Resistencias de 330 Ohms.
- 4 Led's
- Alambres para conexión.

PROCEDIMIENTO

- 1.- Arma el circuito de la figura 3.2.1
- 2.- Para las siguientes entradas de A y B y la línea de selección, complete la tabla 3.2.1.

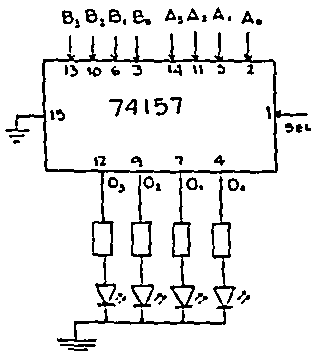


Fig. 3.2.1

Entradas			Salidas
<u>A₁ A₂ A₃ A₄</u>	<u>B₁ B₂ B₃ B₄</u>	<u>Sol</u>	<u>O₁ O₂ O₃ O₄</u>
0 0 0 0	1 1 1 1	1	
1 1 0 0	1 1 1 1	0	
1 0 1 0	0 1 0 1	0	
0 0 1 0	1 0 0 0	1	

Tabla 3.2.1

PREGUNTAS

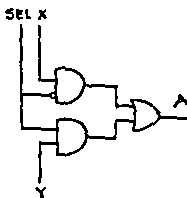
- 1.- Que es un multiplexor?
- 2.- Diseñe un multiplexor con dos entradas y una línea de selección.
- 3.- Explica el funcionamiento del IC 74157.
- 4.- Mencione algunas aplicaciones de multiplexores.

RESPUESTAS

1.- Un multiplexor es un circuito combinatorial que selecciona información de una de muchas entradas para dirigirla a una sola línea de salida.

2.-

Entradas			Salida
x	y	Señ	A
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1



3.- El IC 74157 es un multiplexor de dos canales de cuatro bits tiene una línea de selección que cuando está en 0 selecciona el canal A y cuando está en 1 selecciona el canal B.

- 4.-
- Selección de datos.
 - Enrutamiento de datos.
 - Ordenamiento de operaciones.
 - Generador de formas de onda.
 - Generador de funciones lógicas.
 - Conversión paralelo serie.

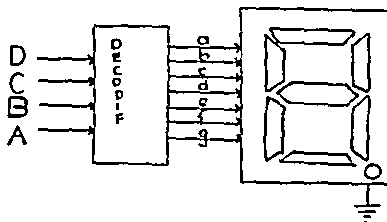
3.3 CIRCUITO DECODIFICADOR.

Un decodificador se puede considerar como dispositivo de salida, si su función es transformar la información de un código de entrada a una salida donde dicha información sea entendida por el usuario.

Un decodificador es un circuito lógico que convierte un código de entrada de n bits en M líneas de salida tales que cada línea será activada para una sola de las combinaciones de entrada.

Para nuestro diseño ocupamos un decodificador BCD a 7 segmentos, éste se usa para tomar una entrada BCD de 4 bits y desplegar un dígito decimal.

El circuito decodificador que estamos utilizando es el siguiente:



Este circuito toma la señal de entrada de un registro de 4-bits en donde se encuentre la información que va a estar presente en el display mientras no se realice ninguna operación.

La salida del decodificador va a un display de cátodo común.

PRACTICA No. 10.

OBJETIVO:

Entender el funcionamiento del decodificador como dispositivo de salida.

LISTA DE MATERIAL:

- Proto.
- Fuente de alimentación de 5V dc.
- Multímetro.
- 1 IC TTL SN74175.
- 1 IC TTL SN7448.
- 1 Display de cátodo común.
- Alambres para conexión.

PROCEDIMIENTO:

- 1.- Arma el circuito de la figura 3.4.1
- 2.- Obtén su tabla de verdad.

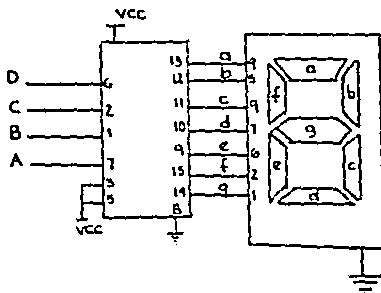


Fig. 3.4.1

Tabla de verdad.

Entrada				Salida
P	C	B	A	Número decimal.
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	

Tabla 3.4.1

PREGUNTAS.

1.- Por qué utilizamos el IC 7448 y no el 7447 ?

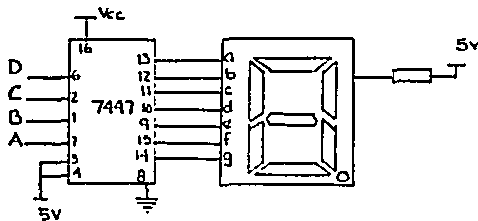
2.- Dibuja las conexiones para el IC 7447 y un display de ánodo-común.

3.- Qué aplicaciones tienen los decodificadores ?

RESPUESTAS:

1.- Porque el IC 7447 es un decodificador diseñado para manejar display de ánodo común, y el display que estamos usando es de cátodo común.

2.-



3.- Para direccionar diferentes dispositivos en un sistema.
Como dispositivo de salida alfanumérico.

DESPLIEGADO DE INFORMACION.

Para desplegar la información estamos usando 3 display uno para el dígito que estamos manejando, para este ocupamos del -- decodificador.

El segundo display es para desplegar el segundo dígito que siempre será uno, porque la operación que nos da el resultado -- más grande es la suma de $9 + 9 = 18$, por lo tanto no se ocupa de un decodificador.

Estamos utilizando la señal de sobreflujo que se explicó en el circuito anterior para manejar este display, recordemos que:

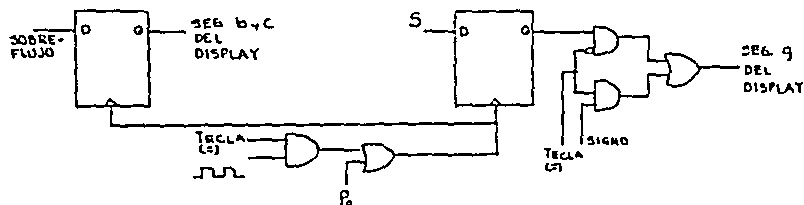
$$\text{Sobreflujo} = 2'0v.$$

Cuando el sobreflujo = 1 hay que desplegar el número 1 por lo que esta señal va a la entrada de un flip-flop para que sea -- desplegada al mismo tiempo que el resultado. La salida del flip-flop va conectada a los segmentos b y c del display.

El tercer display es para el signo, también tenemos una señal llamada signo que es la que nos indica el valor del signo -- del resultado. Hay un flip-flop que almacena el valor del signo del dígito presente que también hay que desplegarlo por lo que -- hay que hacer un arreglo para mostrar ya sea el signo del dígito de entrada o el del resultado.

La salida de éste arreglo va al segmento g del tercer display.

Enseguida mostramos los circuitos que van hacia los display para desplegar el número 1 y el signo.



3.6 PRACTICA No. 11

OBJETIVO:

Armar el circuito de salida, integrando las --- prácticas 9 y 10. Este circuito acepta como entrada datos en código BCD, donde serán decodificados y enviados a display.

LISTA DE MATERIAL:

- Proto.
- Fuente de alimentación de 5V dc.
- Multímetro.
- Prácticas 9 y 10.
- 1 IC TTL SN74175.
- 1 IC TTL SN7474.
- 2 Display de cátodo común.
- Alambres de conexión.

PROCEDIMIENTO:

- 1.- Arma el circuito de salida que corresponde al de la figura - 3.6.1
- 2.- Prosigue a armar todo el circuito completo del calculador en la siguiente práctica.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

CIRCUITO
DE
SALIDA

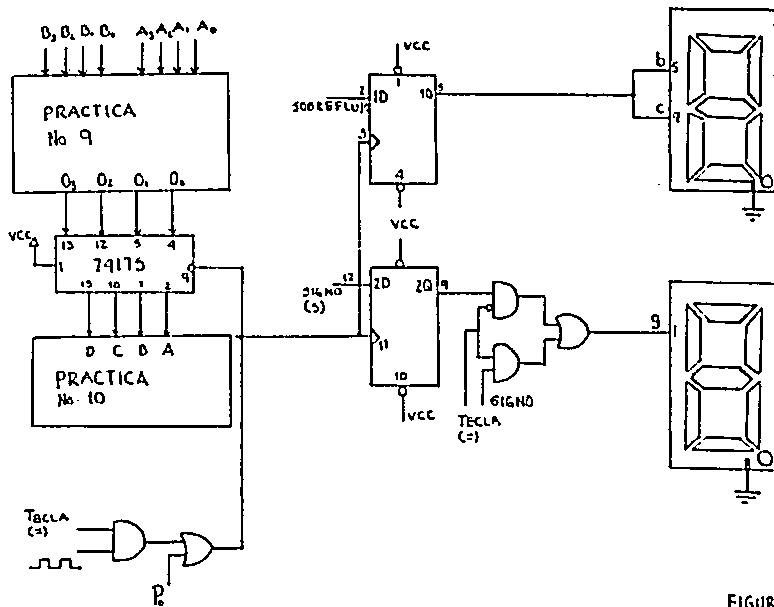


FIGURA 36.1

3.7 PRACTICA No 12

Esta práctica integra los circuitos de entrada, el aritmético y el de salida.

El circuito de entrada nos proporciona la información codificada para ser procesada por la sección aritmética que nos va a entregar los resultados obtenidos, ésta información todavía está en código BCD.

El diagrama eléctrico se muestra en la figura 3.7.1

OBJETIVO:

Arsar el calculador y realizar las operaciones de suma y resta de números con signo.

LISTA DE MATERIAL:

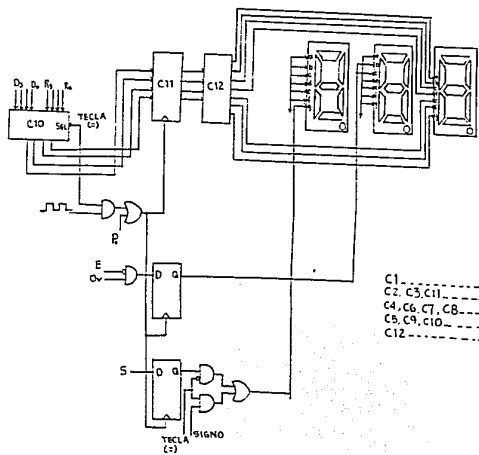
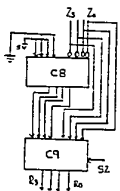
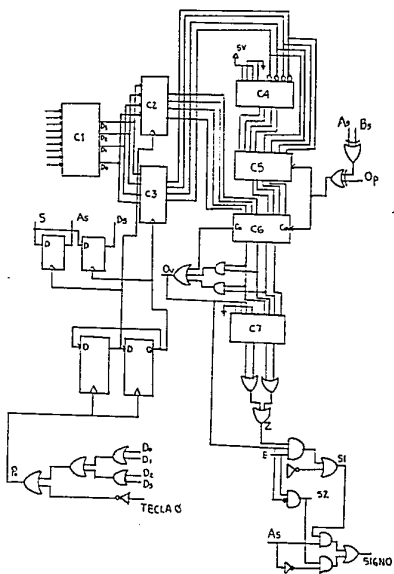
- Multímetro.
- Fuente de alimentación de 5V dc.
- Alambres de conexión.

PROCEDIMIENTO:

- 1.- De acuerdo a la figura 3.7.2 realice las conexiones que se indican.
- 2.- Efectúe las siguientes operaciones y anote el resultado.

Valor de A	Valor de B	Operación	Resultado
5	4	Suma	
-5	6	Suma	
8	-2	Suma	
-7	-3	Suma	
4	9	Resta	
1	1	Resta	
3	2	Resta	
-8	-5	Resta	

DIAGRAMA
ELECTRICO



- C1----- 74147
- C2, C3, C11----- 74175
- C4, C6, C7, C8----- 7483
- C9, C9, C10----- 74157
- C12----- 7448

FIGURA 3.7.1

DIAGRAMA DE BLOQUES DEL CALCULADOR.

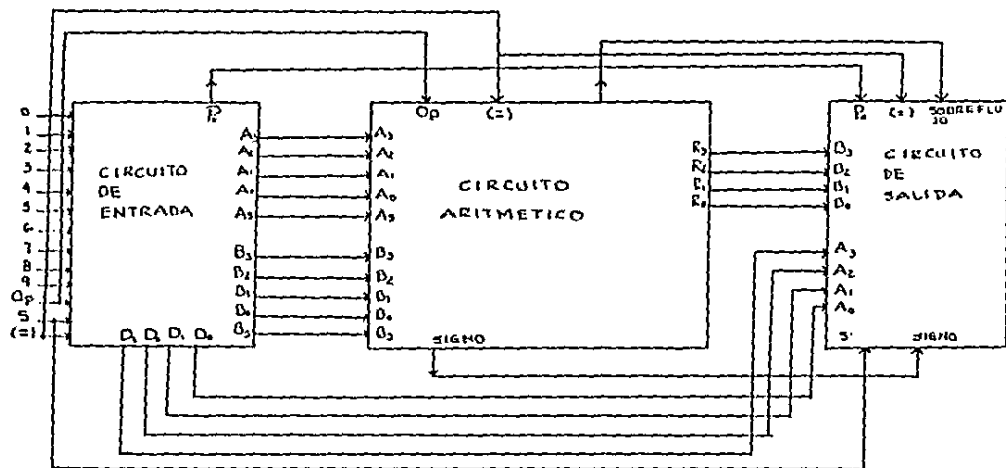


FIGURA 3.7.2

CONCLUSIONES:

Al realizar este tesis se ha logrado ampliar tanto los conocimientos así como el criterio para diseñar.

Al conocer más a fondo un tema de como resultale el querer-saber más, contestando un sin número de interrogantes que surgen al ir realizando los diferentes objetivos propuestos. Es altamente gratificante comprobar con la práctica algo que se conocía en teoría logrando así conocimientos más firmes y completos. Aún la sencillez o la complejidad de un proyecto dejó ver el interés por lograr una finalidad.

Por medio de éstas prácticas que están enfocadas a que el alumno tenga conocimiento de cada dispositivo utilizado para diseñar cualquier sistema por más sencillo que sea, como lo fué el calculador que se armó, ya que son fundamentales éstos conocimientos para iniciarse en el campo del diseño de sistemas digitales.

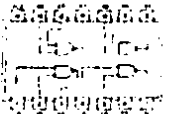
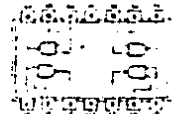
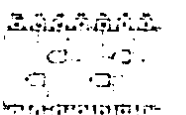
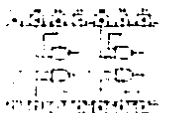
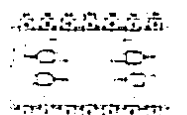
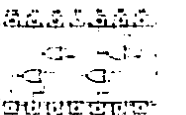
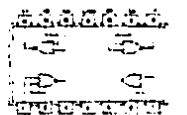
Se espera haber cumplido el principal objetivo de ésta tesis de integrar la parte teórica y práctica en el laboratorio, así como de despertar el interés por este campo del diseño que cada día cobra más importancia.

Lo anterior también tiene como resultado una fuerte motivación y estímulo para quien lo realiza a continuar preparándose en estudios superiores.

A P E N D I C E

54/74 FAMILIES OF COMPATIBLE TTL CIRCUITS

PIN ASSIGNMENTS (TOP VIEWS)

<p>QUADRUPLE 2-INPUT POSITIVE-AND GATES 00</p> <p>Package Type P - 48</p> <p>See page 8-2</p>	 <p>7400 (00) 74700 (00, N1) 7404 (02) 74704 (02, N1) 7401 (01) 74701 (01, N1) 7402 (03) 74702 (03, N1) 7403 (04) 74703 (04, N1)</p>	 <p>7401 (01) 7404 (02) 7402 (03)</p>
<p>QUADRUPLE 2-INPUT POSITIVE-AND GATES WITH OPEN-COLLECTOR OUTPUTS 01</p> <p>Package Type P - 48</p> <p>See page 8-4</p>	 <p>7401 (01) 74701 (01, N1) 7402 (03) 74702 (03, N1)</p>  <p>7402 (03) 74702 (03, N1)</p>	 <p>7401 (01) 7402 (03)</p>
<p>QUADRUPLE 2-INPUT POSITIVE-NOR GATES 02</p> <p>Package Type P - 48</p> <p>See page 8-6</p>	 <p>7402 (03) 74702 (03, N1) 7403 (04) 74703 (04, N1) 7401 (01) 74701 (01, N1) 7404 (02) 74704 (02, N1)</p>	 <p>7402 (03) 7403 (04)</p>

54/74 FAMILIES OF COMPATIBLE TTL CIRCUITS

PIN ASSIGNMENTS (TOP VIEWS)

<p>HEX BUFFER/DRIVERS WITH OPEN COLLECTOR HIGH VOLTAGE OUTPUTS</p> <p>07</p> <p>direction logic V = A</p> <p>See page 6-24</p>	<p>EN1407 (2, NI) EN1407 (12, NI)</p>
<p>QUADRUPLE 2 INPUT POSITIVE AND GATES</p> <p>08</p> <p>direction logic V = A</p> <p>See page 6-10</p>	<p>EN1408 (2, NI) EN1408 (12, NI) EN1408 (8, NI) EN1408 (10, NI) EN1408 (6, NI) EN1408 (14, NI)</p>
<p>QUADRUPLE 3 INPUT POSITIVE AND GATES, WITH OPEN COLLECTOR OUTPUTS</p> <p>09</p> <p>direction logic V = A</p> <p>See page 6-12</p>	<p>EN1409 (2, NI) EN1409 (12, NI) EN1409 (8, NI) EN1409 (10, NI) EN1409 (6, NI) EN1409 (14, NI)</p>
<p>TRIPLE 2 INPUT POSITIVE NAND GATES</p> <p>10</p> <p>direction logic V = A</p> <p>See page 6-2</p>	<p>EN1410 (2, NI) EN1410 (12, NI) EN1410 (10, NI) EN1410 (8, NI) EN1410 (14, NI) EN1410 (6, NI) EN1410 (4, NI) EN1410 (16, NI)</p>

54/74 FAMILIES OF COMPATIBLE TTL CIRCUITS

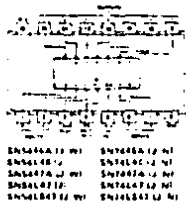
PIN ASSIGNMENTS (TOP VIEW)

BCD TO DEVENEMENT DECODERS/DRIVERS

46 ACTIVE LOW, OPEN-COLLECTOR 20V OUTPUTS

47 ACTIVE LOW, OPEN-COLLECTOR 15V OUTPUTS

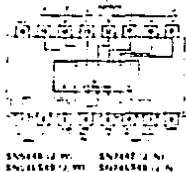
See page 7-22



BCD TO DECREMENTAL DECODERS/DRIVERS

48 INTERNAL PULL UP OUTPUT

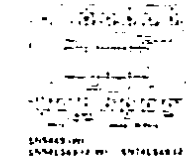
See page 7-22



BCD TO DECREMENTAL DECODERS/DRIVERS

49 OPEN-COLLECTOR OUTPUTS

See page 7-22

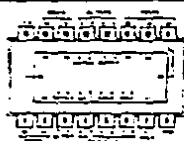


54/74 FAMILIES OF COMPATIBLE TTL CIRCUITS

PIN ASSIGNMENTS (TOP VIEWS)

4-BIT MAGNITUDE COMPARATORS

85



See Also 753

SN7485-12 (N) SN7485-12 (N)
SN7485-12 (N) SN7485-12 (N)
SN7485-12 (N) SN7485-12 (N)

SN7485-12 (N) SN7485-12 (N)

QUADRUPLE 2-INPUT EXCLUSIVE-OR GATES

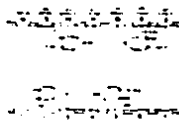
86

$A \oplus B = A \bar{B} + \bar{A} B$

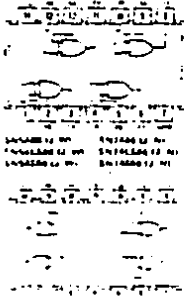
FUNCTION TABLE

INPUTS		OUTPUT
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

See Also 745



SN7486-12 (N) SN7486-12 (N)



SN7486-12 (N) SN7486-12 (N)
SN7486-12 (N) SN7486-12 (N)
SN7486-12 (N) SN7486-12 (N)

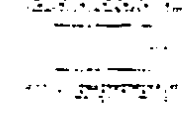
SN7486-12 (N)

4-BIT 2-LEVEL COMPLEMENT ZERO ONE ELEMENTS

87

INPUTS		OUTPUTS	
A	B	\bar{A}	\bar{B}
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0	0

See Also 779



SN7487-12 (N) SN7487-12 (N)
SN7487-12 (N) SN7487-12 (N)

TTL
MSI

TYPES SN5483A, SN54LS83A, SN7483A, SN74LS83A
4-BIT BINARY FULL ADDERS WITH FAST CARRY

LSI DESIGN CENTER, SANTA CLARA, CALIF. 95051, 1978, REVISED OCTOBER 1978

- Full Carry Look-Ahead across the Four Bits
- Systems Achieve Partial Look-Ahead Performance with the Economy of Ripple Carry
- SN54283/SN74283 and SN54LS283/SN74LS283 Are Recommended For New Designs as They Feature Supply Voltage and Ground on Corner Pins to Simplify Board Layout

TYPE	TYPICAL ADD TIME		TYPICAL POWER DISSIPATION PER 4-BIT ADDER
	2ND 8BIT WORDS	3RD 16BIT WORDS	
SN	21ns	41ns	30mw
LS83A	27ns	45ns	40mw

Description

These improved full adders perform the addition of two 4-bit binary numbers. The seven (2³) outputs are provided for each bit and the resultant carry (C₄) is obtained from the fourth bit. These adders feature full internal look-ahead across all four bits operating the carry term in ten nanoseconds typically. This provides the system designer with detailed timing performance at the bit level and reduced average count of propagation delay variation.

The carry logic including the carry is implemented in a true form meaning that the end-around carry can be accomplished without the need for a carry-in flip-flop.

Designed for medium speed applications, the units comply with the JEDEC 10B test that is compatible with most other TTL families and power calculated from the test families.

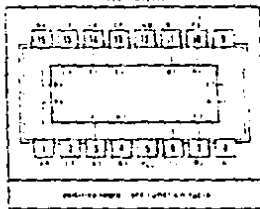
Series EA and LS series units are characterized for operation over the full military temperature range of -55°C to 125°C and series TA and 74LS series are characterized for operation from 0°C to 75°C.

absolute maximum ratings over operating free air temperature range (unless otherwise noted)

Supply voltage (V _{CC}) (see Note 1)		7 V
Input voltage (V _I)		5.5 V
LS83A		7 V
Operating temperature (see Note 2)		55 °C
Operating free air temperature range	SN54283A SN74283A	-55°C to 125°C
	SN7483A SN74LS83A	0°C to 75°C
Storage temperature range		-65°C to 150°C

NOTES: 1. Voltage values are with respect to ground and are in the positive or negative direction.
2. This is the storage temperature range unless a different temperature is specified. The testing temperature for the 283 is a maximum of 75°C and for the 83 and 83A and 83 and 83A is 0°C to 75°C.

SN54283, SN74283A 28-PIN PACKAGE
SN7483A, SN74LS83A 28-PIN PACKAGE
(TOP VIEW)



FUNCTION TABLE

A ₃	A ₂	A ₁	A ₀	C ₀				C ₄
				0	1	0	1	
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	0
0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	1	0	0
0	1	0	1	1	0	0	0	0
0	1	1	0	1	1	0	0	0
0	1	1	1	1	1	1	0	0
1	0	0	0	1	1	1	1	0
1	0	0	1	1	1	1	1	0
1	0	1	0	1	1	1	1	0
1	0	1	1	1	1	1	1	0
1	1	0	0	1	1	1	1	0
1	1	0	1	1	1	1	1	0
1	1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	1	0

NOTES: 1. C₀ = 0 for all cases.
2. C₀ = 1 for all cases.
3. C₀ = 0 for all cases.
4. C₀ = 1 for all cases.
5. C₀ = 0 for all cases.
6. C₀ = 1 for all cases.
7. C₀ = 0 for all cases.
8. C₀ = 1 for all cases.

TTL
MSI

TYPES SN54147, SN54148, SN54LS147, SN54LS148,
SN74147, SN74148 (TIM9907), SN74LS147, SN74LS148
10-LINE-TO-4-LINE AND 6-LINE-TO-3-LINE PRIORITY ENCODERS

Full Data Sheet: DLS9701 (2) (1-00000-1000)

147, 148

- Encodes 10-Line Decimal to 4-Line BCD

- Applications Include:
Keyboard Encoding
Range Selection

148, 148A

- Encodes 6 Data Lines to 3-Line Binary (Octal)

- Applications Include

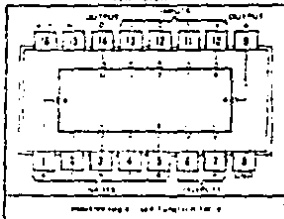
N Bit Encoding
Code Converters and Generators

	TYPICAL TYPE	TYPICAL DATA SUSCEPTIBILITY	TYPICAL POWER DISSIPATION
147	10	100	20mW
148	10	100	20mW
147A	10	100	20mW
148A	10	100	20mW

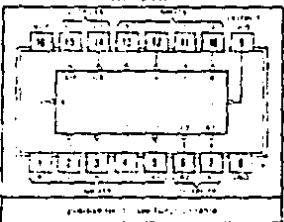
Description

The TTL encoders feature priority decoding of the inputs to produce the binary code for the 4-line BCD or the 3-line binary code. The 10-line-to-4-line decoder (147) and the 6-line-to-3-line decoder (148) are the most popular TTL MSI 10-bit devices available. The 10-line-to-4-line decoder (147) is used where a 4-bit binary code is required. The 6-line-to-3-line decoder (148) is used where a 3-bit binary code is required. The 10-line-to-4-line decoder (147) has been designed to allow data expansion without the need for external circuitry. The 6-line-to-3-line decoder (148) is used where a 3-bit binary code is required. The 10-line-to-4-line decoder (147) and the 6-line-to-3-line decoder (148) are the most popular TTL MSI 10-bit devices available.

10-LINE-TO-4-LINE PRIORITY ENCODER
SN54147, SN74147 (DIP PACKAGE)
TOP VIEW



6-LINE-TO-3-LINE PRIORITY ENCODER
SN54148, SN74148 (DIP PACKAGE)
TOP VIEW



147, 148A

FUNCTION TABLE

INPUTS	OUTPUTS												
A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	1	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	1	1	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	1	0	0	0	0	0	0
0	0	1	1	1	0	1	0	0	0	0	0	0	0
0	0	1	1	1	0	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	1	0	0	0	0	0	0	0
0	1	0	0	1	0	1	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	1	0	0	0	0	0	0
0	1	0	1	0	0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	1	1	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0	0	0	0	0
0	1	0	1	1	0	0	1	0	0	0	0	0	0
0	1	0	1	1	0	1	0	0	0	0	0	0	0
0	1	0	1	1	0	1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0	0	0	0
0	1	1	0	0	0	1	1	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	1	0	0	0	0	0	0
0	1	1	0	1	0	1	0	0	0	0	0	0	0
0	1	1	0	1	0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
0	1	1	1	0	0	1	1	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	1	0	0	0	0	0	0
0	1	1	1	1	0	1	0	0	0	0	0	0	0
0	1	1	1	1	0	1	1	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	1	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	1	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	1	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	1	0	0	0	0	0	0
1	0	0	1	1	0	1	0	0	0	0	0	0	0
1	0	0	1	1	0	1	1	0	0	0	0	0	0
1	0	0	1	1	1	0	0	0	0	0	0	0	0
1	0	0	1	1	1	0	1	0	0	0	0	0	0
1	0	0	1	1	1	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	1	1	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	1	0	0	0	0	0	0
1	0	1	0	1	0	1	0	0	0	0	0	0	0
1	0	1	0	1	0	1	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0	0	0
1	0	1	1	0	0	1	0	0	0	0	0	0	0
1	0	1	1	0	0	1	1	0	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	0	1	0	0	0	0	0	0
1	0	1	1	1	0	1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1	0	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0	0	0	0	0	0
1	0	1	1	1	1	1	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	1	1	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	1	0	0	0	0	0	0
1	1	0	0	1	0	1	0	0	0	0	0	0	0
1	1	0	0	1	0	1	1	0	0	0	0	0	0
1	1	0	0	1	1	0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	1	0	0	0	0	0	0
1	1	0	0	1	1	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0							

**TYPES SN54174, SN54175, SN54LS174, SN54LS175, SN54S174, SN54S175,
SN74174, SN74175, SN74LS174, SN74LS175, SN74S174, SN74S175**
HEX/QUADRUPLE D-TYPE FLIP-FLOPS WITH CLEAR

'174, 'LS174, 'S174... HEX D-TYPE FLIP-FLOPS
'175, 'LS175, 'S175... QUADRUPLE D-TYPE FLIP-FLOPS

- '174, 'LS174, 'S174 Contain Six Flip-Flops with Single-Rail Outputs
- '175, 'LS175, 'S175 Contain Four Flip-Flops with Double-Rail Outputs
- Three Performance Ranges Offered See Table Lower Right
- Buffered Clock and Direct Clear Inputs
- Individual Data Input to Each Flip-Flop
- Applications include Buffer/Storage Registers Shift Registers Pattern Generators

DESCRIPTION

These series of complementary pair-type flip-flops utilize TTL circuitry to implement D-type flip-flop logic. All have a direct clear input and the '175, 'LS175, and 'S175 feature complementary outputs from each flip-flop.

Information at the D inputs meets the setup time requirements as transferred to the Q outputs on the positive-going edge of the clock pulse. Clock frequency occurs at a particular voltage level and is not necessarily related to the transition time of the test set-up logic when the clock input is at either the high or low level, the D input signal has no effect at the output.

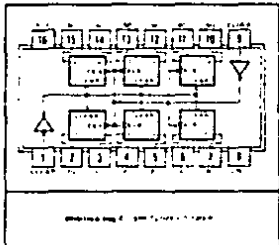
These outputs are fully compatible for use with most TTL or DTL outputs.

FUNCTION TABLE
LEACH BIPOLAR

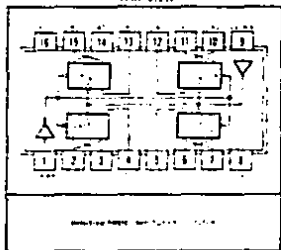
INPUTS		OUTPUTS	
CLEAR	CLOCK	Q	Q'
L	X	X	X
X	F	D	D'
X	L	L	H
X	H	H	L

- X = unknown state, L = low level, H = high level, F = fast transition
- L = low level, H = high level
- D = data
- D' = complement of D
- L = low level, H = high level
- L = low level, H = high level
- L = low level, H = high level
- L = low level, H = high level

SN54174 SN54LS174 SN54S174 JDR PACKAGE
SN74174 SN74LS174 SN74S174 JDR PACKAGE
(TOP VIEW)



SN54175 SN54LS175 SN54S175 JDR PACKAGE
SN74175 SN74LS175 SN74S175 JDR PACKAGE
(TOP VIEW)



TYPES	USUAL MAXIMUM CLOCK FREQUENCY	USUAL POWER DISSIPATION PER FLIP-FLOP
'174, '175	20 MHz	26 mW
'LS174, 'LS175	33 MHz	16 mW
'S174, 'S175	110 MHz	70 mW

TEXAS INSTRUMENTS
POST OFFICE BOX 5008 DALLAS, TEXAS 75222

172

TYPES SN54157, SN54L157, SN54LS157, SN54LS158, SN54S157, SN54S158, SN74157, SN74L157, SN74LS157, SN74LS158, SN74S157, SN74S158
QUADRUPLE 2-LINE-TO-1-LINE DATA SELECTORS/MULTIPLEXERS

SN54157, SN54L157, SN54LS157, SN54LS158, SN54S157, SN54S158 ... J-OM PACKAGE
 SN74157, SN74L157, SN74LS157, SN74LS158, SN74S157, SN74S158 ... J-OM PACKAGE
 (TOP VIEW)

features

- Buffered Inputs and Outputs
- Three Speed/Power Ranges Available

TYPES	TYPICAL AVERAGE PROPAGATION TIME	TYPICAL POWER DISSIPATION
'157	9 ns	150 mW
L157	18 ns	75 mW
LS157	9 ns	48 mW
'157	5 ns	250 mW
LS158	7 ns	24 mW
'158	4 ns	195 mW

applications

- Expand Any Data Input Point
- Multiplex Dual Data Buses
- Generate Four Functions of Two Variables (One Variable is Common)
- Source Programmable Counters

description

These monolithic data selectors/multiplexers contain inverters and drivers to supply full on-chip data selection to the four output gates. A separate strobe input, \bar{S} , provides a 5-bit word selection from one of two sources and is routed to the four outputs. The '157, L157, LS157, and 'S157 present true data whereas the LS158 and 'S158 present inverted data to minimize propagation delay time.

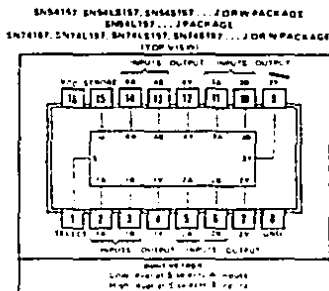
FUNCTION TABLE		OUTPUT Y			
INPUTS		DATA			
STROBE SELECT	A	B	'157, L157, LS157, 'S157	LS158, 'S158	LS158, 'S158
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	0	0
1	1	1	0	1	1

0 = High level (1); 1 = Low level (0)

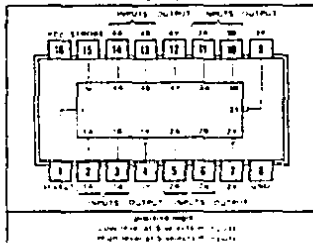
absolute maximum ratings over operating free air temperature range (unless otherwise noted)

Supply voltage, VCC (Type 1)	7 V
Input voltage, '157, L157, 'S158	5.5 V
LS157, LS158	7 V
Operating free air temperature range	0°C to 70°C (SN54, SN54S, SN54LS, SN54S Circuits) -55°C to 125°C (SN74, SN74L, SN74LS, SN74S Circuits)
Storage temperature range	0°C to 100°C

NOTE 1: See typical characteristics graph for recommended ground return.



SN54LS158, SN54S158 ... J-OM PACKAGE
 SN74LS158, SN74S158 ... J-OM PACKAGE
 (TOP VIEW)



BIBLIOGRAFIA

- Taub Herbert. Circuitos digitales y microprocesadores. Edo. de México, México: McGraw Hill, 1982.
- The engincering staff of Texas Instruments, Inc. TTL data -- book for desing enginners. Dallas, Texas, U.S.A.: Texas - Instruments Inc., 1981.
- Morris Mano M. Lógica digital y diseño de computadores. Edo. - de México, México: Prentice Hall Hispanoamericana, 1982.
- Mandado Enrique. Sistemas electrónicos digitales. Edo. de México, México: Publicaciones Marcombo, 1983.
- J. Tecci Ronald. Sistemas digitales principios y aplicaciones. Edo. de México, México: Prentice Hall Hispanoamericana, 1981
- Morris Mano M. Arquitectura de computadores. Edo. de México, - México: Prentice Hall Hispanoamericana, 1983.
- Terres Hernández Leopoldo. Manual de laboratorio de diseño - lógico. Guadalajara, Jal., México.: Universidad Autónoma de L. Guadalajara, 1987.