



Universidad De Oriente Núcleo De Anzoátegui  
Escuela De Ingeniería Y Ciencias Aplicadas  
Departamento De Computación Y Sistemas  
Lab De Diseño Lógico Código 0724531

Profesor: Dalvin Falcón.

## Pre-informe Práctica #5: Lógica Combinacional MSI. MUXes. Decoders

### INTRODUCCIÓN

En computación, la unidad aritmético lógica o ALU es un circuito digital que calcula operaciones aritméticas y operaciones lógicas entre dos números, a la que se le atribuyen ventajas como bajos costos, más flexibles, mayor facilidad de diseño y capacidad de realizar tareas muy complejas.

Muchos tipos de circuitos electrónicos necesitan realizar algún tipo de operación aritmética, así que incluso el circuito dentro de un reloj digital tendrá una ALU minúscula que se mantiene sumando 1 al tiempo actual, y se mantiene comprobando si debe activar el sonido de la alarma. Por mucho, los más complejos circuitos electrónicos son los que están contruidos dentro de los chips de microprocesadores modernos. Por lo tanto, estos procesadores tienen dentro de ellos un ALU muy complejo y potente. De hecho, un microprocesador moderno y los mainframes puede tener múltiples núcleos, cada núcleo con múltiples unidades de ejecución, cada una de ellas con múltiples ALU.

Partiendo de los trabajos hechos por el matemático John von Neumann, quien propuso el concepto de la ALU en 1945, cuando escribió un informe sobre las fundaciones para un nuevo computador llamado EDVAC y que luego se convertiría en el prototipo para muchos computadores posteriores, esta práctica tendrás como finalidad indagar, describir, diseñar y poner a prueba una ALU simplificada.

### OBJETIVOS

Diseñar e implementar circuitualmente un circuito ALU simplificado combinacional utilizando la técnica de MUX y otros MSI para apoyo. El circuito será capaz de realizar: complemento de A; incremento de A en una unidad; decremento de A en una unidad; rotación a la izquierda de A (a través del acarreo)

### BASES TEÓRICAS

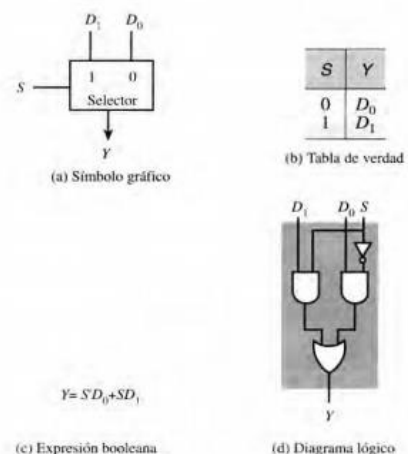
#### 1. MULTIPLEXOR (mux)

Según Morris-Mano, multiplexar significa transmitir una gran cantidad de unidades de información por un número pequeño de canales o líneas. Un multiplexor es un circuito combinacional que selecciona información binaria de una de muchas líneas de entrada y la envía a una sola línea de salida. La selección de una línea de entrada dada se controla con un conjunto de líneas de selección. Normalmente, hay  $2^n$  líneas de entrada y n líneas de selección cuyas combinaciones de bits determinan cuál entrada se selecciona.

Un componente combinacional que permite seleccionar una o varias fuentes como operandos de una ALU, para almacenarlos en una memoria o para transmitirlos a un bus. En general, un selector tendrá n entradas, una salida y  $\log_2 n$  señales de selección que seleccionarán datos desde cualesquiera de las n entradas y las llevarán a la salida. Aunque se podría construir cualquier selector de n a 1, en la práctica la mayoría de las bibliotecas lógicas suelen incluir sólo multiplexores de 2 a 1 y de 4 a 1, simplemente porque los multiplexores con un número mayor de entradas pueden realizarse fácilmente a partir de multiplexores de 2 a 1 y de 4 a 1.

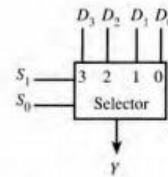
En la Fig. BT 1.1 se muestra un multiplexor de 2 a 1, cuyo símbolo gráfico se presenta en la Fig. BT 1.1 (a) y su tabla de verdad en la Fig. BT 1.1 (b). Como puede verse, este selector de 2 a 1 dispone de dos entradas, D1 y D0, y una salida, Y, además de una señal de selección, S, que selecciona una de las entradas del siguiente modo: siempre que  $S = 0$  la salida  $Y = D_0$ , mientras que

**Fig BT 1.1.** Multiplexor 2 a 1



siempre que  $S = 1$  la salida  $Y = D1$ . La expresión booleana para la salida  $Y$  se indica en la Fig. BT 1.1(c) y el diagrama lógico completo de este selector de 2 a 1 se presenta en la Fig. BT 1.1(d).

En contraposición, la Fig. BT 1.2 muestra un multiplexor de 4 a 1, que selecciona cualesquiera de las cuatro líneas que indiquen las dos señales de selección  $S_1$  y  $S_0$ . En esta figura puede verse el símbolo gráfico, la tabla de verdad, la expresión booleana de salida y el diagrama lógico de este multiplexor de 4 a 1.



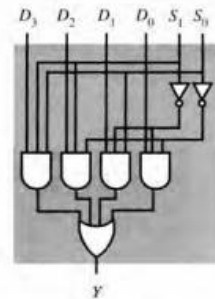
(a) Símbolo gráfico

$S_1$	$S_0$	$Y$
0	0	$D_0$
0	1	$D_1$
1	0	$D_2$
1	1	$D_3$

(b) Tabla de verdad

$$Y = S_1'S_0'D_0 + S_1'S_0D_1 + S_1S_0'D_2 + S_1S_0D_3$$

(c) Expresión booleana



(d) Diagrama lógico

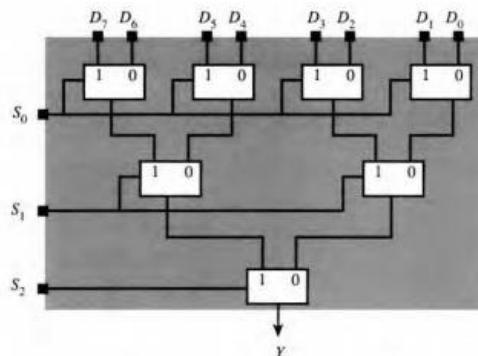
Si es necesario, los multiplexores de mayor tamaño pueden construirse a partir de un número de selectores de 2 a 1. Por ejemplo, para un multiplexor de  $n$  a 1, donde  $n$  es una potencia de 2, se utilizarán  $\log_2 n$  señales de multiplexores y  $\log_2 n$  niveles de multiplexores de 2 a 1. Cada señal de selección servirá para controlar uno de los niveles de multiplexores. En el primer nivel, cada multiplexor seleccionará entre dos fuentes de datos, mientras que en el segundo nivel cada multiplexor seleccionará entre las dos salidas del multiplexor del primer nivel. Es decir, en cada nivel superior al primero, cada multiplexor se utiliza para seleccionar una de las dos salidas producidas por los selectores del nivel anterior. Como ejemplo de multiplexor de mayor tamaño, la Fig. BT 1.3 muestra un selector de 8 a 1, cuya tabla de verdad se indica en la

**Fig. BT 1.2. Multiplexor 4 a 1**

Fig. BT 1.3 (a), mientras que en la Fig. BT 1.3 (b) se muestra una implementación que utiliza selectores de 2 a 1.

$S_2$	$S_1$	$S_0$	$Y$
0	0	0	$D_0$
0	0	1	$D_1$
0	1	0	$D_2$
0	1	1	$D_3$
1	0	0	$D_4$
1	0	1	$D_5$
1	1	0	$D_6$
1	1	1	$D_7$

(a) Tabla de verdad



(b) Implementación con selectores de 2 a 1

**Fig. BT 1.3. Multiplexor 8 a 1 con multiplexores 2 a 1**

Sobre los tipos de multiplexores, los autores consultados difieren en los parámetros de categorización. En algunas fuentes el tipo viene establecido por la cantidad de entradas (8 entradas, 16 entradas, doble 8 entradas, etc), o por la funcionalidad de su composición: digital o analógica. Para este informe se desarrollará la última aproximación que se aprovechará para detallar los dispositivos integrados correspondientes:

## El multiplexor digital

En una definición breve, el multiplexor digital está compuesto por un circuito combinatorio generalmente con compuertas AND o NAND, negadores y una compuerta OR o NOR a la salida.

La Fig BT 1.4 corresponde a un MUX 8:1 comercial, el 74ls151. En la figura puede verse el diagrama de conexiones del integrado y en la Fig BT 1.5 se ve el diagrama interno y la tabla de verdad.

**Fig. BT 1.4** Multiplexor IC 74ls151

Para este dispositivo, se tienen tres entradas de selección: A, B, C más ocho entradas de datos: D0 hasta D7, una salida de datos Y y su negada W y una entrada de control STROBE que es una función adicional al multiplexor. Además, como siempre, se tiene los pines de alimentación, que en general son dos: Vcc y GND para dispositivos TTL y Vdd y Vss para circuitos de tecnología CMOS.

Las entradas D0 a D7 son seleccionadas a través de las líneas de selección A, B y C, donde según el número binario que corresponde a alguna de las 8 combinaciones posibles con tres entradas, dará el acceso a la salida Y ó W, a alguna de ellas. Se considera aquí que C es la línea del bit más significativo ( MSB) y A la línea correspondiente

**Fig. BT 1.5** Diagrama interno del Mux IC74ls151 al bit menos significativo ( LSB).

Se dispone además de una línea de habilitación S denominada Strobe, la cual se habilita con nivel bajo, esto es, cuando  $S = 0$ , la entrada seleccionada pasará a la salida, caso contrario ( $S = 1$ ), las salidas no responderán a las entradas de datos ( $Y = 0$  y  $W = 1$  permanentemente).

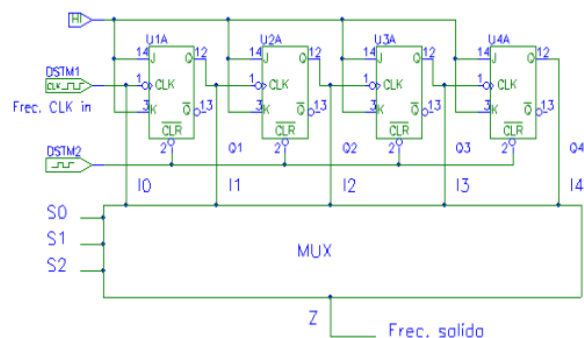
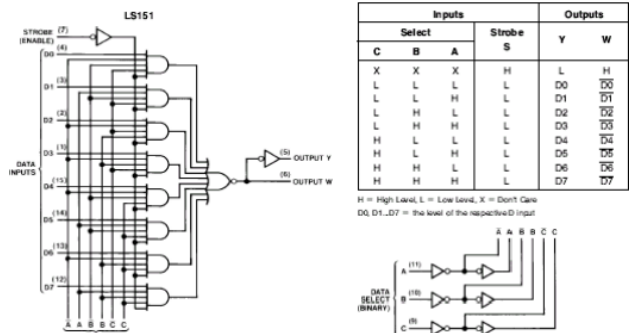
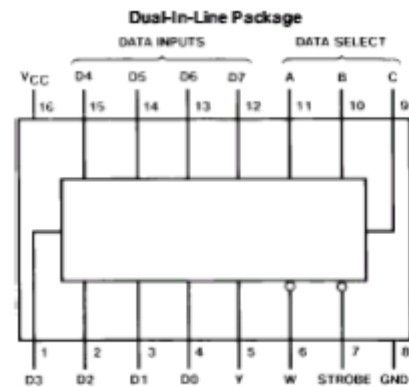
Las dos principales aplicaciones de un MUX son como:

- Selector de fuentes de señal.

Dado que se puede elegir una de entre n entradas posibles, un uso lógico sería el de selector de diversas fuentes de señales digitales para que sola una pueda conectarse con la salida. Véase por ejemplo que se quiera diseñar un divisor de frecuencias de señal de clock, tal que se puedan obtener la señal de reloj de un oscilador de cuarzo y sus submúltiplos en potencia de dos ( $f_{CLK}/2$ ,  $f_{CLK}/4$ ,  $f_{CLK}/8$  y  $f_{CLK}/16$ ).

Empleando un contador binario de 4 bits se puede dada una señal de reloj ( clock), obtener en Q1, Q2, Q3 y Q4, señales de clock pero cuyas frecuencias sean  $f_{CLK}$  dividido 2 ( saliendo de Q1), dividido 4 ( saliendo de Q2), dividido 8 ( de Q3) y dividido 16 ( saliendo de Q4). Para obtener un selector, se puede emplear un MUX 8:1 como el descrito donde se utilizarán sólo 5 entradas: CLK, Q1, Q2, Q3 y Q4.

Según la combinación binaria en las entradas de selección del MUX (en este ejemplo denominadas S0, S1 y S2, en vez de A, B y C), se podrá elegir una de entre las citadas fuentes de señal de clock, que se conectan a las entradas de datos del dispositivo ( aquí las entradas de datos se llamaron I0, I1.....I7 ). La salida Z, copiará la señal del canal seleccionado. El circuito definitivo se muestra en la Fig. BT 1.6.



**Fig. BT 1.6** Selector de señales digitales con Mux 4 a 1

- Generador de funciones booleanas.

Si se analiza con detenimiento el circuito del MUX 74LS151, se puede observar que básicamente se compone de una compuerta OR (considerando la NOR mas el negador para salir por Y) de ocho entradas, donde cada una de esas entradas, proviene, compuerta AND de por medio, de una entrada de datos diferente.

Cada una de esas compuertas AND, tiene además tres entradas que provienen de las líneas de selección: A, B y C, y con una combinación diferente de ellas. Por lo tanto cada combinación de A,B y C, sólo habilitará en principio a una sola compuerta AND de las ocho que hay. Dada una AND habilitada por A, B y C, dependiendo de la entrada de datos en cuestión, será el valor de la salida. Se puede formular, entonces, la siguiente ecuación lógica de este dispositivo:

$$F = /C/B/A D0 + /C/BA D1 + /CB/A D2 + /CBA D3 + C/B/A D4 + C/BA D5 + CB/A D6 + CBA D7$$

Como las entradas de datos pueden valer "0" ó "1", se puede pensar entonces que cuando se selecciona una combinación dada de las entradas de selección, en realidad se está eligiendo un término producto ( un mintérmino) C B A de los  $2^n$  posibles con n entradas de datos.

La salida será "1" ó "0" dependiendo si la entrada correspondiente está a "1" ó a "0". Por lo tanto se puede implementar en principio, cualquier función canónica de primera forma ( unión de mintérminos) de n variables, donde las variables se conecta a las entradas de selección del MUX y las entradas cuyos términos producto intervengan en la función, se pondrán a "1" y el resto a "0". En caso que la función no esté descrita en 1ra forma ( caso más probable), se deberá desarrollar y seguir los pasos antes descriptos.

Como ejemplo, tómesese que se desea, dada la función  $D = /B + /A B /C$ , implementarla con un MUX 74LS151. Dado que la función tiene tres variables, se puede emplear un MUX 8:1 de tres líneas de selección. Como la función no está escrita en primera forma, se puede desarrollar empleando el diagrama de Karnaugh como se ve en la Fig BT 1.7.

Desarrollando a B para que quede en primera forma y sumando el término  $/CB/A$ , queda:

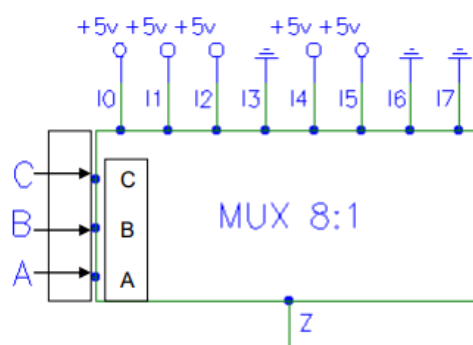
$$F = /C/B/A + /C/BA + C/B/A + C/BA + /CB/A$$

B A					
C		/B/A	/BA	BA	B/A
		1	1		1
	C	1	1		

**Fig BT 1.7** Mapa Kargnaug para generador de funciones de tres variables

La función consta de los mintérminos: 0, 1, 2, 4 y 5, por lo tanto la implementación con el MUX será tal que las entradas D0, D1, D2, D4 y D5 se pondrán a Vcc ( "1" lógico), mientras que las entradas: D3, D6 y D7, se conectarán a GND ( "0" lógico). El circuito definitivo será el mostrado en la Fig BT 1.8

**Fig BT 1.8** Circuito generador de funciones con multiplexor 8:1



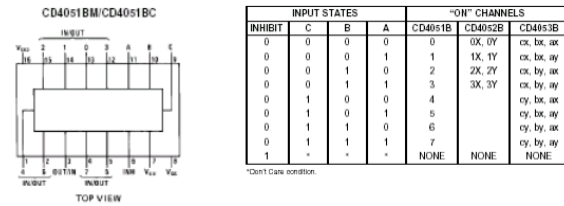
### El multiplexor analógico

El Mux analógico se emplea principalmente como selector de entrada para conectar varias fuentes de tensión analógicas a la entrada de otro dispositivo. Un ejemplo clásico de ello es el del conversor analógico digital. Este dispositivo convierte una señal de tensión en un número binario, cuyo valor es proporcional a la amplitud de dicha señal. Tiene una única entrada, por lo que si se desea convertir varias fuentes de tensión con el mismo conversor, se deberá disponer de un selector como el MUX. Dado que el MUX analógico dispone de entradas de

selección como el digital, es posible comandar vía una interface, el proceso de conversión de datos, seleccionando en forma secuencial los canales necesarios para la adquisición a través de una PC u otro equipo controlador.

En la Fig BT 1.9 se ve un circuito de un multiplexor analógico con tecnología CMOS, el CD4051, MUX 8:1.

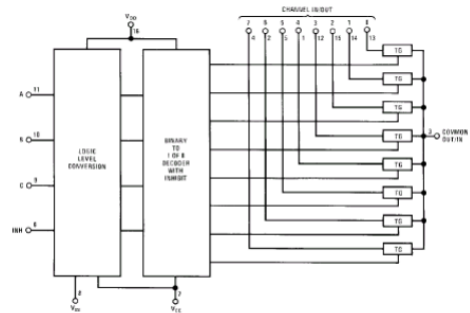
A diferencia del MUX digital, el MUX analógico dispone de una serie de llaves de tensión analógicas bidireccionales construidas con transistores de efecto de campo en configuración antiparalelo, lo cual no sólo se puede trabajar con el MUX de la forma tradicional ( n entradas y una salida) sino también como un DeMUX, es decir, usar la salida como entrada de señal y las entradas como salidas. La selección de cuál llave se habilitará de las n posibles, se realiza con un circuito decodificador similar al empleado en un MUX digital.



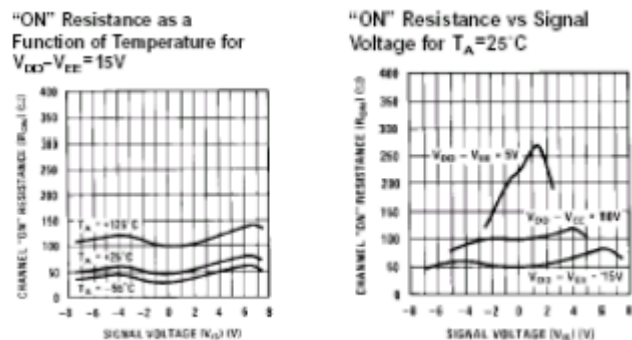
**Fig BT 1.9** IC Mux 8:1 CD4051

Debido a que la idea es trabajar con señales analógicas, los MUX tienen en general doble fuente de alimentación +/-Vdd, de tal manera de poder trabajar con tensiones tanto positivas como negativas.

El CD4051 se puede conectar a una fuente partida de hasta +/- 7.5V. Posee 8 canales de entrada/salida y un canal de salida/entrada. Con las líneas A, B y C se selecciona la entrada/salida que se desea esté activa. Además posee otra entrada de Inhibición que puede ser empleada para el caso de necesitar expandir la cantidad de entradas, utilizando uno ó más MUXs de este tipo.



Una consideración importante a tener en cuenta en los MUX analógicos es que las llaves analógicas distan mucho de ser ideales. Poseen una resistencia eléctrica Ron cuando están activas cuyo valor es de cientos de ohms y una resistencia Roff cuando están desactivadas de algunos Megohms. En el diseño hay que considerar estos valores, además que la Ron varía con la tensión de entrada a la llave, la tensión de alimentación ( $\Delta R_{on}$  de algunas decenas de ohms) y con la temperatura (Fig BT 1.10).



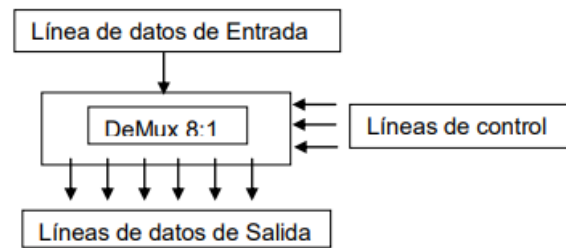
**Fig BT 1.10** Variaciones de las resistencia electrónica Ron y Roff en función de la temperatura

Este dispositivo no dispone de interfaz entre la sección digital de control y la analógica, por lo que si se alimenta con +Vdd al circuito, esa misma tensión es la que se emplea para definir el nivel de tensión de alimentación a las entradas digitales. Esto es importante cuando se desee conectar un circuito TTL como comando del MUX.

Otros autores, añadidos a los tipos ya explicados, categorizan los multiplexores de la siguiente manera: multiplexación inversa (IMUX), multiplexación por división de longitud de onda (WDM), multiplexación por división de longitud de onda Densa (DWDM), multiplexación por división de longitud de onda convencional (CWDM), multiplexor óptico reconfigurable Add-Drop (ROADM), multiplexación por división de frecuencias (FDM), multiplexación por división de frecuencia ortogonal (OFDM), Agregar/Dejar multiplexación (ADM).

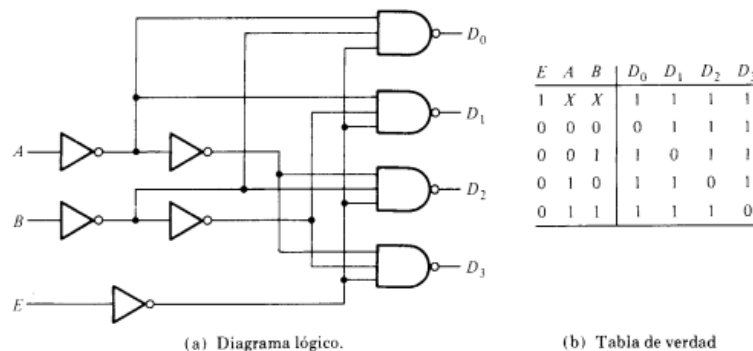
## 2. DEMULTIPLEXOR (DeMux)

Son dispositivos que permiten seleccionar por cual de n salidas se presenta una sola entrada. Su función es la inversa del Mux y junto con él se puede por ejemplo formar un enlace de transmisión multicanal por TDM (Time Division Multiplexing: Multiplexado por División de Tiempo), es decir, poder transmitir n canales por un mismo cable y recuperar la información de los mismos del otro extremo, controlando adecuadamente las líneas de selección de ambos dispositivos en sincronismo con una señal de clock. Un diagrama en bloques de un demultiplexor se ve en la Fig BT 2.1



**Fig BT 2.1** Diagrama de bloques de un Demultiplexor

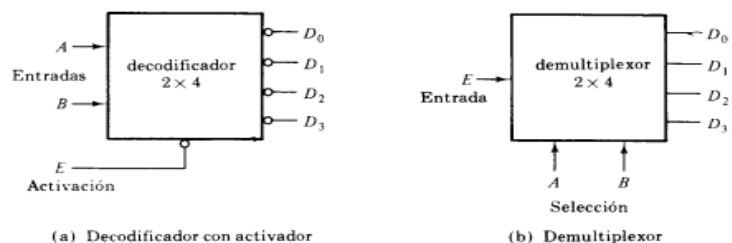
Vale dejar en claro que un decodificador (que se explicará más detalladamente más adelante en este apartado) con una entrada de habilitación puede funcionar como demultiplexor. Algunos IC se construyen con compuertas NAND. Como una compuerta NAND produce una operación AND con una salida invertida, es más económico generar los términos mínimos del decodificador en su forma complementada. La mayoría si no todos los IC decodificadores, incluyen una o más entradas de activación (enable), para controlar la operación del circuito. Un decodificador en línea de 2 a 4 con una entrada de activación y construido con compuertas NAND se muestra en la Fig. BT 2.2. Todas las salidas son iguales a 1 si la entrada de activación E es 1, no importando los valores de las entradas A y B. Cuando la entrada de activación es igual a 0, el circuito opera como decodificador con salidas complementadas.



**Fig. BT 2.2** Un decodificador de línea 2 a 4 con entrada activadora (E)

El diagrama de bloque del decodificador se muestra en la Fig. BT 2.3(a). El circuito pequeño en la entrada E indica que el decodificador se activa cuando E=0. El pequeño círculo a la salida indica que todas las salidas están complementadas.

Un demultiplexor es un circuito que recibe información por una sola línea y transmite esta información en una de las  $2^n$  líneas posibles de salida. La selección de una línea de salida específica se controla por los valores de los bits de n líneas de selección.



**Fig BT 2.3** Diagramas de

bloque para circuito de la Fig. BT 2.2

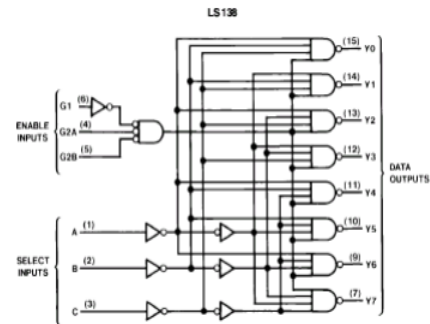
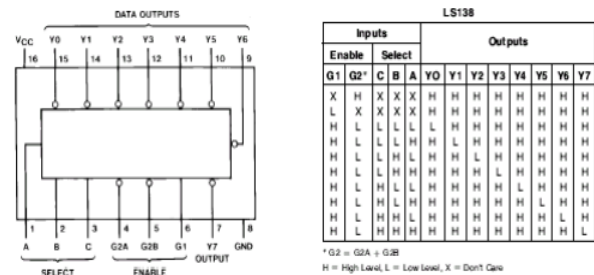
El decodificador de la Fig. BT 2.2 puede funcionar como demultiplexor si la línea E se toma como línea de entrada de datos y las líneas A y B como líneas de selección tal como se muestra en la Fig. BT 2.3(b). La sola variable de entrada E tiene un camino a todas las salidas, pero la información de entrada se dirige solamente a una de las líneas de salida de acuerdo al valor binario de las dos líneas de selección A y B. Esto puede verificarse de la tabla de este circuito mostrada en la Fig. BT 2.2(b). Por ejemplo si la selección de las líneas AB = 10 la salida D2 tendrá el mismo valor que la entrada E, mientras que las otras salidas se mantienen en 1. Como las operaciones decodificador y demultiplexor se obtienen del mismo circuito, un decodificador con una entrada de activación se llama un decodificador/demultiplexor. Es la entrada de activación la que hace al circuito un demultiplexor; el decodificador de por sí puede usar compuertas AND, NAND y NOR.



También vale recalcar que los circuitos decodificador/demultiplexor pueden conectarse conjuntamente para formar un circuito decodificador mayor.

Como ejemplo de un IC véase el 74LS138, un Demultiplexor y Decodificador. En la Fig BT 2. se ve el diagrama de conexiones, tabla de verdad y diagrama esquemático de este dispositivo. Analizando la tabla de verdad y el diagrama esquemático, se puede observar que utilizando cualquiera de las líneas de habilitación /E1, /E2 o E3, como línea de datos de entrada, a través de la selección de las líneas A0, A1 y A2, se puede seleccionar por cual de las ocho salidas (/O0 ... /O7), la señal digital saldrá.

**Fig. BT 2.** Diagrama de conexiones, tabla de verdad y diagrama esquemático del IC demultiplexor/decodificador 74ls138



## Aplicaciones del demultiplexor

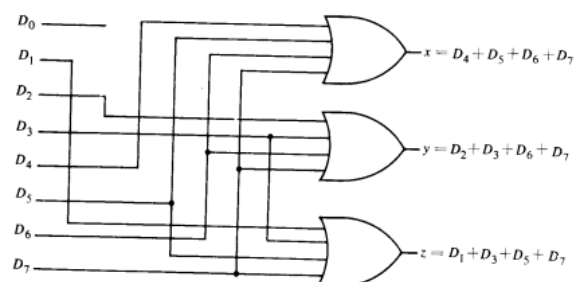
Los demultiplexores se utilizan para conectar una sola fuente a múltiples destinos. Estas aplicaciones incluyen lo siguiente:

- Sistema de comunicación: el multiplexor y el demultiplexor se utilizan en los sistemas de comunicación para llevar a cabo el proceso de transmisión de datos. Un demultiplexor recibe las señales de salida del multiplexor; y, en el extremo del receptor, los convierte nuevamente a la forma original.
- Unidad lógica aritmética: la salida de la unidad lógica aritmética se alimenta como una entrada al Demultiplexor, y la salida del demultiplexor está conectada a múltiples registros. La salida de la ALU se puede almacenar en múltiples registros.
- Convertidor de serie a paralelo: el convertidor de serie a paralelo se utiliza para reformar datos paralelos. En este método, los datos en serie se proporcionan como una entrada al Demultiplexor en un intervalo regular, y se conecta un contador al demultiplexor en la entrada de control para detectar la señal de datos en la salida del demultiplexor. Cuando se almacenan todas las señales de datos, la salida del demultiplexor se puede leer en paralelo.

## 3. CODIFICADOR

Igualmente, según Morris-Mano, un codificador es un circuito digital que efectúa la operación inversa de la que efectúa un decodificador. El codificador tiene  $2^n$  (o menos) líneas de entrada y n líneas de salida. Estas últimas generan el código binario correspondiente al valor de entrada.

Un ejemplo de un codificador se muestra en la para Fig. BT 3.1. El codificador octal a binario consiste en ocho entradas, una para cada uno de los ocho dígitos y tres salidas para generar el número binario correspondiente. El codificador en la asume que solamente una línea de entrada puede ser igual a 1 en cualquier momento; de otra forma el circuito no tiene significado.. Nótese que el circuito tiene ocho entradas y podría tener  $2^8 = 256$  combinaciones de entrada posibles. Solamente ocho de estas combinaciones tienen significado. Las otras combinaciones son condiciones de no importa.



**Fig. BT 3.1** Codificador octal a binario

Los codificadores de este tipo (Fig. BT 3.1) no se encuentran en IC que se pueden construir fácilmente con compuertas OR. El tipo de codificador que se encuentra en la forma de IC es el codificador de prioridad. Estos codificadores establecen una prioridad de entrada para asegurar que solamente la línea de entrada de la más alta prioridad se codifica. Este, junto con los codificadores sin prioridad, constituyen los dos tipos principales de codificadores existentes y que se tratarán en este apartado.

**Los codificadores sin prioridad** se caracterizan porque cuando se activan simultáneamente varias entradas, en las salidas aparece la combinación lógica correspondiente a la suma lógica de las salidas que se obtendrían con cada una de las entradas activadas por separado. Esto implica que solamente una de las entradas puede ser activada cada vez, ya que si no, el código obtenido en las salidas sería falso.

Tómese como ejemplo de comparación, un codificador 8 a 3 sin prioridad (en el apartado de “con prioridad” se realizará el mismo codificador para ilustrar las diferencias). Como se ha de cumplir que  $8 \leq 2^n$ , el número n de salidas será 3. Por un lado, se ponen las ocho entradas de E0 a E7 y, por otro, las tres salidas. Como se puede observar en la tabla de verdad (Fig. BT 3.2), ante cada entrada se presentará una combinación a la salida.

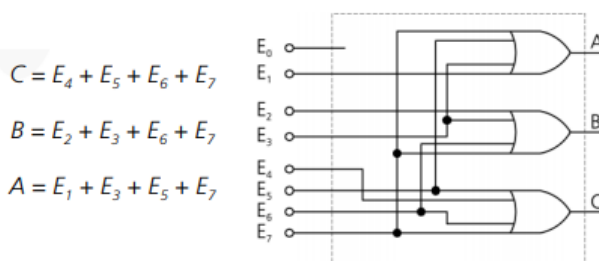
E0	E1	E2	E3	E4	E5	E6	E7	C	B	A
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	1	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

**Fig BT 3.2** Tabla de verdad para el codificador 8 a 3

De esto, las ecuaciones de salida y el circuito realizado con compuertas lógicas en la Fig. BT 3.3, aplicándose tres puertas OR de cuatro entradas. Si se activasen dos entradas simultáneamente se produciría un código falso.

**Fit BT 3.3** Ecuaciones de salida y circuito lógico para codificador 8 a 3 sin prioridad

A este grupo también se le añade el codificador decimal/BCD sin prioridad.



Como ejemplo, véase un codificador 10 a 4. Puesto que se tienen diez entradas y cuatro salidas, como se ve en la tabla de verdad de la Fig BT 3.7.

ENTRADAS DECIMAL	SALIDAS BCD			
	D	C	B	A
E0	0	0	0	0
E1	0	0	0	1
E2	0	0	1	0
E3	0	0	1	1
E4	0	1	0	0
E5	0	1	0	1
E6	0	1	1	0
E7	0	1	1	1
E8	1	0	0	0
E9	1	0	0	1

Fijándose en la tabla de verdad, se obtiene cuatro funciones de salida que se corresponden con cada uno de los dígitos del código BCD (Fig BT 3.8).

**Fig BT 3.7** Tabla de verdad del codificador decimal BCD

**Fig BT 3.8** Funciones de salidas del codificador decimal a BCD

A partir de las ecuaciones se implementa el circuito lógico visto en la Fig BT 3.9.

Es un codificador sin prioridad ya que, si se presentan dos entradas simultáneas, las salidas serán las que correspondan a cada entrada por separado.

**Fig BT 3.** Circuito lógico del codificador decimal a BCD sin prioridad  
**Los codificadores con prioridad** son aquellos en los que, cuando se activan simultáneamente varias entradas, las combinación lógica de salida corresponde a una sola de las entradas activadas, que normalmente es la de mayor valor decimal. El circuito es más complejo, pero el método para obtenerlo es similar.

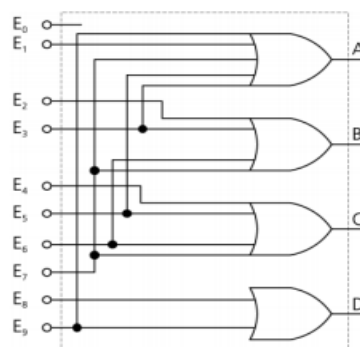
Un codificador de prioridad es casi el complemento de un decodificador, en el sentido de que un codificador de prioridad conectado a las salidas de un decodificador produce una función identidad, es decir, la salida del codificador de prioridad es igual a la entrada al decodificador. Por otra parte, un decodificador conectado a las salidas de un codificador no produce una función identidad porque un codificador de prioridad se diseña para codificar la posición de la entrada con valor uno más significativa, prescindiendo de los valores del resto de entradas en posiciones menos significativas. Por este

$$D = E_8 + E_9$$

$$C = E_4 + E_5 + E_6 + E_7$$

$$B = E_2 + E_3 + E_6 + E_7$$

$$A = E_1 + E_3 + E_5 + E_7 + E_9$$





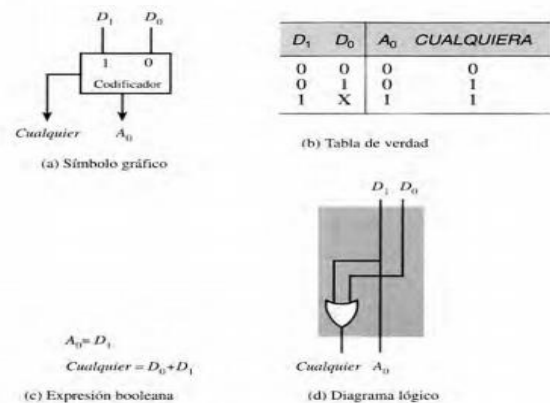
motivo, los valores de entrada del codificador no pueden reproducirse a las salidas del decodificador conectado a él.

En términos generales, un codificador de prioridad tiene  $n$  entradas,  $D_{n-1} \dots D_0$ , siendo  $n = 2^m$  para algún  $m$ . También tiene  $\log_2 n = m$  salidas,  $A_{m-1}, \dots, A_0$ , así como una salida adicional llamada Cualquiera, que será 1 siempre que alguna de las entradas tenga algún valor distinto de 0. Las salidas  $A_{m-1} \dots A_0$  representan el índice del bit de entrada  $D$ ; más significativo que sea 1.

El más sencillo de todos los posibles codificadores es el codificador de prioridad de 2 a 1 mostrado en la Fig. BT 3.4. Su símbolo gráfico se muestra en la Fig. BT 3.4 (a) y su tabla de verdad en la Fig. BT 3.4 (b). Como puede verse, el codificador de prioridad de 2 a 1 tiene dos entradas,  $D_1$  y  $D_0$  y una salida,  $A_0$ , además de la salida Cualquiera descrita anteriormente. En este codificador, siempre que  $D_1 = 0$  y  $D_0 = 1$  entonces  $A_0 = 0$

**Fig BT 3.4** Codificador prioridad de 2 a 1

e, inversamente, siempre que  $D_1 = 1$  y  $D_0$  sea indiferente entonces  $A_0 = 1$ . Las expresiones booleanas de la salida pueden obtenerse fácilmente a partir de la tabla de verdad, y se muestran en la Fig. BT 3.4 (c). Finalmente, en la Fig. BT 3.4 (d) se muestra el circuito lógico de este codificador de prioridad de 2 a 1.



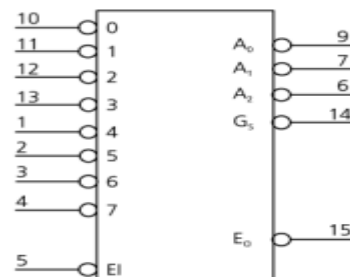
Sin embargo, un codificador de prioridad más utilizado que el de 2 a 1 es el de 4 a 2. En sí, la importancia de los codificadores de 2 a 1 y de 4 a 1 reside en que combinándolos con selectores se pueden construir codificadores de prioridad con mayor número de entradas

Como ilustración se ejemplificará estos codificadores con un IC comercial, el 74148, codificador de 8 a 3 líneas con prioridad. En la Fig BT 3.5 se aprecia la tabla de la verdad de este dispositivo. El codificador 74148 acepta ocho entradas activas a nivel bajo y saca en salida una representación binaria de tres salidas, también activas, a nivel bajo que son  $A_2$ ,  $A_1$  y  $A_0$ . Dispone además de una Enable input que pone inactivas todas las salidas si no está activada. La salida GS se activa cuando hay alguna entrada activa y la salida EO se encuentra activa cuando todas las señales de entrada están inactivas.

ENTRADAS									SALIDAS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	H	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

**Fig. BT 3.5** Tabla de verdad del IC 74148

**Fig. BT 3.6** Símbolo lógico del IC 74148



Además, se añade al codificador decimal/BCD con prioridad como otro tipo añadido.

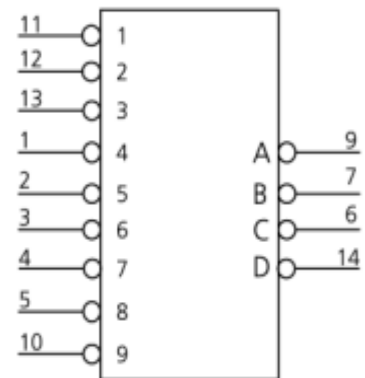
Como ejemplo del codificador decimal/BCD con prioridad véase que cuando se activa una de las entradas decimales, las salidas toman el estado correspondiente a su código BCD.

ENTRADAS									SALIDAS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

**Fig BT 3.** Tabla de verdad del IC 74147

El circuito integrado 74147 es un codificador decimal a BCD con prioridad, que asegura que ante varias entradas activas solamente la de mayor orden aparecerá codificada en la salida. El cero queda codificado con todas las entradas inactivas, con lo cual las salidas quedan puestas a nivel alto.

**Fig BT 3.** Símbolo lógico del IC 74147



#### Aplicaciones del codificador:

##### a) Codificador de teclado:

Los codificadores de prioridad se pueden usar para reducir la cantidad de cables necesarios en un circuito o aplicación en particular que tienen múltiples entradas. Por ejemplo, suponga que un microordenador necesita leer las 104 teclas de un teclado QWERTY estándar donde solo se presionará una tecla "ALTA" o "BAJA" en cualquier momento. Una forma sería conectar los 104 cables de las teclas individuales del teclado directamente a la entrada de la computadora, pero esto no sería práctico para una PC doméstica pequeña. Otra alternativa y mejor manera sería conectar el teclado a la PC usando un codificador de prioridad. Los 104 botones o teclas individuales podrían codificarse en un código ASCII estándar de solo 7 bits (0 a 127 decimal) para representar cada tecla o carácter del teclado y luego ingresarse como un código BCD de 7 bits mucho más pequeño directamente en la computadora. Los codificadores de teclado, como el codificador de 20 teclas 74C923, están disponibles para hacer precisamente eso.

##### b) Codificadores posicionales:

Como codificador de prioridad, el control magnético de posición como se usa en la navegación de barcos o para la posición del brazo robótico, etc. Aquí, por ejemplo, la posición angular o giratoria de una brújula se convierte en un código digital mediante un codificador de prioridad de línea de 8 a 3 de 74LS148 y la entrada a la computadora del sistema para proporcionar datos de navegación y un ejemplo de un encoder de compás de salida de 8 bits a 3 bits simple se muestra a continuación. Los imanes y los interruptores de láminas podrían usarse en cada punto de la brújula para indicar la posición angular de las agujas.

##### c) Solicitudes de interrupción en aplicaciones de microprocesadores:

El microprocesador utiliza interrupciones para permitir que los dispositivos periféricos, como la unidad de disco, el escáner, el mouse o la impresora, se comuniquen con él, pero el microprocesador sólo puede "hablar" con un dispositivo periférico a la vez, por lo que necesita alguna forma de saber cuándo. Un dispositivo periférico particular quiere comunicarse con él.

El procesador hace esto mediante el uso de señales de "Solicitudes de interrupción" o "IRQ" para asignar prioridad a todos los dispositivos periféricos para garantizar que el dispositivo periférico más importante reciba servicio primero. El orden de importancia de los dispositivos dependerá de su conexión al codificador de prioridad.

Debido a que la implementación de un sistema de este tipo que utiliza codificadores de prioridad, como el IC codificador de prioridad 74LS148 estándar, implica circuitos lógicos adicionales, se encuentran disponibles circuitos integrados diseñados específicamente como el Controlador de Interrupción de Prioridad Programable 8259.

## 4. DECODIFICADOR

Ya quedó claro en las prácticas pasadas donde se estudió dedicadamente los (de)codificadores que las cantidades discretas de información se presentan en sistemas digitales con códigos binarios. Un código binario de  $n$  bits es capaz de representar hasta  $2^n$  elementos diferentes de información codificada. Un decodificador es un circuito combinacional que convierte la información binaria de  $n$  líneas de entrada a un máximo de  $2^n$  líneas únicas de salida. Si la información decodificada de  $n$  bits tiene combinaciones no usadas o de no importa, la salida del decodificador tendrá menos de  $2^n$  salidas.

La Fig. BT 4.1, muestra el diagrama general de un decodificador de N entradas y M salidas. Puesto que cada una de las entradas puede ser 1 o 0, hay  $2^N$  combinaciones o códigos de entrada. Para cada una de estas combinaciones de entrada sólo una de la M salidas estará activada 1, para lógica positiva; todas las otras salidas estarán en 0. Muchos decodificadores se diseñan para producir salidas 0 activas, lógica negativa, donde la salida seleccionada es 0 mientras que las otras son 1.



Fig. BT 4.1. Diagrama general de un decodificador

Para entender su funcionamiento, tómese como ejemplo, el circuito decodificador en línea de 3 a 8 de la Fig. BT 4.2. Las tres entradas se decodifican en ocho salidas y cada salida representa uno de los términos mínimos de las variables de 3 entradas. Los tres inversores generan el complemento de las entradas y cada una de las ocho compuertas AND generan uno de los términos mínimos. Una aplicación particular de este decodificador sería una conversión binaria a octal. Las variables de entrada podrían representar un número binario y las salidas representarían los ocho dígitos en el sistema de numeración octal. Sin embargo, un decodificador en línea de 3 a 8 puede ser usado para decodificar cualquier código de 3 bits para genera. ocho salidas, una para cada elemento del código.

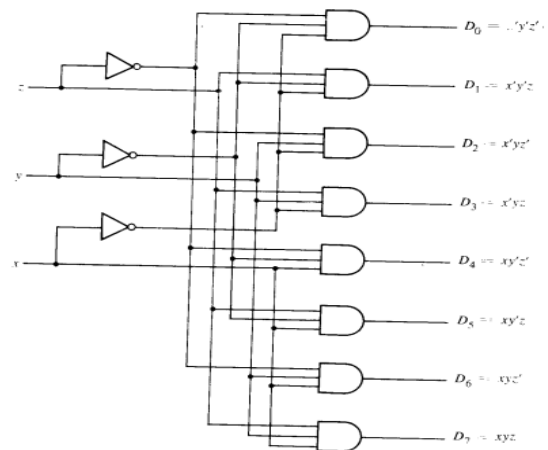
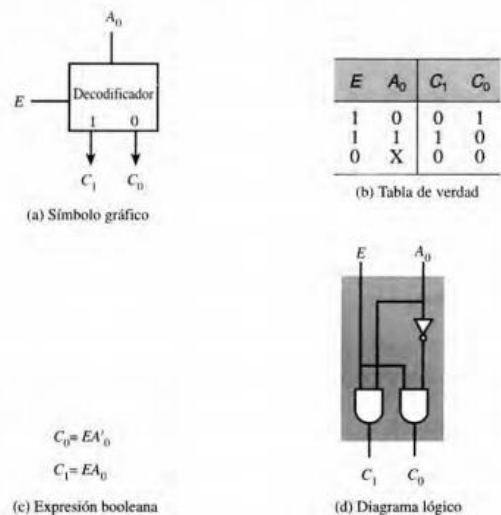


Fig. BT 4.2 Decodificador 3 a 8

Como segundo ejemplo, véase la Fig BT 4.3. Se muestra un decodificador de 1 a 2, y el símbolo gráfico y la tabla de verdad se dan en las Fig. BT 4.3 (a) y (b). Como puede verse, el decodificador tiene una línea de dirección, A<sub>0</sub>, y una línea de habilitación, E, con dos líneas de salida Co y C1'. Mientras E valga 1, Co = 1 siempre que A<sub>0</sub> = 0, y C1 = 1 siempre que A<sub>0</sub> = 1. Las expresiones booleanas de las salidas C1 y Co se muestran en la Fig. BT 4.3 (c), mientras que su implementación con puertas se presenta en la Fig. BT 4.3 (d).

Fig BT 4.3. Decodificador de 1 a 2



Entre los tipos de decodificadores, usualmente se habla de dos: los no excitadores y los excitadores. Los primeros son capaces de excitar a otras puertas de circuitos integrados, pero no pueden activar relés ni dispositivos de presentación, como ejemplo, tómese un excitador siete segmentos. Por otra parte, los decodificadores excitadores sí son capaces de activar relés y diferentes tipos de visualizadores.

De la misma manera, también suele considerarse el decodificador BCD a decimal como un tipo de decodificador. Se trata de un circuito combinacional decodificador que convierte el código BCD a decimal. Para cada combinación en BCD que se presenta a la entrada, se activará la salida equivalente en decimal, por tanto, se tiene 4 entradas y 10 salidas. Si el número de salidas N es menor que  $2^n$ , siendo n el número de entradas, es que hay  $(2^n - N)$  entradas denominadas condiciones de indiferencia. En este caso hay seis condiciones de indiferencia. Si estas condiciones de indiferencia se consideran como datos falsos de entrada y ante estas entradas no se selecciona ninguna salida, se dice que el decodificador rechaza datos falsos de entrada. Si, por el contrario, estas entradas de datos falsos se usan para simplificar las expresiones de salida, el decodificador no rechaza datos falsos.

Véase como ejemplo real el IC 7445 que es un decodificador excitador BCD a decimal, que rechaza datos falsos de entrada desactivando todas las salidas. Ante una entrada en BCD, activa la salida decimal correspondiente poniéndola a nivel bajo. Se puede emplear como excitador de relés o indicador.

Véase como segundo ejemplo el decodificador excitador BCD-7 segmentos. Los elementos de visualización de siete segmentos son muy comunes. Como ya se ha visto en prácticas pasadas, los segmentos son diodos LED, colocados de tal forma que pueden representar diferentes caracteres según los diodos que se iluminen. El decodificador BCD a 7 segmentos ante una entrada en código BCD se encarga de activar las salidas correspondientes a los segmentos luminosos que han de encenderse para representar en decimal el código introducido.

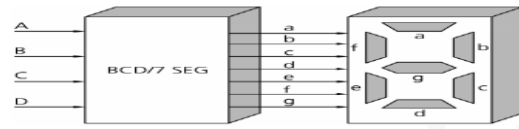
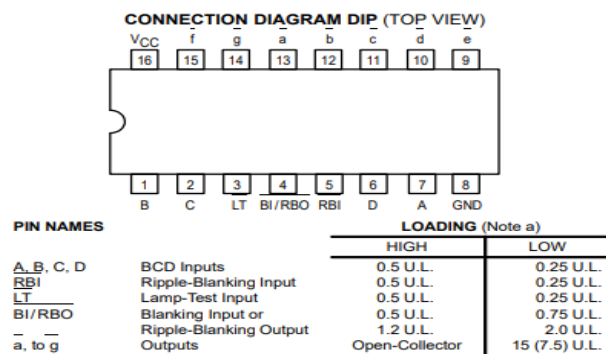


Fig BT 4.4 Visualizador de 7 segmentos

## ● El decodificador como IC

El Decodificador 74LS47 es un circuito integrado TTL LS de bajo consumo eléctrico y alta velocidad, operacionalmente es un decodificador/driver BCD a 7-segmentos con etapas de salida colector abierto activas en 0 lógico, capaces de manejar directamente leds 7-segmentos ánodo común. Este componente acepta como entrada código de 4 bits BCD (patillas 7, 1, 2 y 6). Mediante un circuito combinacional formado por compuertas lógicas AND, NAND, NOR y OR, convierte cada combinación de entrada en su equivalente en código 7-segmentos. Las salidas son activas en cero y colector abierto por lo que necesitan un resistor conectado en serie (pull up) con la carga para fijar la corriente de salida, el valor es de 24mA máximo por segmento para el 74LS47 y 40mA para el 7447. Regularmente el elemento de indicación es un display Led 7 segmentos ánodo común, aunque también la carga puede ser una interfaz para manejar lámparas de mayor consumo. Las patillas de salida son: 9 a la 15.

Fig BT 4.5 Diagrama de conexión del decodificador 74ls47



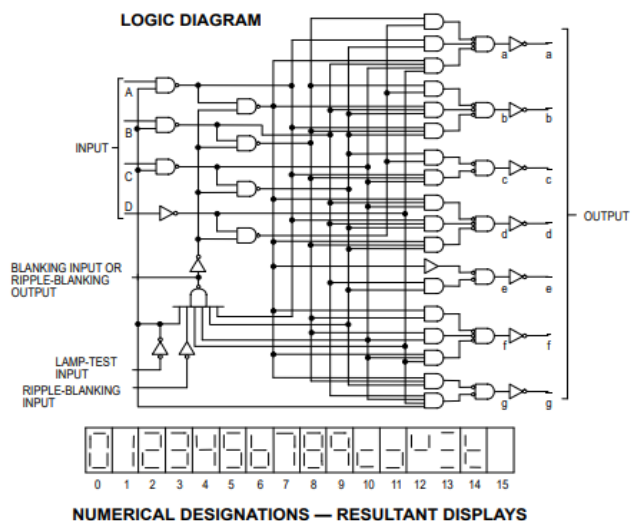
El circuito integrado presenta funciones adicionales:

- LT Lamp Test
- RBI Ripple Blanking Input
- BI/RBO Blanking Input/Ripple Blanking Output.

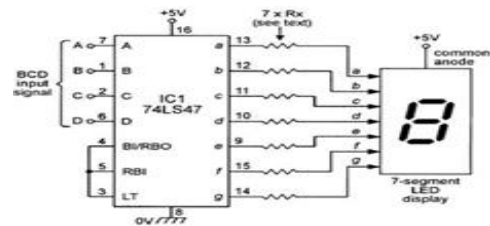
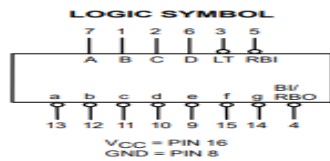
La entrada Blanking Input (BI) debe estar a nivel alto para el funcionamiento normal del decodificador. Con un nivel bajo a la entrada LT y la entrada BI/RBO a nivel alto, se activan todos los segmentos. Esto se utiliza para probar el funcionamiento del display. Con un nivel bajo en la entrada BI todos los segmentos se desactivan (off) La entrada RBI permite la supresión de cero, una función utilizada en displays de varios dígitos para eliminar los ceros innecesarios. Se activa en cero.

El 74LS47 posee características las cuales son:

- Salida de colector abierto y Ánodo común
- Las salidas de colector abierto conducen directamente los indicadores
- Provisto con prueba de lámpara
- Puesta a cero con flanco positivo/negativo
- Todos los tipos de circuitos incluyen la capacidad de modulación de intensidad de la lámpara
- Supresión de cero / arrastre de cero
- Todos los tipos de circuitos cuentan con capacidad de modulación de intensidad de lámpara
- Conduce un LED ánodo común mediante indicadores incandescentes
- Disposición de la prueba de la lámpara



- Líder / Supresión cero final
- Aplicaciones: Procesado de Señal, Defensa, Militar y Aeroespacial



**Fig BT**

**4.6**

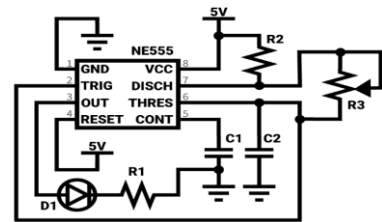
Símbolo lógico, diagrama lógico y conexión a 7 segmentos de un decodificador 74ls47

Algunas aplicaciones del decodificador pueden ser:

- Reloj digital.

Un reloj es una de las aplicaciones más comunes de circuitos digitales, antes de los microcontroladores se usaron circuitos digitales más sencillos, estos circuitos integrados realizan funciones más específicas pero con esta aplicación se demuestra que son bastante capaces e incluso en algunas ocasiones son más convenientes que un microcontrolador. Para diseñar un reloj, el decodificador debe venir acompañado también de un contador (74ls90, por ejemplo), los displays 7 segmentos, una compuerta AND (74ls08), un NE555, un potenciómetro, capacitores, resistencias y jumpers necesarios para la conexión. Por su parte, la señal de reloj, que ajustará la velocidad del reloj, está basado en el NE555 y se presenta en un circuito funcional de la Fig 4.7

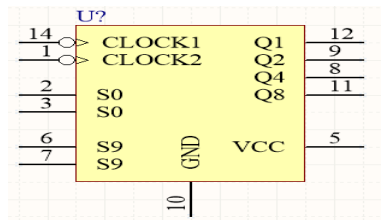
**Fig BT 4.7** Circuito para NE555 reloj digital



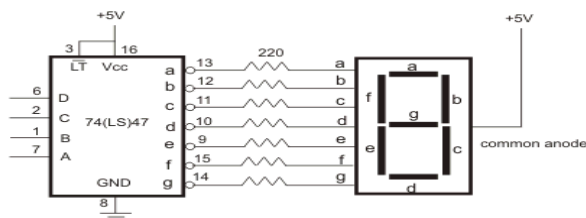
Para realizar el conteo se utiliza el circuito integrado 74LS90, el cual es un contador binario, tiene la posibilidad de establecer el límite de conteo (por defecto el límite es 10) y además genera una segunda señal de reloj que sirve para encadenar varios contadores. En la Fig BT 4.8 se puede ver el pinout

**Fig BT 4.8.** Pin Out del IC 74ls90

En la entrada de la señal de reloj se conecta la señal obtenida con el NE555, más adelante se mostrará el circuito completo. Con cada pulso de la señal de reloj aumenta en 1 la salida del contador.



Hasta ahora los contadores permitirán visualizar números binarios con LEDs, los display de 7 segmentos permitirán visualizar el dato en sistema decimal, para esto se usará como un intermediario al circuito integrado 74LS47, el cual es un decodificador para poder visualizar el dato binario en el display de 7 segmentos (Fig BT 4.10).



**Fig BT 4.10.** Configuración de decodificador 74ls47 con el 7 segmentos

Pin	Función
VCC	Alimentación 5V
GND	Tierra
CLOCK2	Salida de señal de reloj generada
CLOCK1	Entrada de señal de reloj
S0	Reinicio de conteo
S9	Reinicio
Q1	Salida (bit 0)
Q2	Salida (bit 1)
Q4	Salida (bit 2)
Q8	Salida (bit 3)

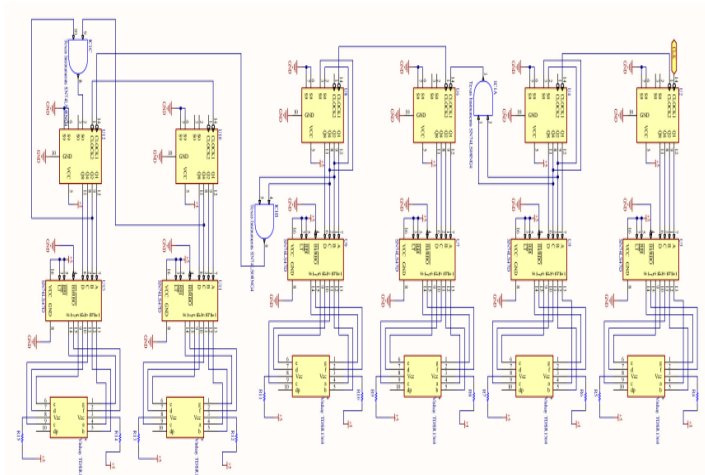


**Fig BT 4.9.** Tabla de funciones de pines del IC 74ls90

En la Fig BT 4.11 se encuentra el diagrama completo del circuito para el Reloj con 74LS90 y display de 7 Segmentos. Los resistores que van conectados a los display de 7 segmentos son de 330 ohms.

**Fig BT 4.11.** Circuito final para reloj digital

Otros usos y aplicaciones de sistemas construidos con decodificadores involucran a los tacómetros digitales, los circuitos contadores en sentido general, la electromedicina, circuitos donde se empleen PICs, frecuencímetros digitales, circuitos de presentación de información y registro.



## 5. SWITCH:

En electrónica, el acrónimo DIP cuando se aplica a los tipos de switch significa "paquete dual en línea" (dual in-line package, en inglés). Un switch de paquete dual en línea, o DIP Switch, en realidad es un conjunto de pequeños interruptores electrónicos manuales que están diseñados para empaquetarse con otros circuitos. El término DIP switch puede referirse a un interruptor individual en una unidad de conmutadores múltiples o a la unidad completa en su conjunto.



**Fig BT 5.1** DIP switch

En pocas palabras, el papel de los DIP switch es permitir a los usuarios controlar el flujo de electricidad alrededor de una placa de circuito impreso (PCB), tarjeta de expansión u otro periférico de la electrónica / computadora y, por lo tanto, cambiar el modo de funcionamiento de un dispositivo.

Cuando se instalan en una PCB junto con otros componentes eléctricos, los DIP switch ofrecen al usuario la posibilidad de personalizar el comportamiento del dispositivo electrónico en cuestión, lo que permite un control más completo sobre la función precisa que el dispositivo realizará en una variedad de escenarios o aplicaciones definidos. Los DIP switch generalmente se presentan como una fila de pequeños interruptores de activación / desactivación, en lugar de una serie de interruptores de luz estándar en miniatura, montados secuencialmente en un bloque de polímero termoplástico resistente al calor.

Como componente completo, un conjunto de interruptores DIP es casi siempre muy pequeño y requiere cierta destreza por parte del usuario para alternar cada interruptor entre las posiciones de encendido y apagado. En la práctica diaria, esta función a menudo se realiza con la punta de un lápiz afilado o una herramienta similar con la forma adecuada, ya que los interruptores individuales tienden a ser demasiado pequeños para la operación cómoda de los dedos.

Un DIP switch es un componente relativamente simple y rentable que se usa comúnmente como una forma económica de seleccionar y cambiar entre varias opciones de hardware o dispositivo. Por lo general, son considerablemente más baratos que otras opciones, como paneles de control de software o chips programables, con el compromiso de que los interruptores DIP requieren que el usuario abra físicamente el dispositivo para acceder a la PCB y operar los interruptores manualmente desde allí.

## Funcionamiento

Un DIP switch de montaje en superficie estándar se vende como un conjunto de interruptores electromecánicos básicos de dos terminales (encendido / apagado o 1/0), ensamblados dentro de una carcasa de bloque de plástico térmicamente estable. A menudo se les conoce genéricamente como "puentes", aunque técnicamente son en realidad una alternativa al puente tradicional; sin embargo, su función general es muy similar.

La parte 'dual in line' del acrónimo se refiere a la disposición de los pines de contacto debajo del interruptor DIP, que se conectan directamente a la PCB o placa de pruebas; estos están dispuestos en dos filas paralelas en la parte inferior del bloque del DIP switch, por lo tanto doble en línea. En general, habrá el doble de pines de contacto que los interruptores en un módulo dado.





**Fig BT 5.2** DIP switch de un sólo interruptor

Un interruptor DIP estándar puede consistir en 1, 2, 4, 7, 8, 10, 16 o incluso más interruptores basculantes, basculantes, deslizantes o rotativos montados en un solo bloque pequeño. Aunque son componentes simples, la belleza de los interruptores DIP es que tener varios conmutadores dispuestos en paralelo significa que cada uno puede moverse a 1/0 de forma independiente, lo que puede ofrecer un gran número de combinaciones diferentes a c

Todos los conmutadores individuales en una sola matriz de DIP switch están aislados eléctricamente entre sí. Cuando se mueve un interruptor en un interruptor DIP entre sus posiciones de encendido / apagado, un resorte flexible debajo del interruptor responde al movimiento del deslizador o balancín, y se dobla ligeramente hacia arriba o hacia abajo para cerrar o abrir un contacto estático debajo. o responder con un gran número de posibles funciones / salidas del dispositivo.

Los DIP switch se clasifican para funciones y aplicaciones específicas por sus clasificaciones de corriente, voltaje y potencia operacionales seguras. Es importante que los usuarios observen y se mantengan dentro de las especificaciones dadas por el fabricante para un DIP switch en particular, de lo contrario, pueden ocurrir problemas tales como interruptores auto-soldados y arcos eléctricos durante la conmutación, con el riesgo de inestabilidad del rendimiento o daños al dispositivo.

## Tipos

Existe una diversa cantidad de dispositivos en la familia de los DIP switch. Para no extender demasiado este pre-informe, se enfatizará en sólo algunos más comunes.

### a) DIP switch rotativo

Un interruptor DIP rotativo es aquel en el que el usuario gira a través de un rango de movimiento circular, en lugar de alternar de un lado a otro, ya sea manualmente o con una herramienta como un destornillador si esa es la forma en que el interruptor fue diseñado para actuar. Los interruptores DIP codificados rotativos, al igual que las variedades de balancines y controles deslizantes, también están destinados a montarse directamente en una PCB donde controlarán los comportamientos de salida de un dispositivo electrónico.



**Fig BT 5.3** DIP switch rotativo

Se pueden diseñar para girar continuamente 360 grados o para detenerse en las posiciones predeterminadas a medida que giran. La cantidad de rotación que atraviesa el eje del interruptor DIP es lo que determina el código de circuito binario o hexadecimal generado y, a su vez, la señal de salida enviada al dispositivo al que está conectado.

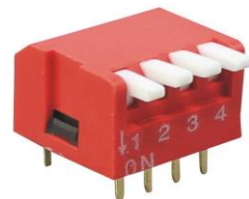
Si bien la mayoría de los interruptores DIP rotativos se pueden usar en casi las mismas aplicaciones que cualquier otro tipo, a menudo se consideran opciones particularmente útiles en escenarios donde el espacio en el PBC es especialmente limitado, o donde la facilidad de ajuste para el usuario se considera primordial.

Están disponibles en modelos impermeables, resistentes a la intemperie y al polvo de algunos vendedores, y son ampliamente utilizados en medidores de precisión, mezcla y edición de sonido, equipos de prueba, comunicaciones de voz / datos aeroespaciales y militares, y transmisores y receptores de radio.

### b) Slide/rocker/piano DIP switches

Los DIP switch slide, los DIP switch rocker y los DIP switch de piano son realmente sólo variaciones del segundo tipo principal de interruptor DIP, es decir, el interruptor de palanca más tradicional. Para todos estos tipos de conmutadores, puede comprar múltiples variantes, que incluyen:

- Normalmente abierto (NO, Normally Open), lo que significa que su posición predeterminada debe desconectarse de un circuito, **Fig BT 5.4** DIP switch piano y al accionar el interruptor se completará nuevamente.



- Normalmente cerrado (NC, Normally Close), lo que significa que al accionar el interruptor se rompe el contacto con el circuito, en lugar de cerrarlo



Un switch de tres etapas, que tendrá tres posiciones seleccionables (a menudo Encendido / Apagado / Encendido), donde la posición central será la predeterminada y presionar o deslizar el interruptor hacia ambos lados lo activará. Además, como es el caso con la gran mayoría de todos los tipos de switch de palanca, los interruptores DIP también se pueden

encontrar en varias configuraciones de pole y throw, que incluyen:

**Fig BT 5.5** DIP switch rocker

- Single pole single throw (SPST): este es el tipo de switch más simple que encontrará en la mayoría de los dispositivos, en el que una sola palanca conecta o desconecta un solo terminal a / de otro
- Single pole double throw (SPDT): un interruptor SPDT obliga al operador a elegir entre conectarse a uno de los dos terminales; como tal, el interruptor siempre estará conectado a uno u otro, y los dos terminales nunca estarán conectados entre sí (aunque puede haber una tercera posición para "ninguno" disponible, particularmente cuando se usan slide switches).
- Double pole double throw (DPDT): esencialmente funciona exactamente como dos switch SPDT separados, pero donde ambos conjuntos de terminales duales están conectados a un solo actuador; el switch DPDT puede controlar dos circuitos separados, pero estos siempre se cambian juntos cuando se mueve la palanca.

## Aplicaciones

Sobre las aplicaciones de los dispositivos DIP Switch en ámbitos reales se puede decir que actualmente no cuentan con la popularidad que supieron tener, pero muchos dispositivos aún los utilizan por ser económicos y permitir verificar configuraciones sin la necesidad de un encendido. Un ejemplo de dispositivos DIP en uso hoy en día es el abre puertas utilizado en garajes. Los interruptores proveen un código de seguridad que coincide con el de la puerta del garaje. Cuando ambos se configuran de manera correcta, los dos pueden comunicarse mutuamente con la misma frecuencia sin la necesidad de un software externo o programa para realizar las configuraciones.

Otros usos involucran la configuración de una amplia gama de opciones en hardware y periféricos para PC como placas base, tarjetas de video, discos duros, módems, módulos de expansión y otros accesorios o componentes auxiliares. También se usan comúnmente en controles remotos universales / multidispositivos para evitar interferencias eléctricas que conducen al control accidental de un dispositivo no deseado, el DIP switch se usará para establecer una frecuencia de radio diferente para cada emparejamiento de transmisor y receptor.

## 6. TÉCNICAS DE DISEÑO DIGITAL

Son varias las técnicas para el diseño de sistemas digitales, en apoyo con la programación y las tecnologías y entornos existentes computacionales, cada una con sus ventajas e inconvenientes. A continuación se describen cada una de ellas:

### • Edición de Esquemáticos

La técnica más intuitiva y usada desde hace decenios es la edición de esquemáticos. De esta forma el diseñador dibuja y describe los sistemas digitales usando puertas lógicas, registros, biestables, etc.; básicamente colocándolos y conectándolos. Esto se llama "captura de esquemáticos" y en el proceso se describe el sistema completo, pudiendo usarse niveles de jerarquía o sistemas ya creados. Obviamente, para evitar el dibujo manual de los sistemas, pronto se hizo necesario entornos de diseño gráfico sobre computador. Este método puede usarse para sistemas pequeños o bien conocidos por el diseñador, donde se pueden describir rápidamente y son fácilmente modificables. Para esta técnica se deben elaborar, con o sin ayuda de herramientas auxiliares, las tablas de Karnaugh de los sistemas combinacionales y secuenciales. Cuando los sistemas aumentan en tamaño la elaboración de estas tablas y circuitos puede llegar a ser inviable. Existe un formato estándar de intercambio de diseño electrónico (EDIF, Electronic Design Interchange Format) independiente de los fabricantes. El formato EDIF puede usarse para intercambio de esquemáticos, pero los sistemas están descritos en modo de texto. Por ejemplo, Xilinx y Altera, principales suministradores de FPGA, permiten introducir diseños en formato EDIF. Altera genera el formato EDIF desde otras formas de descripción en su propio entorno. En cambio, Xilinx solo genera el formato propio NGC (Native Generic Database); esto es así para mejora el rendimiento del flujo de diseño y optimizar el uso de sus circuitos. El formato estándar EDIF es de uso limitado en el intercambio de esquemáticos, y Xilinx obliga a diseñar los esquemáticos con su propio editor. Esto hace que un circuito esquemático editado sobre una

herramienta de diseño de un fabricante de FPGA no sea exportable a la herramienta de otro fabricante; a menos que se genere el formato EDIF, que no siempre es posible. Finalmente, pueden usarse herramientas de diseño de empresas, estas desarrollan entornos que soportan la edición de esquemáticos para diversos fabricantes.

- **Lenguajes de Descripción de Hardware**

Los lenguajes (HDL, Hardware Description Language) son una técnica que permiten describir sistemas digitales usando texto, facilitando su diseño y modificación. Por medio de una compilación se genera la arquitectura del circuito. De esta forma se solventan las dificultades del diseño usando esquemáticos. En los lenguajes de programación ordinarios las sentencias se ejecutan de forma secuencial, y se mantiene la dependencia entre ellas aunque haya una ejecución paralela. En contraposición, en los HDL las sentencias son concurrentes; es decir, se ejecutan de forma simultánea, tal y como se comportan un grupo de puertas lógicas en un circuito digital. Para ejecutar un grupo de sentencias de forma secuencial se han de incluir dentro de lo que se conoce con el nombre de proceso. Un proceso es la descripción de un sistema secuencial, que pasa por una serie de estados. Los procesos son concurrentes entre ellos; es decir, se ejecutan de forma simultánea. Existen dos HDL estándar y ampliamente usados. Uno es VHDL, que es un doble acrónimo (Very High Speed Integrated Circuit Hardware Description Language, VHSIC-HDL). La traducción correcta de este acrónimo sería "lenguaje muy rápido de descripción hardware para circuitos integrados". El otro HDL estándar es Verilog, el cual es más compacto que VHDL, y más fácil de aprender y usar. Esta facilidad se debe a su parecido con el lenguaje C. Verilog es preferido para el diseño en sistemas ASIC por acceder a niveles más bajos en hardware, pero VHDL permite el uso de tipos de datos más complejos. Aparte de los dos HDL estándares cabe destacar AHDL (Altera Hardware Description Language), que es específico de Altera para sus dispositivos programables. La sintaxis de este lenguaje es parecida a la del lenguaje ADA. Una ventaja de este lenguaje es que aprovecha mejor las características de los dispositivos de Altera, pero como desventaja no es portable a otros fabricantes.

- **Lenguajes de Alto Nivel**

Con la aparición de VHDL y Verilog aparece un curioso fenómeno, que se describe a continuación. Los programadores siguen usando los lenguajes de programación de alto nivel (C, Java, etc.) para ejecutar programas secuenciales. Algunos de estos programadores, mayormente formados en el ámbito de la computación y no en el diseño electrónico, se preguntan por qué no pueden diseñar sistemas digitales sobre dispositivos programables. Este deseo se basa en el hecho de que la descripción del circuito se hace en forma de texto, igual que en los lenguajes de alto nivel. A partir de aquí empiezan a desarrollarse entornos de descripción de circuitos basados en esos lenguajes de alto nivel, que no fueron concebidos para la descripción hardware. Estos entornos se basan sobre todo en Java y C. Como inconveniente presentan el hecho de que el diseño solo es portable a pocos fabricantes; y dentro de cada fabricante, a unas pocas familias de dispositivos. Además, los entornos gratuitos pueden quedar restringidos; bien por no incluir los dispositivos más recientes de un fabricante o por no dar suficiente soporte a los diseñadores. En el lenguaje Java se basa JHDL (Java Hardware Description Language), este entorno sólo soporta dispositivos de Xilinx. El entorno C-to-Verilog traduce código C a Verilog y el entorno SystemC se basa en C++, que además es un estándar en código abierto de IEEE. SystemC está soportado por más de veinte desarrolladores de herramientas de diseño. También pueden describirse sistemas con editores de forma de onda, que si bien para circuitos combinacionales pueden ser útiles, para circuitos secuenciales pueden ser de uso complicado. Algunas herramientas incluyen editor de diagrama de flujo o de estados para sistemas secuenciales. Con estos entornos se edita el gráfico de los estados y las transiciones; son muy útiles para el diseño de este tipo de sistemas.

- **Otras Técnicas y Herramientas**

Actualmente los suministradores y/o empresas, ponen a disposición de los usuarios entornos de diseño estándar, en ellos se pueden describir los circuitos de las formas descritas anteriormente. Estas herramientas permiten compilar y simular el diseño, para finalmente generar el fichero de programación del fabricante. También dan información de los recursos hardware necesarios, de la velocidad máxima del circuito y de la potencia consumida. Cabe destacar que, incluye librerías de componentes para su implementación y simulación. Tenemos entonces de diseño: MatLab y Proteus, estas herramientas suponen una ampliación del entorno estándar, están disponibles en las mismas condiciones y solo sirven para los dispositivos del fabricante. A su vez, tenemos Simulink, el cual es un entorno de diseño gráfico que usa diagrama de bloques, permitiendo diseñar de forma rápida y flexible. Los bloques son configurables mediante ventanas de diálogo. Una vez diseñado el sistema puede simularse aprovechando las ventajas de Simulink. La primera ventaja es el acceso directo al espacio de variables de Matlab, lo que facilita la generación de las señales de entrada y el análisis de las salidas. Por un lado, se pueden usar las diferentes fuentes de señal de Simulink, estas se encuentran en el Blockset Sources. Por otro lado, se pueden usar sus visualizadores y destinos de señal, que se encuentran en el Blockset Sinks. Las señales obtenidas se muestran como si fueran de punto flotante, lo que facilita el análisis de las formas de onda. Estas simulaciones son muy rápidas porque incluyen un nivel pobre de detalle de los circuitos de la FPGA, esto hace que la estimación de área ocupada sea aproximada. Por otro lado, no aporta estimaciones de máxima velocidad o potencia consumida. Como ventaja es posible la comprobación de la total funcionalidad del sistema. Esto es posible por ser estas simulaciones rápidas y Matlab poder manejar gran cantidad de datos. Al disminuir el tiempo

de diseño, y ser entornos más versátiles, se facilita al diseñador la comprobación de diferentes arquitecturas y configuraciones. Una vez simulado el sistema se realiza una compilación que genera el proyecto para el entorno estándar del fabricante. El sistema queda descrito de forma estructural en VHDL o Verilog.

- **Técnicas a través de entornos**

Existen compañías que crean software no gratuito que puede emplearse como una técnica para diseñar circuitos digitales de diferentes fabricantes. La empresa Synopsys ofrece la utilidad Synplify en diferentes modalidades. Este entorno de diseño permite elegir entre los fabricantes de FPGA: Xilinx, Altera, Lattice, Atmel, Microsemi y Achronix. Este sistema permite la portabilidad del diseño para los fabricantes habilitados. Como se puede compilar el diseño para cada uno de los fabricantes es posible comparar las prestaciones, sin necesidad de usar el entorno de cada fabricante por separado. La compañía National Instruments ha desarrollado LabView (Laboratory Virtual Instrumentation Engineering Workbench) y NI Multisim, son entornos donde se diseña de forma visual y gráfica. Se usa para diseñar sistemas de instrumentación, control, procesamiento de señal y comunicaciones y con la capacidad de simulación de diseños; admite gran cantidad de equipamiento externo, librerías, componentes ideales, reales y puertos de entrada-salida. Para LabView, los diseños se pueden introducir usando VHDL, Verilog y de forma esquemática.

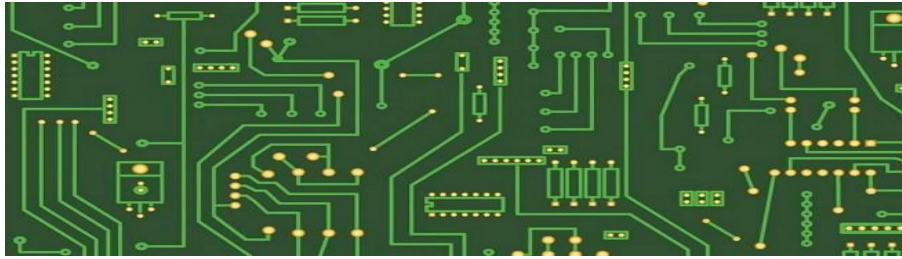
Tenemos también el entorno Matlab, el cual se ha convertido en un estándar de hecho, tanto para la comunidad científica, como para la industria del diseño. Primeramente cabe destacar Filter Design HDL Coder que tiene la capacidad de generar el hardware necesario para el diseño de filtros. Permite obtener en punto fijo la descripción del filtro en VHDL o Verilog, este código es portable a todos los fabricantes. También genera las señales necesarias de entrada para la simulación y verificación del filtro. La utilidad Fixed-Point Designer de Matlab permite manejar tipos de datos y herramientas para el desarrollo de algoritmos en punto fijo, usando código de Matlab, modelos de Simulink o el editor de diagramas de flujo Stateflow. Esta herramienta propone automáticamente el número de bits y el método de redondeo, que también se pueden especificar manualmente. En Matlab cabe destacar la utilidad HDL Coder donde los sistemas se describen usando funciones de Matlab, modelos de Simulink o el editor de diagramas de flujo Stateflow. Esta herramienta automatiza y facilita el flujo del diseño, permite control de la arquitectura y de los retardos críticos; además, da una estimación del área ocupada.

## **Aplicaciones**

Conforme avanza la tecnología, los PCBs se convierten en parte fundamental en el desarrollo y aplicaciones de circuitos electrónicos, cada vez con mayor densidad y por ende exige exactitud y precisión, esto hace necesario la utilización de diseños asistidos por computadora del Arte en PCB ya una vez aplicando las técnicas de diseño anteriormente. A continuación se presentan los pasos generales para la elaboración de PCBs:

- **Diseño:** Un circuito complejo requiere el uso de herramientas computacionales que permitan diseñar y simular el esquema electrónico y su arte de circuito impreso.
- **Cortado:** Conocido el tamaño del circuito impreso, se procede a realizar el corte de la tarjeta.
- **Impresión en la placa de cobre:** una vez que la máscara está lista, se procede a grabarla en la placa, este procedimiento depende de la técnica a utilizar.
- **Atacado del cobre:** se inserta la placa de cobre previamente grabada, en soluciones ácidas que eliminan el cobre no deseado.
- **Limpieza y taladrado:** se realiza el lavado y limpieza de la placa para eliminar todas las impurezas, luego se perforan los orificios en donde se colocaran los componentes.
- **Soldadura:** etapa donde se realiza el montaje (colocación y soldadura) de los componentes.
- **Pruebas de Funcionamiento:** Antes de realizar interconexiones se verifica el funcionamiento del circuito (Cortocircuitos, circuitos abiertos, soldaduras frías, entre otros.)

En el diseño, es necesario definir el esquemático del circuito y el diagrama de conexiones que formarán las pistas de cobre sobre la placa. Generalmente se hace en dos fases; en primer lugar, partiendo de las especificaciones sobre la funcionalidad del circuito, se deciden los componentes a utilizar y las interconexiones necesarias entre ellos, y a través del editor de esquemas del software se realiza el diagrama esquemático. Después, con esa información se define la máscara en el editor del arte del PCB, que es una representación virtual de los componentes sobre la placa, y se establece la forma física de las conexiones entre ellos.



**Fig BT 6.1** Diseño ilustrativo de un PCB

## **TÉCNICAS Y CONSIDERACIONES EN EL DISEÑO DEL ARTE DE PCBs**

Efecto resistivo en las pistas:

- Diseñar las pistas teniendo en cuenta la longitud, grosor y máxima corriente que deban conducir: es recomendable utilizar herramientas de software como para determinar y calcular las dimensiones de pistas a usar según parámetros solicitados; ya que estas dimensiones, mal diseñadas, podrán incrementar el valor de una resistencia no deseada y causar problemas por caídas de tensión o servir de fusible al limitar el paso de corriente. Todo material conductor presenta una resistividad propia.
- En los circuitos de instrumentación y de medición, ubicar los componentes de tal forma que la longitud de las pistas sea lo más pequeña posible, para evitar efectos de carga en la línea de interconexión.

Efecto térmico:

- Estudiar la colocación de los componentes teniendo en cuenta la interconexión, interferencias térmicas e interferencias electromagnéticas. Las condiciones de temperatura podrán causar ligeras variaciones en el valor de resistencia, motivado a que la resistividad de un material dependerá de las condiciones térmicas en las que se encuentre.
- Colocar disipadores a los dispositivos de potencia, acompañado de grasa termo conductiva entre ellos. Los espacios de aire se deben evitar motivado a que es un mal conductor térmico, lo que implicaría una mala disipación, recalentamiento y daño de componente.
- Ubicar los disipadores en lugares ventilados y alejados de componentes susceptibles a la temperatura.

Efecto Capacitivo e Inductivo:

- Para evitar corrientes inducidas, producto de circuitos de alta potencia, se debe distanciar los circuitos de control de los circuitos de potencia.
- Para evitar el efecto de capacitancias parásitas, se debe evitar el paralelismo entre pistas o planos y se aconseja utilizar rectas horizontales en una cara (Cara de Componentes o lado Superior) con rectas verticales en la otra (Lado inferior o cara de soldadura)
- Para reducir el ruido de conmutación en los circuitos digitales, colocar un condensador de 0,1uF entre fuente y tierra, lo más cercano posible a cada integrado. Colocar cada 10 integrados un condensador de 10uF y por cada módulo o tarjeta electrónica colocar un condensador de 47uF.
- Orientar de forma perpendicular al PCB las bobinas y transformadores, para evitar las influencias magnéticas que sobre otros circuitos; ya que en un solenoide, el campo magnético se concentra sobre su eje axial.
- Para los circuitos de alta frecuencia, es recomendable que las curvas de las pistas no superen un ángulo de 45°, ya que podría producirse un auto inducción sobre la misma, deformando su señal.
- Para evitar las EMI (Interferencias Electromagnéticas) y brindar protección eléctrica, se deben separar los planos de tierra analógico y digital o utilizar acoplamientos (transformador, opto acopladores, aisladores de radio frecuencia, etc.).

Transferencia del Arte a la Placa:

- Colocar etiqueta o texto al PCBs, para identificar el lado adecuado a transferir en el cobre.
- Verifique que la impresora no genere discontinuidades y que imprima a la escala correcta.

Pruebas de Funcionamiento:

- Poner puntos de prueba (TP o test point) a la salida de cada etapa, en lugares de la placa donde se pueda conectar instrumental fácilmente.

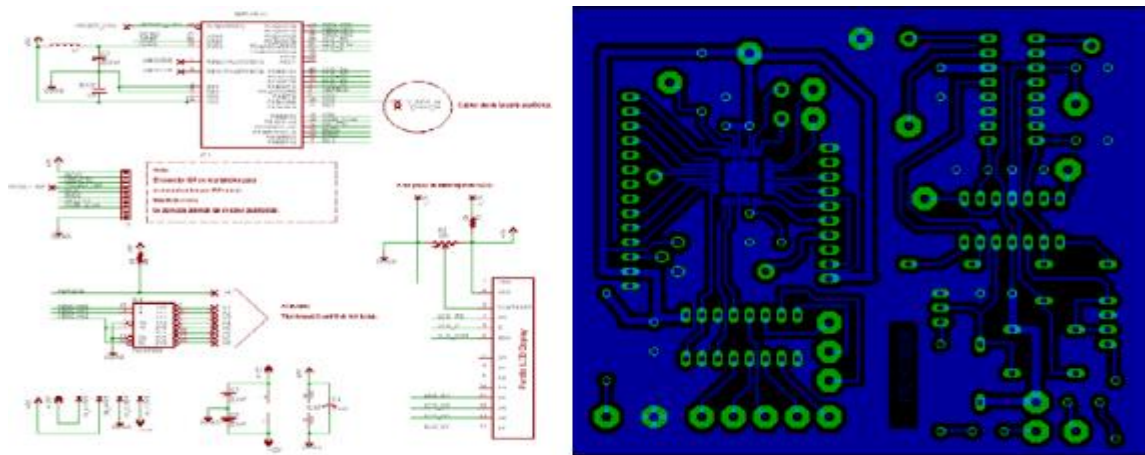
Simplificar el ensamblaje:

- Diseñar las placas de forma estandarizada y modular, de tal manera que la fabricación y ensamblaje sea práctico.

Otras consideraciones:

- No poner pistas, ni colocar componentes cerca de los bordes de las placas donde puedan tener contacto con los tornillos de fijación, guías o con la estructura.
- La separación mínima entre 2 pistas adyacentes debe de ser 0.8 mm, lo que garantiza un buen aislamiento eléctrico de hasta 180 voltios, en condiciones normales.
- Realizar los Pad (Área Plana conductiva donde se realiza las soldaduras de componentes) en función del tamaño y peso de los componentes. Así como también de acuerdo a las fuerzas y tensiones mecánicas que deba soportar.
- Es recomendable que los componentes puedan cambiarse sin necesidad de extraer otro.
- Llenar con planos de tierra la parte de la placa que no lleve pistas; así se evita la contaminación de los oxidantes (ácidos).

Un ejemplo de aplicación de las técnicas de diseño, se observa a continuación en la Fig BT 6.2, donde la figura de la izquierda muestra el diagrama esquemático de un circuito electrónico y la figura de la derecha, muestra el arte de PCB de dicho diagrama:



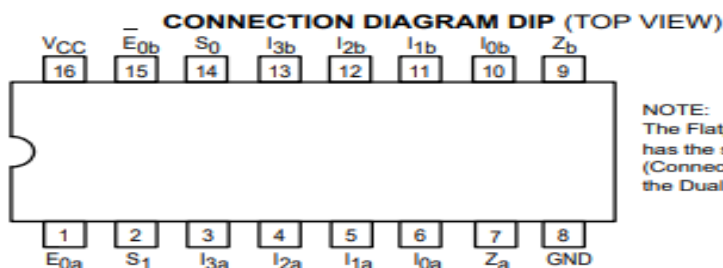
**Fig BT 6.2** Diagrama circuitual diseñado en el Editor de esquema del CAD y a su derecha el Arte del PCB diseñado en el CAD.

## MATERIALES Y EQUIPOS

Multímetro, Alicata, Protoboard, Cables UDP  
Fuente de Voltaje 5V DC, Simulador Ni Multisim

(1) DIP SWITCH, (7) IC Mux 4x1 74LS253  
(1) IC Mux 2x1 74LS158  
(27) Resistencias: (6)  $330\Omega$ , (8)  $150\Omega$ , (13)  $180\Omega$   
(3) Diodos LED Verdes (1) Rojo

## Detalles y Datasheet del IC 74LS253



**Figura M.1** Vista de diagrama de conexión para IC 74LS253



## SN54/74LS253

### LOGIC DIAGRAM

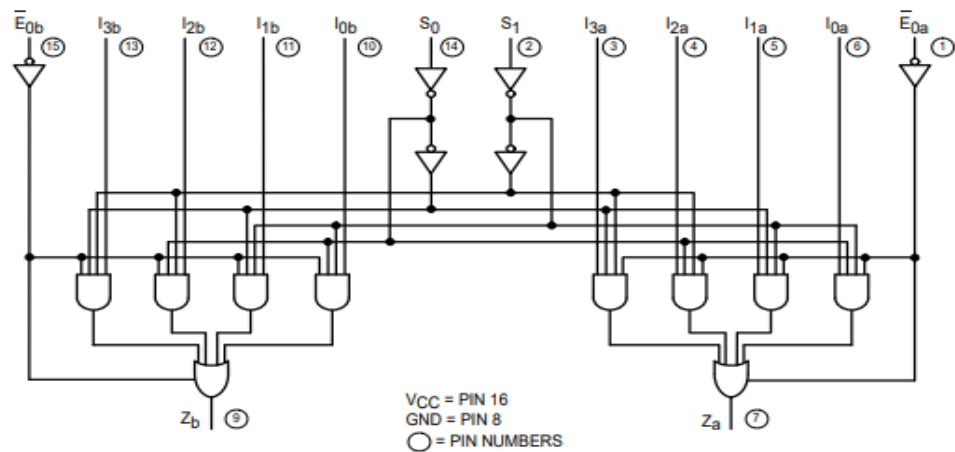


Figura M.2 Vista de diagrama lógico para IC 74LS253

### TRUTH TABLE

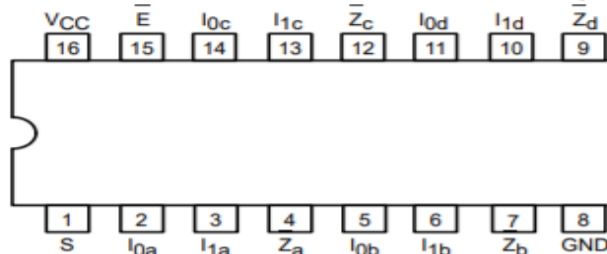
SELECT INPUTS		DATA INPUTS				OUTPUT ENABLE	OUTPUT
S <sub>0</sub>	S <sub>1</sub>	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	E <sub>0</sub>	Z
X	X	X	X	X	X	H	(Z)
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
H	L	X	L	X	X	L	L
H	L	X	H	X	X	L	H
L	H	X	X	L	X	L	L
L	H	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

H = HIGH Level  
L = LOW Level  
X = Irrelevant  
(Z) = High Impedance (off)  
Address inputs S<sub>0</sub> and S<sub>1</sub> are common to both sections.

Figura M.3 tabla de verdad para IC 74LS253

## Detalles y Datasheet del IC 74LS158

### CONNECTION DIAGRAM DIP (TOP VIEW)



NOTE:  
The Flatpak version  
has the same pinouts  
(Connection Diagram) as  
the Dual In-Line Package.

Figura M.4 Vista de diagrama de conexión para IC 74LS158

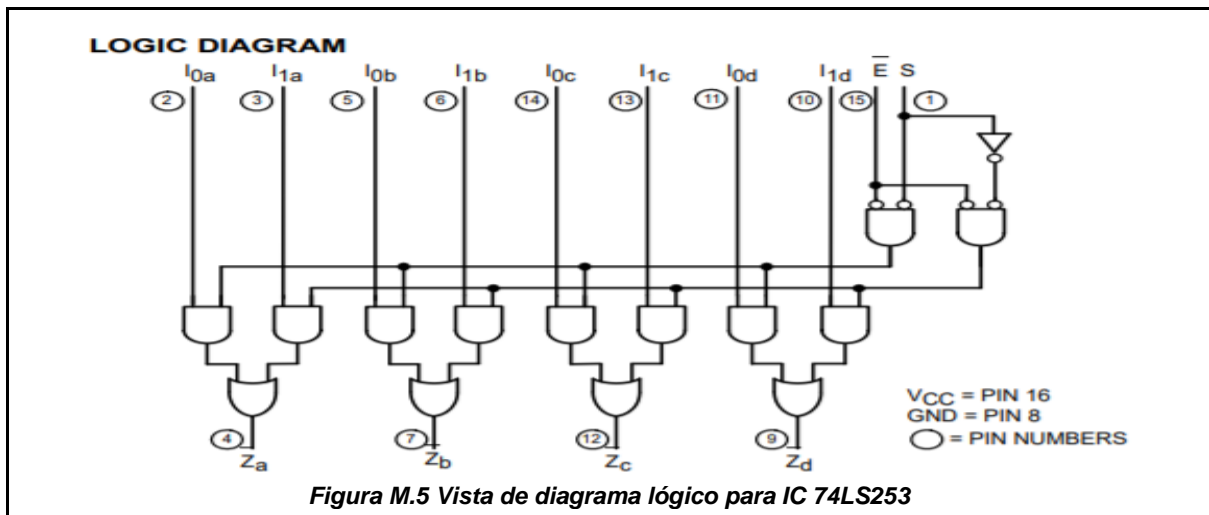


Figura M.5 Vista de diagrama lógico para IC 74LS253

**TRUTH TABLE**

ENABLE	SELECT INPUT	INPUTS		OUTPUT
E	S	I <sub>0</sub>	I <sub>1</sub>	Z
H	X	X	X	H
L	L	L	X	H
L	L	H	X	L
L	H	X	L	H
L	H	X	H	L

H = HIGH Voltage Level  
L = LOW Voltage Level  
X = Don't Care

Figura M.6 tabla de verdad para IC 74LS253

## DESARROLLO

En orden para cumplir esta práctica se presenta una modalidad limitada, o más bien, dirigida de manera diferente a los requisitos de construcción y funcionamiento standard de una ALU, pero que mantiene su función final primaria, esto es, un sistema que produzca operaciones aritméticas y lógicas. Siendo más específico, la construcción prescindirá de algunos implementos como las salidas de arrastre Cout, y sustituyendo otros, como los sumadores paralelos, ideando medidas alternas para la ejecución de las operaciones mediante (de)mux, (de)coders, entre otros.

En sí, una unidad aritmética lógica es una función multioperacional digital de lógica combinacional. Esta puede realizar un conjunto de operaciones aritméticas básicas y un conjunto de operaciones lógicas. La ALU requiere de un número de líneas de selección para seleccionar una operación particular, para este caso se maneja el complemento, el incremento, el decremento y la rotación. Las líneas de selección se decodificarán dentro del ALU de manera que las K variables de selección pueden especificar hasta  $2^k$  operaciones diferentes. Esto es, las K variables de selección corresponde a la combinación binaria de  $K = 2$ , o también conocido como palabra de control, de dos dígitos, dando un total de  $2^k$ ;  $2^2 = 4$  operaciones diferentes como se puede intuir y cumpliendo el requisito planteado por la práctica. Con lo anterior, la entrada de selección puede otorgar dos operaciones cuando  $S1 = 0$ , y otras dos cuando  $S1 = 1$ , y traduciéndose esto como una asignación de la unidad a utilizar dependiente de la ya mencionada entrada de selección S1. Vale acotar que algunos autores (D. Gajski.) prefieren distanciar a partir de su utilidad (selección de unidad y selección de operación) la nomenclatura en el nombre de entradas de selección que dividen las unidades con el nombre de "Entrada de modo"; así, a partir de ahora la entrada S1 será referida de ese manera. En todo caso, la asignación de las operaciones se suman en la Tabla 1:

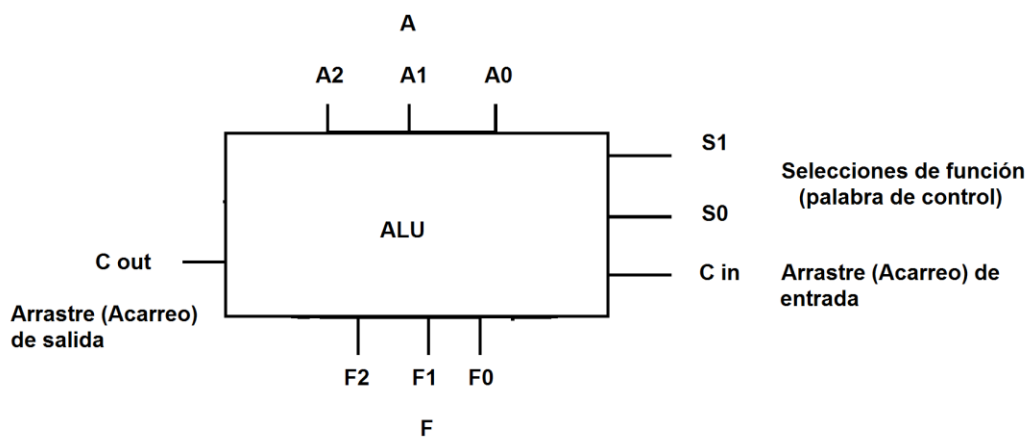
Palabra de control			Salida	Función	Tipo
S1	S0	C in			
0	0	X	$F = \bar{A}$	Complemento	Lógica

0	1	0 / 1	$F = A \leftarrow \text{shl } A$	Rotación	Lógica
1	0	X	$F = A - 1$	Decremento	Aritmética
1	1	X	$F = A + 1$	Incremento	Aritmética

**Tabla 1.** Asignación de operaciones según la palabra de control

La Fig. 1 muestra el diagrama de bloque de la ALU de 3 bits correspondiente a la práctica. Es usual que en el diseño de una ALU se tenga dos términos que se combinan según las operaciones a ejecutar; no obstante, para este caso se trabajará con un único término: tres entradas de datos de A, que tendrá sus operaciones generadas en las salidas F. Los terminales de selección de función S1 y S0 especifican la operación aritmética o lógica a generar según las diversas combinaciones de la palabra de control ya especificados anteriormente.

Los arrastres de entrada y salida sólo tienen significado durante la operación de rotación y se explicará en su apartado correspondiente. Originalmente los arrastres tienen funcionalidades más diversas como el uso a menudo del arrastre de entrada como parte de las variable de selección para doblar el número de operaciones aritméticas posibles a realizar.



**Fig.1** Diagrama de bloque de la ALU

Para diseñar correctamente una ALU típica, Morris-Mano sugiere tres etapas: "Primero, se emprende el diseño de la sección aritmética. Segundo, debe considerarse el diseño de la sección lógica. Finalmente, deberá modificarse la sección aritmética de manera que puedan realizarse ambas operaciones aritméticas y lógicas". Claramente, se seguirá en la medida de lo posible estos lineamientos, modificando la metodología según los requisitos especiales del diseño que satisfaga a esta práctica.

Así mismo, la Fig. 2 ilustra el diseño básico superior del funcionamiento combinatorio de la ALU que, dependiendo del bit de selección en el multiplexor será la unidad a activarse. En caso de esta práctica será la variable en el terminal de selección S1 quien marque la distinción. Cuando la entrada de modo  $S1 = 0$  se selecciona la función de tipo lógica, pero cuando  $S1 = 1$  será de tipo aritmética la salida seleccionada.

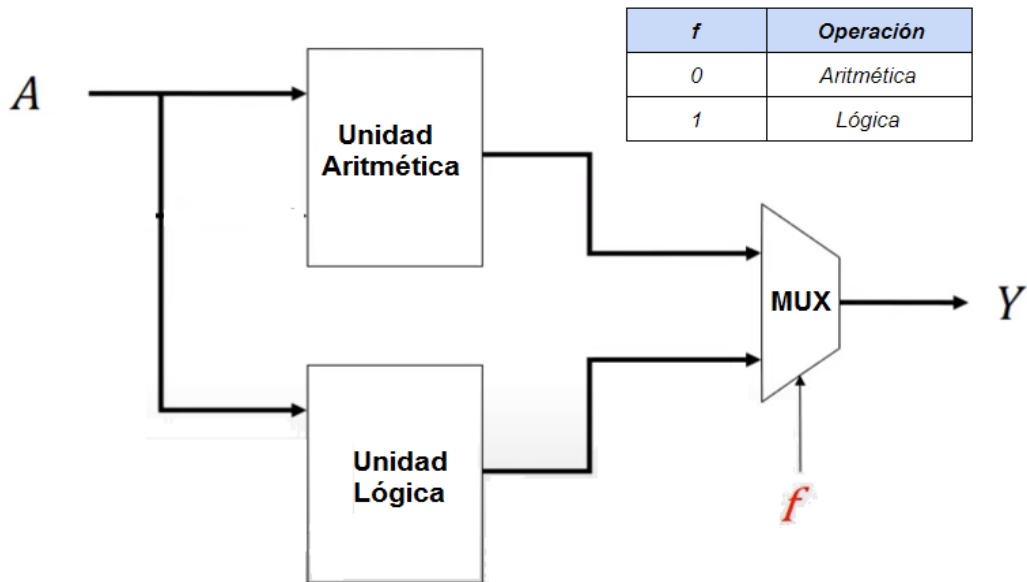


Fig 2. Diseño superior de la ALU

## 1. Diseño de las partes

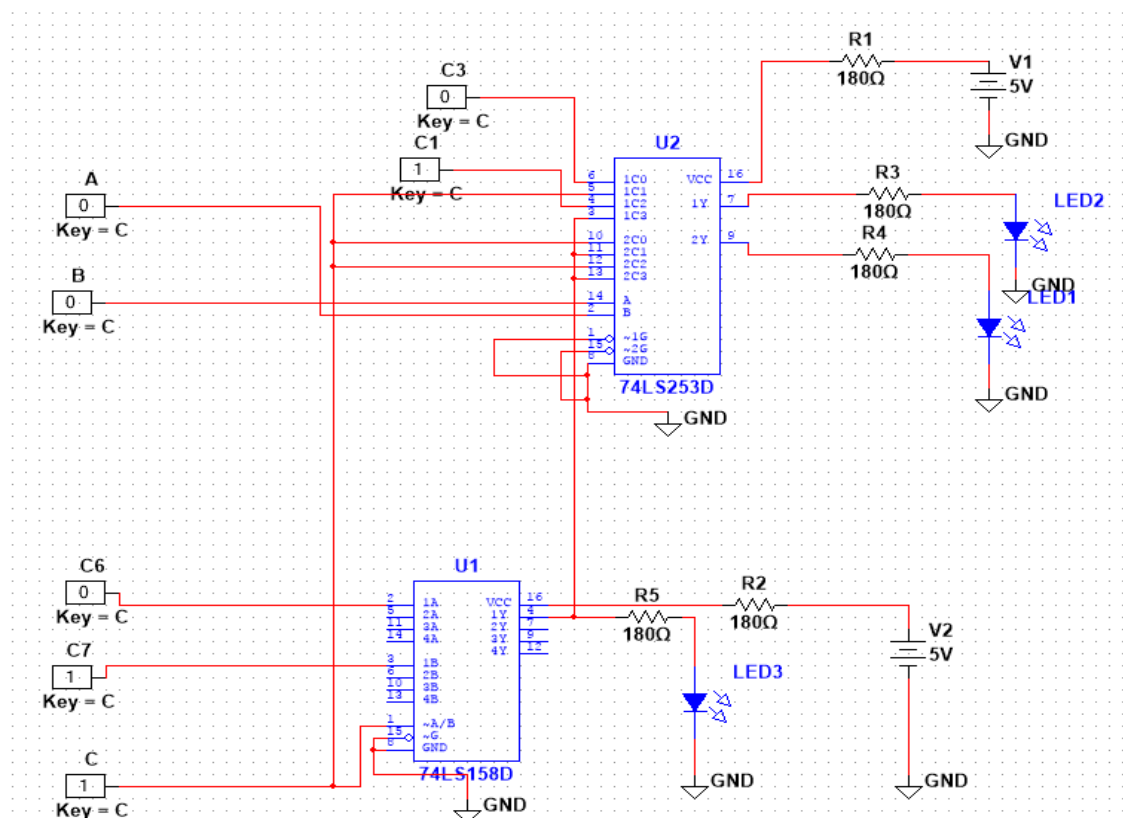
Es importante destacar que, los diseños presentados de incremento, decremento y complemento fueron implementados a través de multiplexores 74LS253 y 74LS158, donde los selectores A y B del MUX 74LS253 controlarán y/o seleccionan las entradas de las secciones A y B del MUX dependiendo de la entrada AB. Si  $AB=00$ , entonces se seleccionan las entradas "I0", si  $AB=01$  corresponde a las entradas "I1", si  $AB=10$  selecciona las entradas "I2" y finalmente si  $AB=11$ , selecciona las entradas "I3", de esta forma el MUX 74LS253 maneja las salidas "Za" y "Zb" correspondientes. En cuanto al MUX 74LS158, controlará y/o maneja la entrada "C" para realizar un complemento y generar la salida "Zc" del circuito. Ahora a continuación se describe y diseña las operaciones individualmente que formarán parte de la ALU:

### 1.1 DISEÑO DEL INCREMENTO (Unidad Aritmética)

En la tabla 1.1.1, por cada par de combinaciones ABC, automáticamente se configura las entradas del multiplexor dependiendo de los valores AB introducidos en los selectores y su salida correspondiente en incremento. Para el primer par de combinaciones de ABC (000 - 001), tenemos que el bit A y Za se mantiene en cero, por lo tanto se ancla un "0" en la entrada I0 de A, de tal manera que seleccione un "0" para Za en ambos casos. Ahora para el caso de las salidas Zb estará asignado con el estado de "C" debido a que en la salida Zb réplica los estados de "C", por lo tanto se asigna a la sección de "B" en la I0 el estado lógico de "C". Para el segundo par de combinaciones ABC (010 - 011), tenemos que para ambas salidas "Za" poseen los mismos estados que en "C", por lo tanto, se asigna el estado lógico de "C" para ambas salidas y en cuanto a la salida de Zb, vemos que la salida Zc es la única donde se puede captar los mismos estados, por lo tanto se asignan estos estados en la sección "B" I1 del MUX, de tal forma que el selector pueda seleccionar dichas estados en las salidas. Ahora tenemos el tercer par de combinaciones de ABC (100 - 101), donde podemos visualizar que en Za para ambos casos representa un "1", por lo tanto, se asigna en la sección "A" I2 del MUX ambas en "1" para satisfacer esta salida respectiva y en cuanto a Zb, tenemos que para ambos casos las salidas son iguales que "C" por lo tanto se asigna en sección "B" I2 el valor respectivo de la entrada "C", cumpliendo así las salidas requeridas. Cabe destacar, el cuarto par de combinaciones ABC (110 - 111) donde ninguna de las entradas ABC puede satisfacer las salidas para estos casos, entonces se opta por verificar la salida Zc, y efectivamente cumple con los mismos estados para las salidas Za y Zb, por lo tanto se asigna "Zc" para ambas secciones: "A" y "B" en I3, de tal forma, se puede obtener los estados "1" y "0" respectivamente para ambas columnas Za y Zb en este último par de combinaciones. Por último, para poder generar las salidas Zc correspondientes para todas los pares de combinaciones ABC antes mencionados, se puede apreciar que todos los estados de la columna de "C" es el complemento de "Zc", entonces si en "C" representa un estado lógico bajo "0", en "Zc" es alto "1" y si "C" representa un estado lógico alto "1", en "Zc" sería bajo "0", generando así las configuraciones para la operación incremento.

	A	B	
--	---	---	--

A	B	C	I0	I1	I2	I3	I0	I1	I2	I3	Za	Zb	Zc
0	0	0	0				C				0	0	1
0	0	1	0				C				0	1	0
0	1	0		C				Zc			0	1	1
0	1	1		C				Zc			1	0	0
1	0	0			1				C		1	0	1
1	0	1			1				C		1	1	0
1	1	0				Zc				Zc	1	1	1
1	1	1				Zc				Zc	0	0	0



## 1.2 DISEÑO DEL DECREMENTO (Unidad Aritmética)

En la tabla 1.2.1 podemos visualizar la configuración del circuito por cada par de combinaciones ABC. Para el primer par de combinaciones de ABC (000 - 001) tenemos que las columnas de ABC no puede representar las salidas Za y Zb, por lo que se tiene en consideración la salida "Zc" como única opción que replica el comportamiento de los estados en las salidas "Za" y "Zb", por lo tanto, como estamos en el primer par AB "00", se asigna en las secciones "A" y "B" I0 del MUX la salida "Zc", de esta forma se logra asignar las salidas respectivas "1" y "0" para estas combinaciones. En cuanto al segundo par de combinaciones de ABC (010 - 011), tenemos que para la salida "Za" en ambas combinaciones requiere un estado lógico bajo "0", por lo tanto se asigna en la sección "A" I1 del MUX un "0" para cumplir efectivamente estos casos y para la salida "Zb", tenemos que los estados lógicos de "C" son exactamente iguales que en "Zb" para ambos casos, por lo tanto, se fija "C" en la sección "B" I1 del MUX. Ahora para el tercer par de combinaciones ABC (100 - 101) podemos notar que en la salida "Za" se requiere un estado lógico bajo "0" para la primera combinación y alto "1" para la segunda, si observamos "C" posee estos mismos estados lógicos, por lo tanto, se asigna "C" en la sección "A" I2 del MUX y para la salida "Zb", la única columna de estados que cumple exactamente con las salidas específicas de "Zb" es "Zc", entonces se asigna "Zc" en la sección "B" I2. Para el cuarto par de combinaciones ABC (110 - 111), tenemos que las salidas "Za" para ambos casos requiere un estado lógico alto "1", entonces se asignan "1's" en la sección "A" I3 y para la salida "Zb" se fija "C" en la sección "B" I3, ya que es la única columna de estados que cumple exactamente con ambas salidas para "Zb". Por último, para poder generar las salidas Zc correspondientes para todas los pares de combinaciones ABC antes mencionados, se puede apreciar que todos los estados de la columna de "C" es el complemento de "Zc", entonces si en "C" representa un estado lógico bajo "0", en "Zc" es alto "1" y si "C" representa un estado lógico alto "1", en "Zc" sería bajo "0", generando así las configuraciones para la operación decremento.

			A				B						
A	B	C	I0	I1	I2	I3	I0	I1	I2	I3	Za	Zb	Zc
0	0	0	Zc				Zc				1	1	1
0	0	1	Zc				Zc				0	0	0
0	1	0		0				C			0	0	1
0	1	1		0				C			0	1	0
1	0	0			C				Zc		0	1	1
1	0	1			C				Zc		1	0	0
1	1	0				1				C	1	0	1
1	1	1				1				C	1	1	0

**Tabla 1.2.1** Cálculo para asignación de entradas



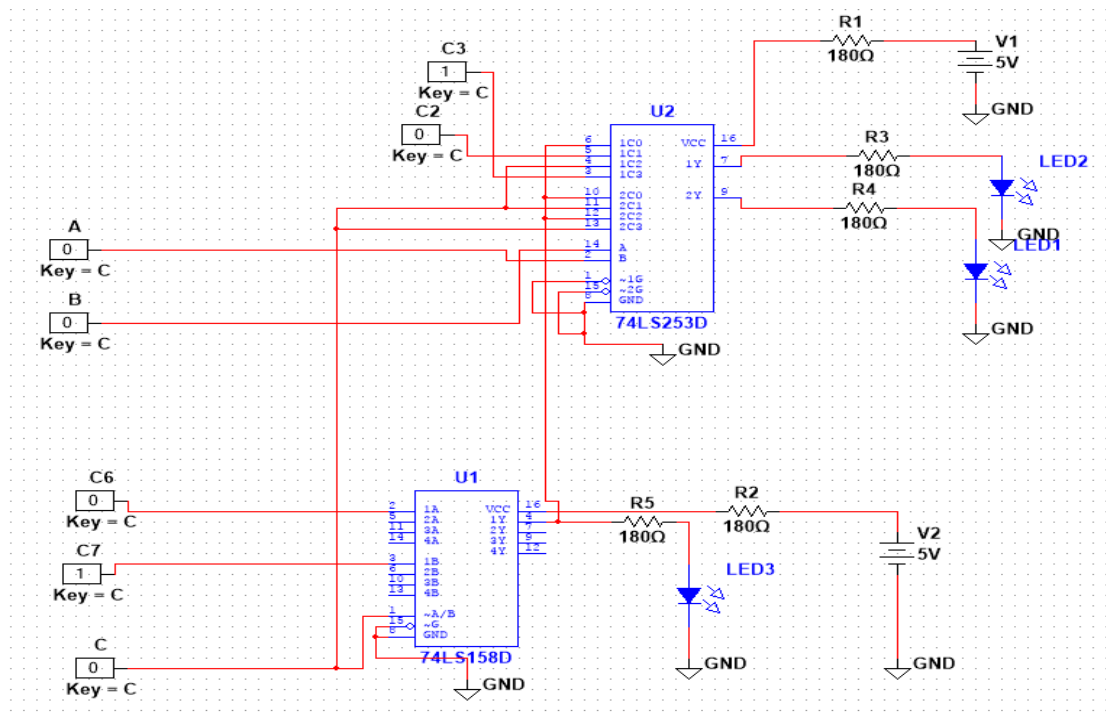


Fig. 1.2.2 Diseño real para el sistema de decremento.

### 1.3 DISEÑO DE COMPLEMENTO (Unidad lógica)

En la tabla 1.3.1 se muestran las configuración para el circuito de la operación complemento, básicamente para el primer par de combinaciones ABC (000 - 001) se asignan en la sección "A" y "B" en I0, estados lógicos alto "1" para satisfacer "Za" y "Zb". Para el segundo par de combinaciones ABC (010 - 011), la columna de estados de la salida "Za" es exactamente la misma que en "B", por lo tanto, se asigna "B" para ambos casos en la sección "A" I1 del MUX y para la salida "Zb" se fija "A" en la sección "B" I1, que es la columna de estados que satisface esta salida. En cuanto al tercer par de combinaciones (100 - 101) tenemos que la columna de estados "B" cumplen con la columna de estados "Za" y "A" cumplen con la columna de estados de "Zb", por tanto se asigna "B" en la sección "A" I2 y "A" en la sección "B" I2. Por último, el cuarto par ABC (110 - 111), en las salidas "Za" y "Zb" se requiere un estado lógico bajo "0", por lo tanto, simplemente se asigna en la sección "A" y "B" del MUX "0's" en I3, de esta manera generar dichas salidas. Con respecto a "Zc", es básicamente lo mismo que en diseños pasados, si "C" posee un estado lógico alto "1", en "Zc" se representará bajo "0" y si en "C" es un estado lógico bajo "0", la salida "Zc" será alto "1", cumpliendo así de esta forma la operación complemento.

			A				B						
A	B	C	I0	I1	I2	I3	I0	I1	I2	I3	Za	Zb	Zc
0	0	0	1				1				1	1	1
0	0	1	1				1				1	1	0
0	1	0		B				A			1	0	1
0	1	1		B				A			1	0	0
1	0	0			B				A		0	1	1

1	0	1			B				A		0	1	0
1	1	0				0				0	0	0	1
1	1	1				0				0	0	0	0

Tabla 1.3.1. Cálculo para asignación de entradas

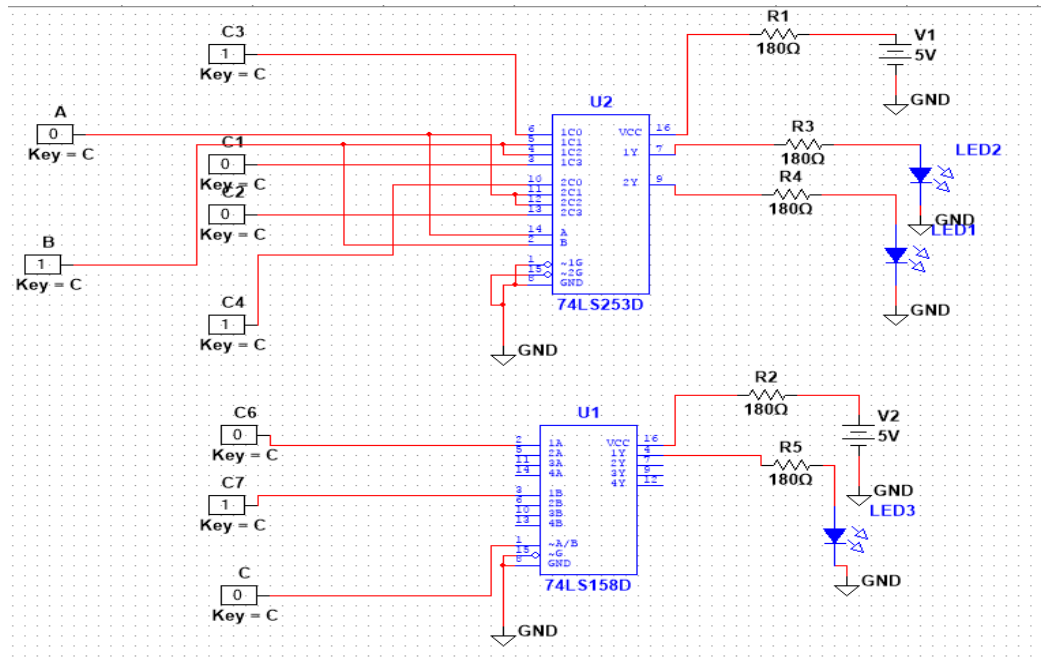


Fig. 1.3.2 Diseño real para el sistema de complemento.

## 1.4 DISEÑO DE LA ROTACIÓN (Unidad lógica)

La unidad de rotación o de desplazamiento transfiere la salida de la ALU al bus de salida. La unidad de desplazamiento puede transferir la información directamente sin un desplazamiento o puede desplazar la información a la derecha o a la izquierda, siendo este último el caso que satisfaga la práctica. Un circuito obvio para este registro de desplazamiento pasa por modificar un registro de desplazamiento bidireccional con carga en paralelo hacia uno unidireccional izquierdo. Igualmente, esta configuración prescindirá del pulso de reloj para la transferencia al registro y del pulso para desplazamiento. Como acotación, los pulsos se encargan de transferir secuencialmente los eventos de desplazamientos con varias iteraciones que funciona junto a los sistemas de memoria mediante flip flops.

La Figura 1.4.1 muestra, de manera esquemática, el funcionamiento del sistema de rotación a la izquierda con acarreo.

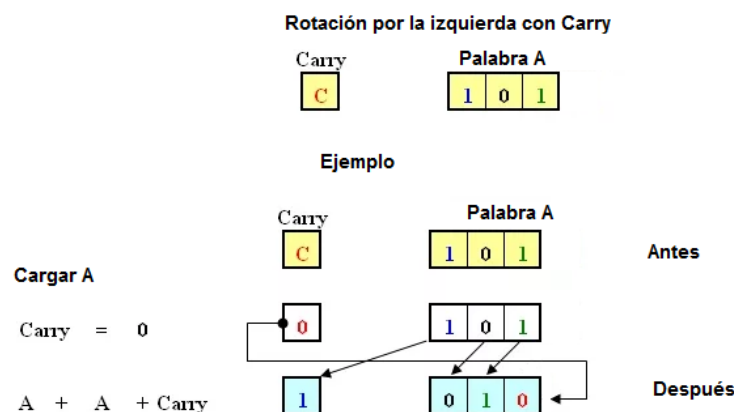
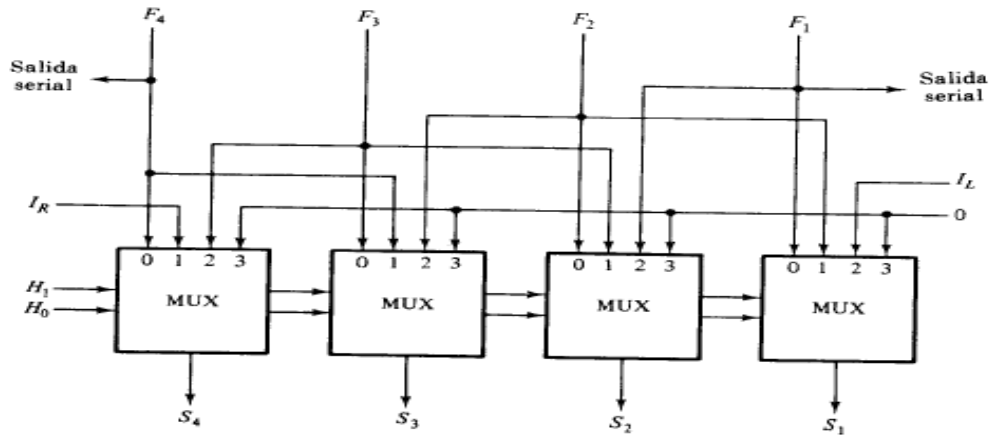


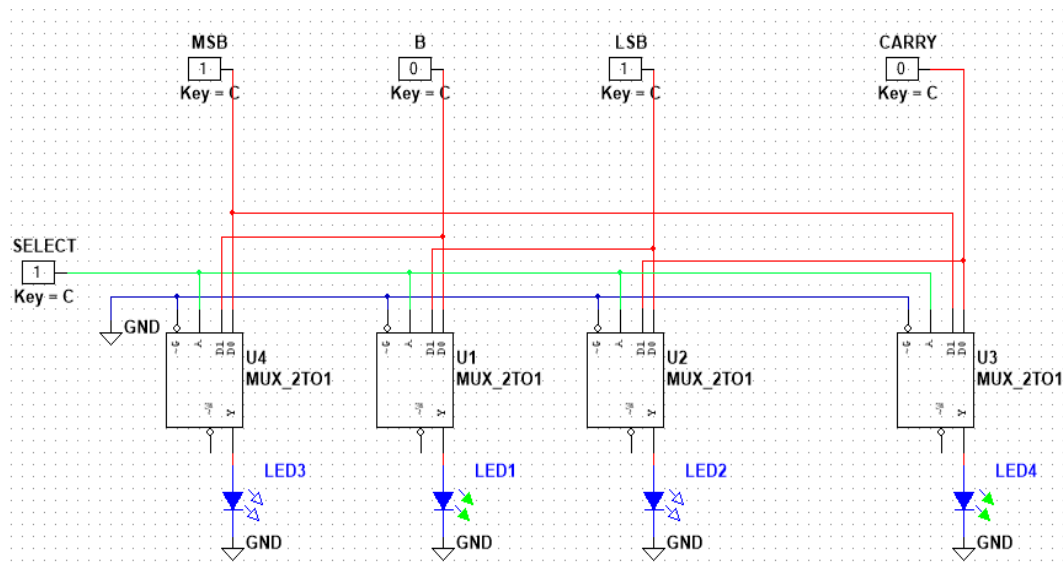
Fig. 1.4.1 Esquema del funcionamiento de rotación de bits por la izquierda con acarreo

El registro de rotación será realizado mediante un circuito combinacional, construido mediante multiplexores, como ha sido constante en cada una de las fases aritméticas y lógicas previas. La Fig 1.4.2 muestra el esquema básico ideal para la construcción del registro - tomado del Morris-Mano-. El esquema presenta unas variables de selección  $H_1$  y  $H_0$  aplicadas a los cuatro multiplexores que seleccionan qué tipo de desplazamiento se aplicaría; dado que la práctica sólo requiere de una única operación, se prescindirá de esas entradas.



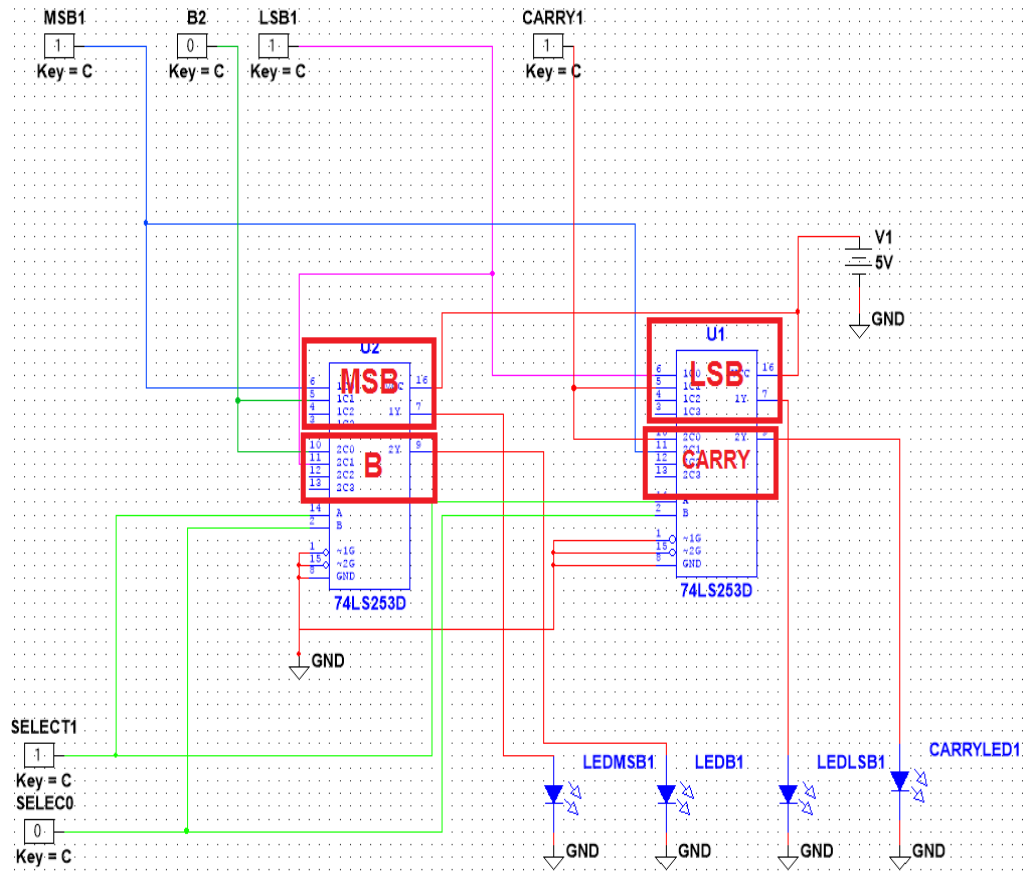
**Fig. 1.4.2** Registro de desplazamiento de 4 bits a base de lógica combinacional

El traspaso a la simulación del diseño en la Fig. 1.4.2 toma en consideración los tres bits de la palabra más un bit de acarreo, dando un total de cuatro estados para el registro de desplazamiento. La palabra A entra en el sistema, en lectura de izquierda a derecha, con el bit más significativo hasta el menos significativo, dejando el cuarto y último estado para el bit de acarreo. El funcionamiento general del sistema es sumamente sencillo, con cada multiplexor conectado a sus subsiguientes laterales configurado para recibir el valor anterior y otorgar el propio al posterior, esto es, la entrada de selección de los multiplexores de la palabra ajustados a 1. De esa manera la palabra se desplaza una posición a la izquierda.

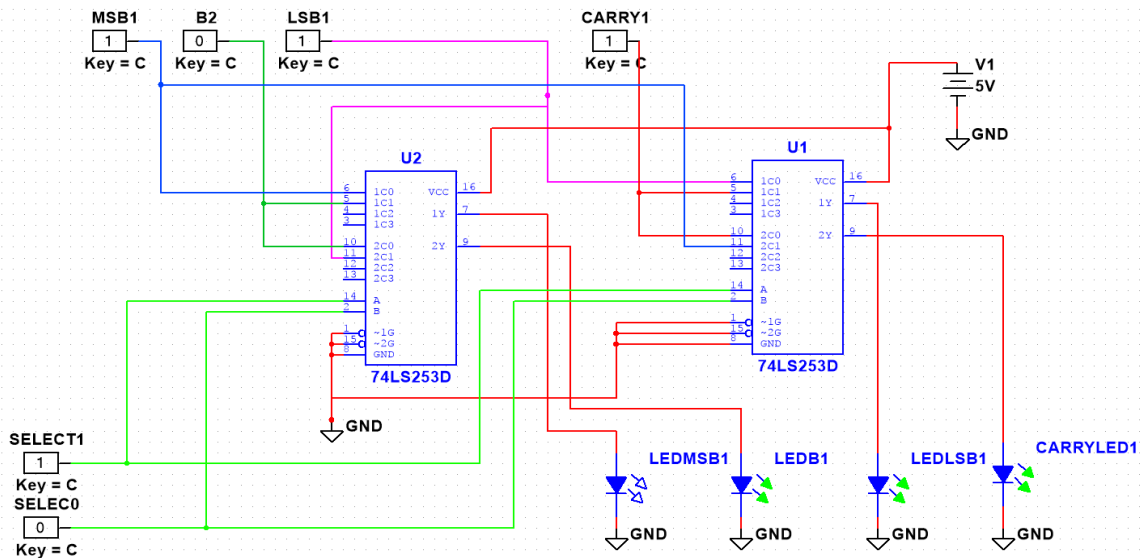


**Fig. 1.4.3** Diseño ideal de registro de desplazamiento de 3 bits + Acarreo para la ALU

Para la posterior traslación simulando los dispositivos reales continúa con el uso de multiplexores 4 a 1 del modelo 74ls253 (datos de referencia consultables en el apartado de Materiales y equipos). Para simplificar la visualización de este traslado, en comparación con el mostrado en la Fig. 1.4.3, puede consultarse la Fig. 1.4.4 donde se muestra a qué bit está designada esa parte de cada multiplexor. Por su parte, las entradas de selección de ambos multiplexores fueron configurados de la siguiente manera:  $S_0 = 1$  (Pin 14) y  $S_1 = 0$  (Pin 2), dándole prioridad a la entrada de datos de  $I_1$ , esto es, pin 5 y pin 11 (Fig. M2), ajustados de esta manera a recibir el dato inmediato anterior y terminando en un ciclo entre el bit más significativo y el acarreo. El diseño final puede consultarse en la Fig. 1.4.5.



**Fig. 1.4.4** Diseño real simulado de registro de desplazamiento de 3 bits + Acarreo para la ALU



**Fig. 1.4.5** Diseño real simulado de registro de desplazamiento de 3 bits + Acarreo para la ALU

## 2. Construcción en una única unidad y optimización

Volviendo a la Fig 2 mostrada al comienzo del desarrollo, una ALU funcional que pretenda permitir la selección de las distintas operaciones debe ser controlada mediante un multiplexor posterior que, mediante una palabra de control, active una u otra unidad. En orden para que esto se cumpla, se decidió hacer uso de dos multiplexores 4 a 1 74ls253. Las salidas de cada operación son dirigidas a las entradas de estos multiplexores y que, ingresando en las entradas de selección la palabra correspondiente a la operación, mostrará el resultado en los LED finales. Los LED, de manera intuitiva, están ordenados siguiendo que el Bit-más-significante y Bit recibirán la señal dada por el multiplexor superior, mientras que el Bit-menos-significante, y un último LED que representa

el acarreo (únicamente utilizado en la operación de rotación) recibirán la señal proveniente del multiplexor inferior. Las salidas de cada operación es dirigida a las entradas siguiendo el funcionamiento de la tabla de verdad de los multiplexores (Fig. M3), esto es, y para ser más claro con un ejemplo conciso, la primera operación será conectada a los pines 6 y 10, así cuando las entradas de selección sean 00 en ambos dispositivos, permitirá la salida de esa operación y finalmente los LED encenderán en su función. Como nota, en las figuras de los diseños cada operación es visiblemente representado por un color diferente siendo rojo: complemento, azul: rotación, verde: decremento, amarillo: incremento. Todo lo anterior es apreciable en la Fig. 2.1 y en la Fig. 2.2.

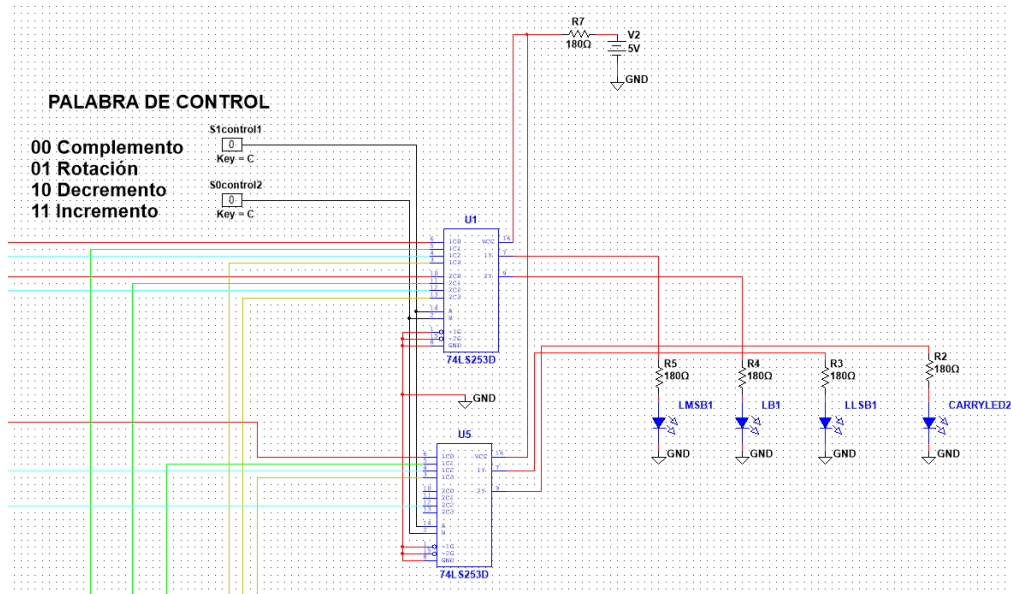
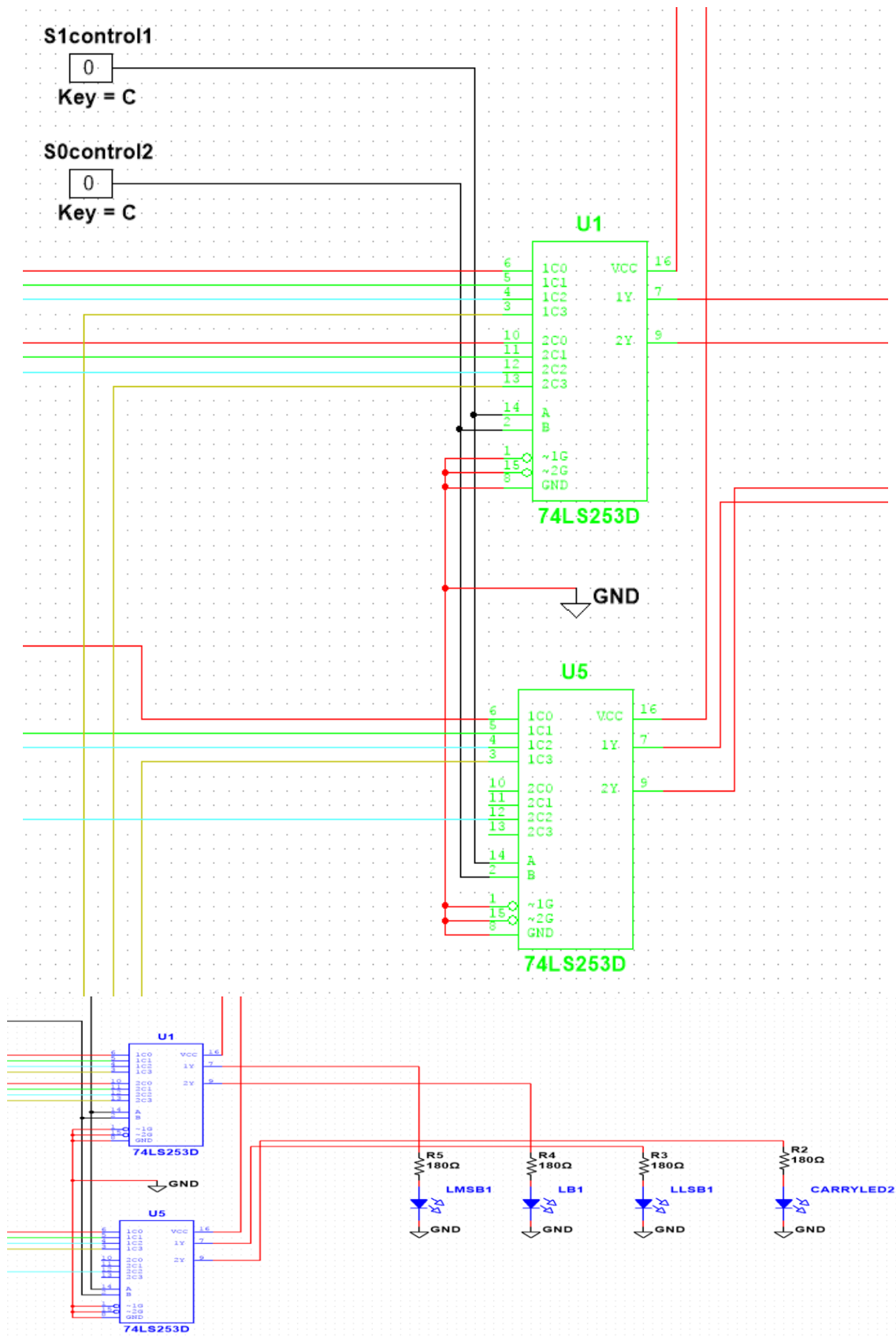


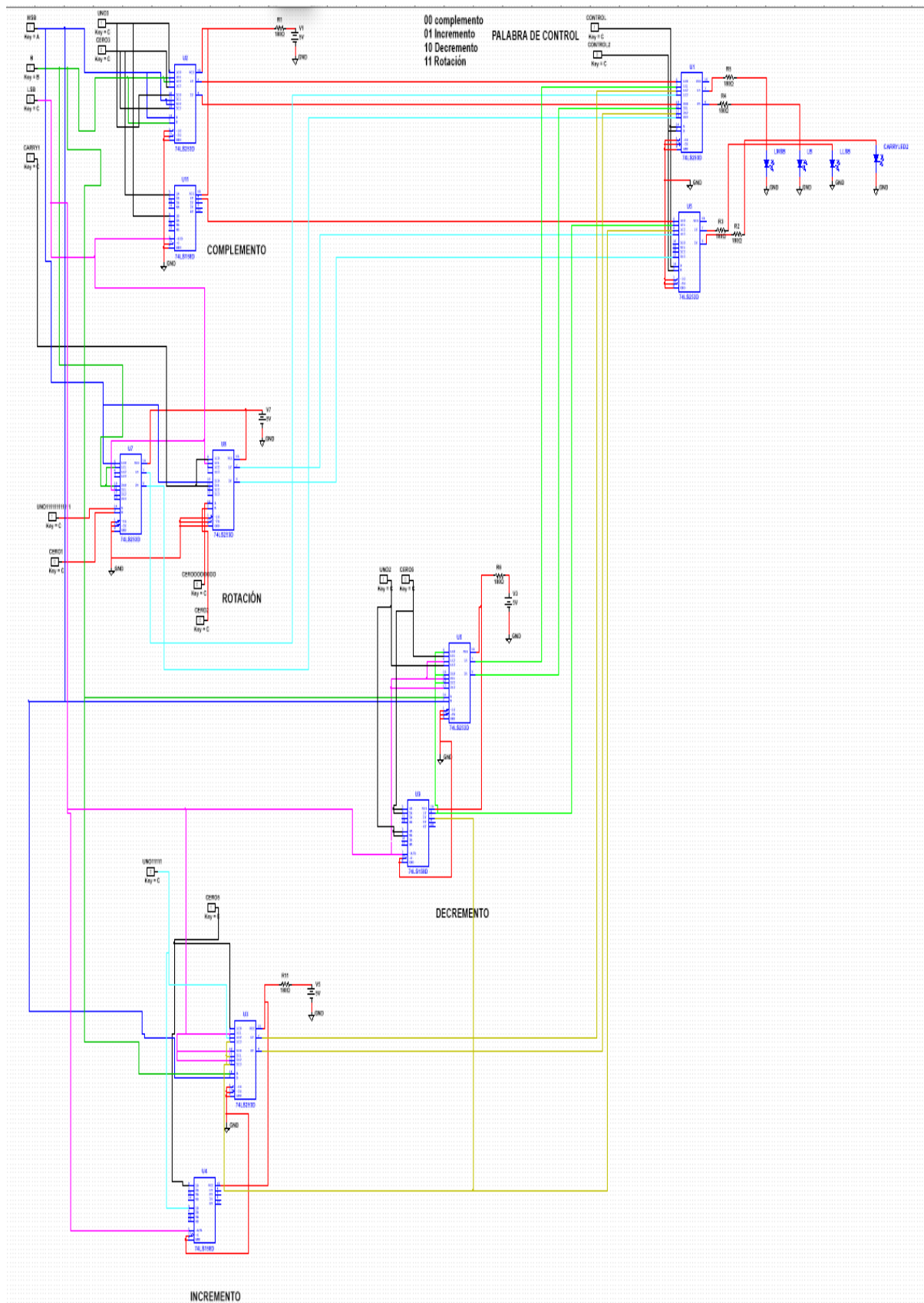
Fig 2.1 Multiplexores para palabra de control



**Fig 2.2 Multiplexores para palabra de control y LED's (Vista ampliada)**

Por su parte, las operaciones se ordenaron según lo ya establecido en la Tabla 1. Las entradas se redujeron a las tres pedidas para la palabra más acarreo y en igual sentido se interconectan todas las entradas estáticas de 1's o 0's (esto último observable en la Fig 2.4). La Fig 2.3 muestra en definitiva esta primera aproximación funcional de la ALU ya como una única unidad, pero la cual todavía carece de refinamiento.

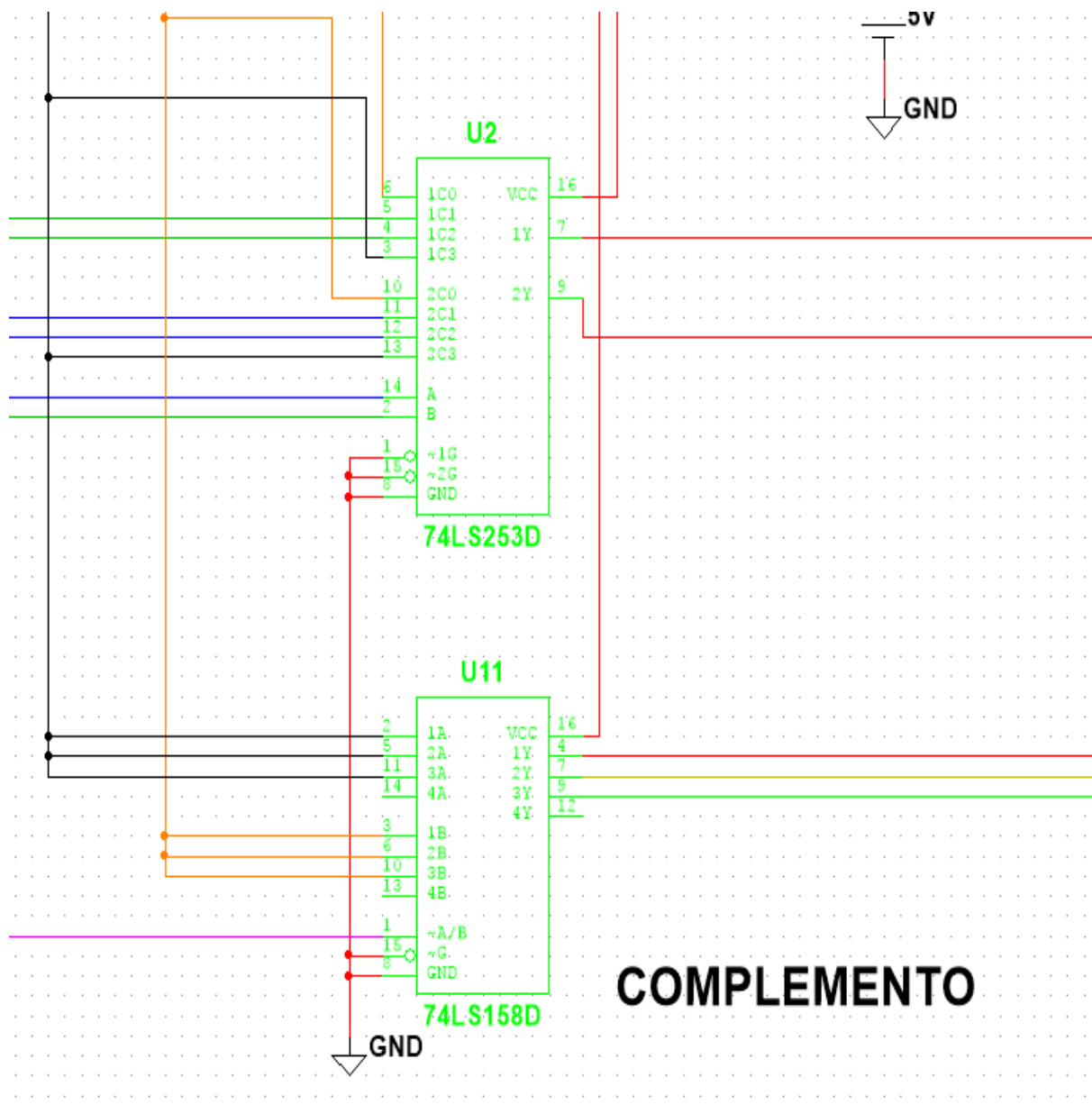




**Fig 2.3** ALU primera interconexión funcional completa de la ALU.

Para resolver lo anterior y siguiendo los lineamientos planteados al inicio de la práctica (haciendo hincapié en los pasos de construcción sugeridos por Morris-Mano) donde se establece que una ALU es un problema de

lógica combinacional y debido a que la unidad tiene un patrón regular, ésta puede fraccionarse en etapas idénticas conectadas mediante diversas maneras como puede ser el arrastre de acarreo o mediante las entradas de selección, siendo esta última la manera en que se trabajó esta práctica. Algo que queda claro es que una ALU más eficiente es aquella que “entremezcla” sus unidades y permite que las operaciones lógicas sean realizables con ayuda de la unidad aritmética. El tercer paso de construcción en el diseño no es un procedimiento directo ni absoluto para todos los diseños, pues depende de la cantidad de IC, las conexiones y de, por supuesto, las operaciones y su jerarquía en el entorno. Para el caso de esta ALU, favorecida por una construcción funcional basada únicamente por multiplexores, la reutilización de estos dispositivos es el pensamiento más directo para la reducción del sistema. En este orden de idea, un detalle que destaca bastante corresponde al multiplexor 74ls158 presente en las operaciones de complemento, incremento y decremento. En todos los casos son infrutilizados dedicándose a una única actividad, desaprovechando sus otras entradas. Así, se tomó la decisión final de unificar estos multiplexores en uno único configurado para recibir las diferentes entradas correspondientes (0 en las 2, 5 y 11 ; 1 en las 3, 6 , 10) y otorgando finalmente las salidas que serán parte de las tomadas por los multiplexores finales controladores de la ALU. Todo lo anterior es observable en la Fig 2.4. Finalmente, la Fig. 2.5 muestra el diseño final eficaz reducido para la ALU.



**Fig 2.4** Enfoque en asignación de un único multiplexor 74ls158 para las operaciones de complemento (cable rojo), incremento (cable amarillo), decremento (cable verde).

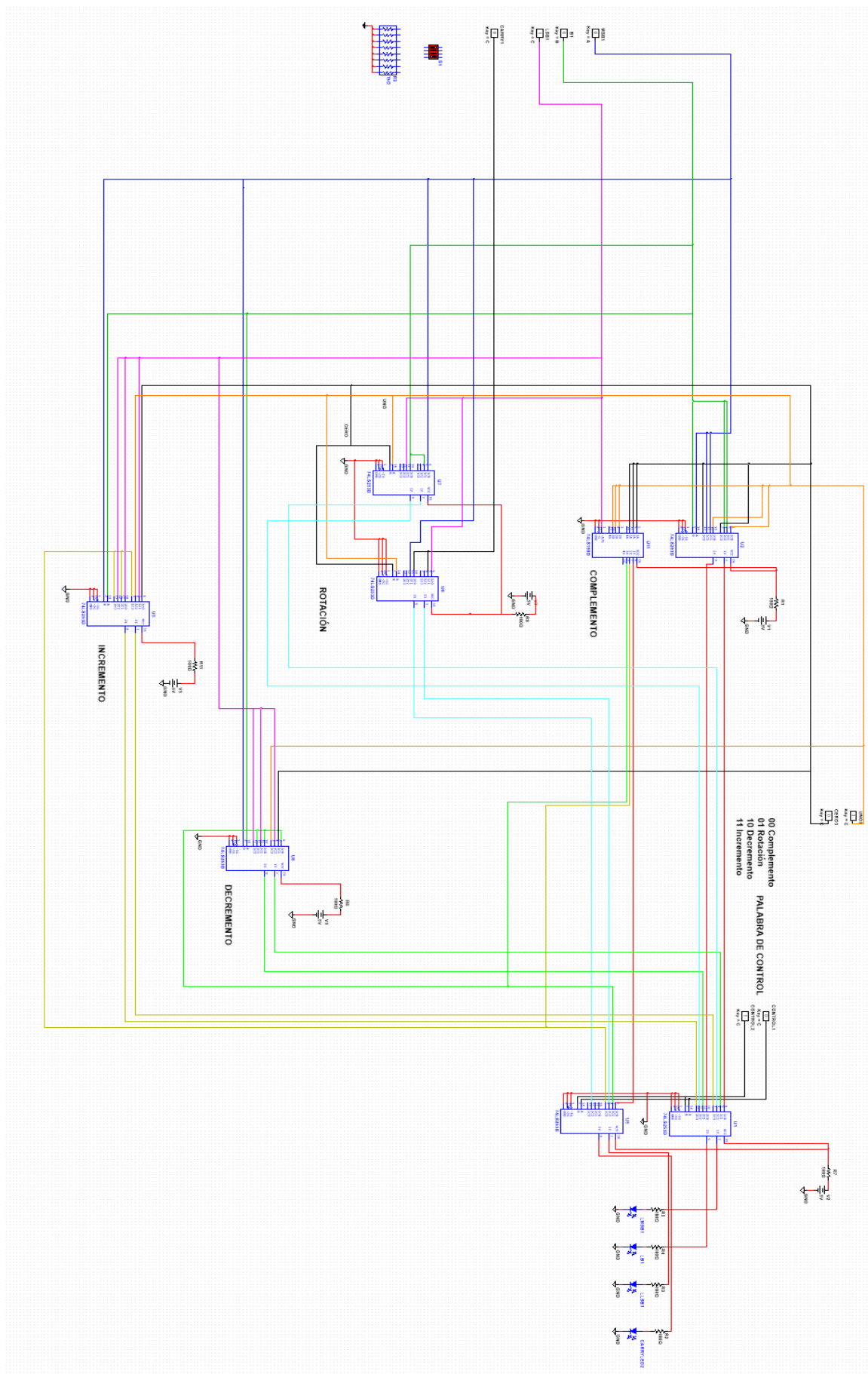


Fig 2.5 Diseño final reducido de la ALU.

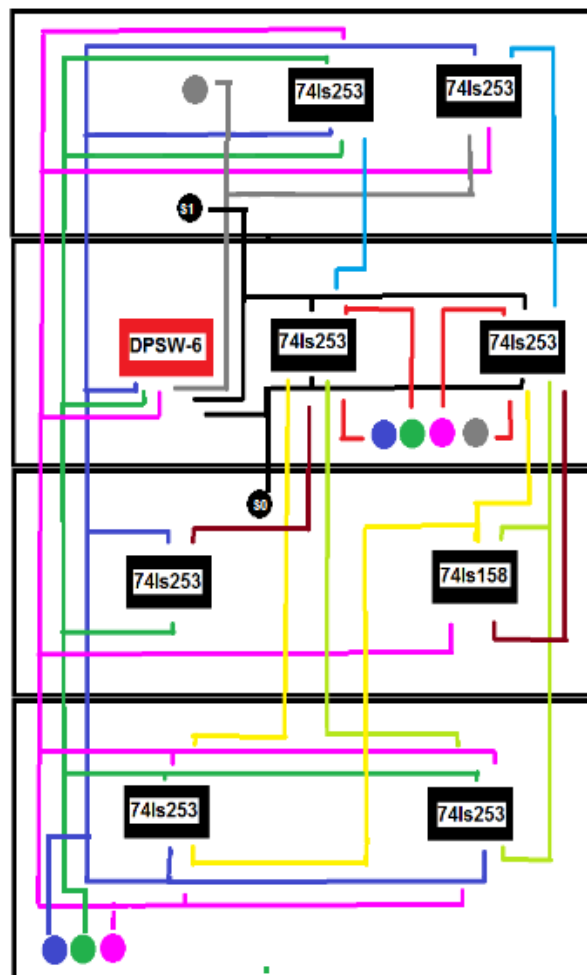
### 3. Traslado a protoboard

Una vez se obtuvo el diseño final óptimo para la ALU la siguiente interrogante responde a su aplicación en un protoboard. Así, teniendo en consideración el tamaño del circuito y las condiciones propias de la cuarentena actual se hizo uso de las herramientas de las que dispone el simulador Multisim para trabajar con un protoboard 3d virtual. De esta manera, la comunicación grupal no se ve reducida y el diseño se beneficia de un ajuste más preciso antes de su instalación en un protoboard real.

El primer paso para llevar el diseño al protoboard pasa por idear una distribución óptima que se adapte a un espacio lo más ajustado posible y de lectura sencilla que, entre otros beneficios, reduzca el consumo de cables y/o su superposición. Como respuesta a lo anterior se llegó a la distribución presentada, en manera simplificada, en la Fig. 3.1. Un diseño que presenta a los multiplexores de las operaciones circundando los multiplexores de salida, LED's y el DipSwitch. La distribución de las operaciones también responde al estado de dependencia entre ellas, sobretodo presente en la reutilización del multiplexor 74ls158 en las operaciones de complemento, incremento y decremento. Se dispone de un protoboard con 4 slats: el superior aloja la operación de rotación y los LED's de representación de entrada de palabra de control S1 y del carry; el segundo, considerado como la parte central donde están tanto las entradas (DipSwitch - lectura de izquierda a derecha de las salidas: MSB, B, LSB, CARRY, Palabra de control S1 "Entrada de modo" y palabra de control S0 "Selección de operación") como las salidas representadas en los LED's, y destacando la parte fundamental de la ALU: los multiplexores de control, que reciben la palabra de control y que deciden qué operación se ejecutará según los parámetros ya pautados anteriormente; por su parte, el tercer slat contiene la operación de complemento, su ubicación responde a la necesidad próxima de comunicación con los mux de salida y con las operaciones con las cuales comparte multiplexor, contiene además el LED indicador de la entrada palabra de control S0; el cuarto y último slat aloja las operaciones de incremento y decremento y ,además, los LED's de representación de entrada de la palabra A.

#### Leyendas

MSB / LED  
B / LED  
LSB / LED  
CARRY / LED  
Rotación OP  
Complemento OP  
Incremento OP  
Decremento OP  
Palabra de control  
Salidas a LED's



**ROTACIÓN /**  
LED's indicadores de  
entrada carry y palabra  
de control S1

**DPSW /**  
**SALIDAS /**  
LED'S de salida

**COMPLEMENTO /**  
LED indicador  
de entrada palabra  
de control S0

**INCREMENTO /**  
**DECREMENTO**

LED's indicadores  
de entrada palabra A

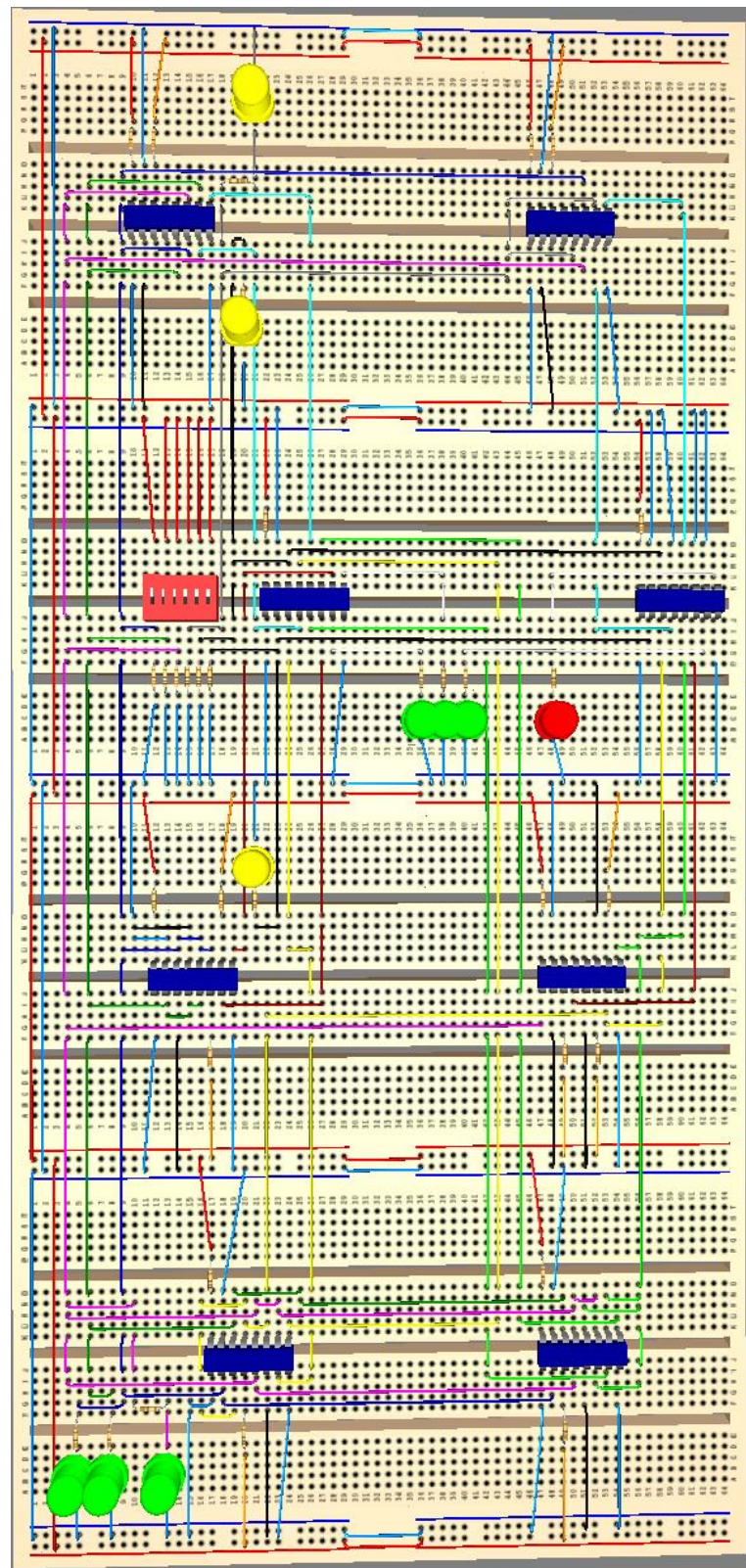
Fig. 3.1 Esquema simplificado de la configuración de la ALU en el protoboard



El traslado completo al protoboard virtual presenta un detalle que vale la pena acotar, y es que debido a lo riguroso en el tamaño de las conexiones de las resistencias se tuvo que añadir dos filas adicionales a cada slat para representar las conexiones a tierra o fuente lo más cercano posible al protoboard real. Las entradas instruidas por el DIP Switch tienen su representación distribuida en LED's acorde con las posiciones de la Fig 3.1 pero alternando los colores, siendo amarillo para la representación del carry y la palabra de control, y verde para la representación de la palabra A. Por su parte, las líneas de funciones tienen colores similares a la de la Fig. 3.1 para facilitar su lectura. A su vez, las conexiones a tierra o fuente son representados por cables de color azul (y negro, cero lógico) o rojo (y naranja, uno lógico), respectivamente. De la misma manera, las conexiones a fuente (5V) son protegidas mediante resistencias de 150  $\Omega$  y las de tierra con las de 330 $\Omega$ , todas ellas asignadas de la siguiente forma: las conexiones a fuente de los IC, así como las que representan las conexiones a uno lógico, utilizarán resistencias protección de 150 $\Omega$ ; por su parte, las resistencia de 180  $\Omega$  encuentran su lugar como protección de los LED's de entrada y salida; y finalmente, los 6 pull down a tierra para las salidas del DipSwitch se realizan mediante resistencias de 330 $\Omega$ .

Por otra parte, si se desea consultar una vista más próxima a cada slat puede remitirse a las Fig. 3.3, 3.4, 3.5 y 3.6

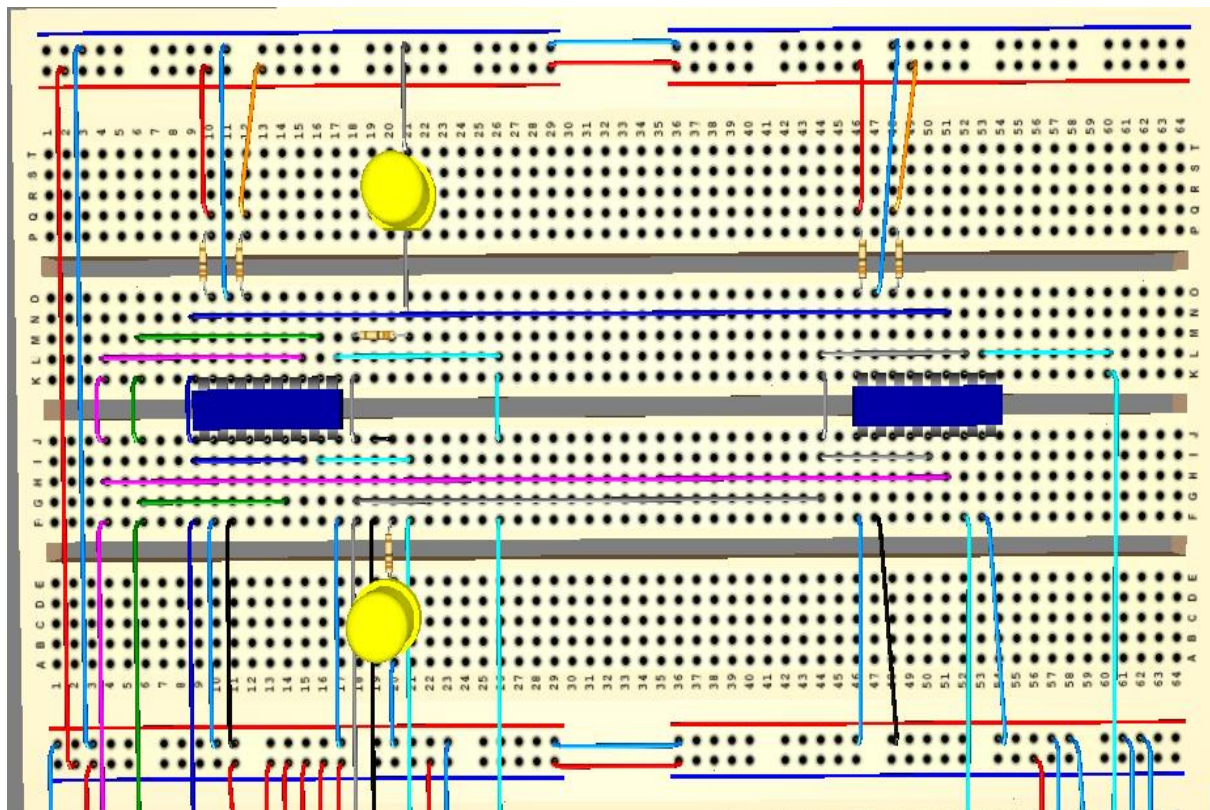
Finalmente, una aproximación y un estudio más detallados sobre la distribución en el protoboard de la ALU será plasmada en el informe correspondiente a esta práctica.



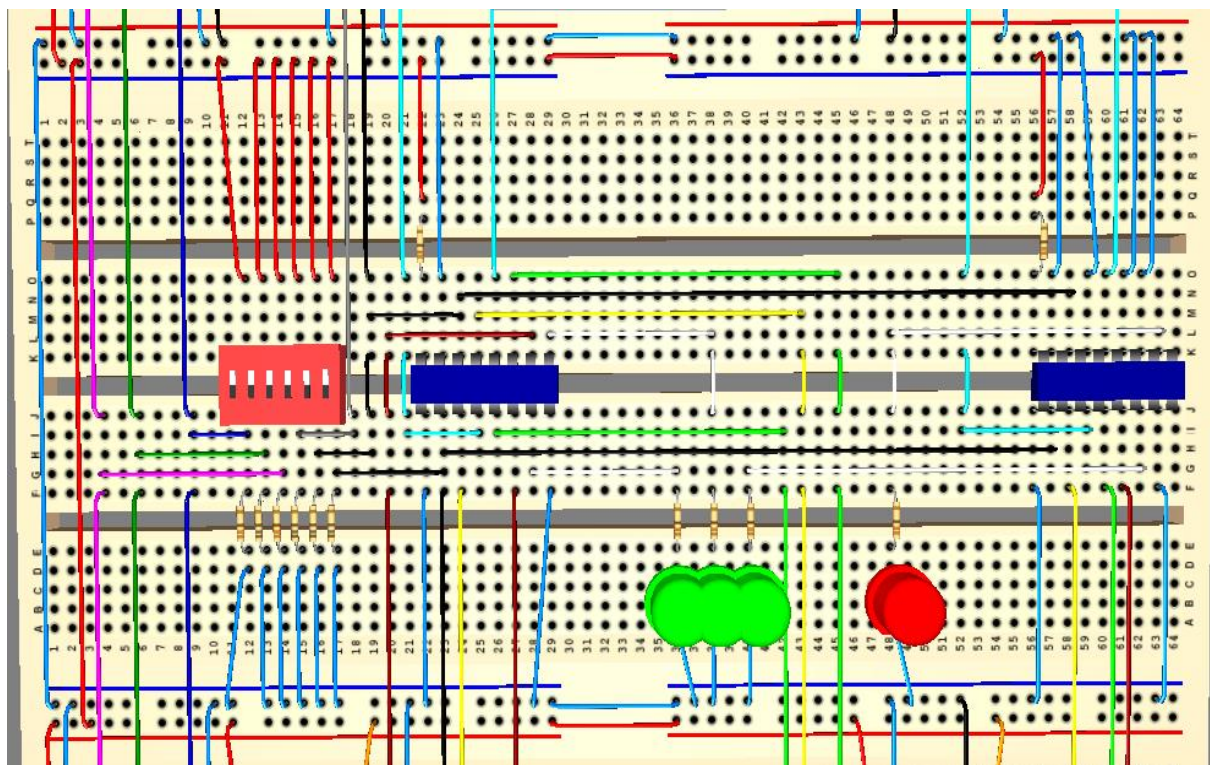
**Fig. 3.2** Distribución en protoboard virtual de

la ALU



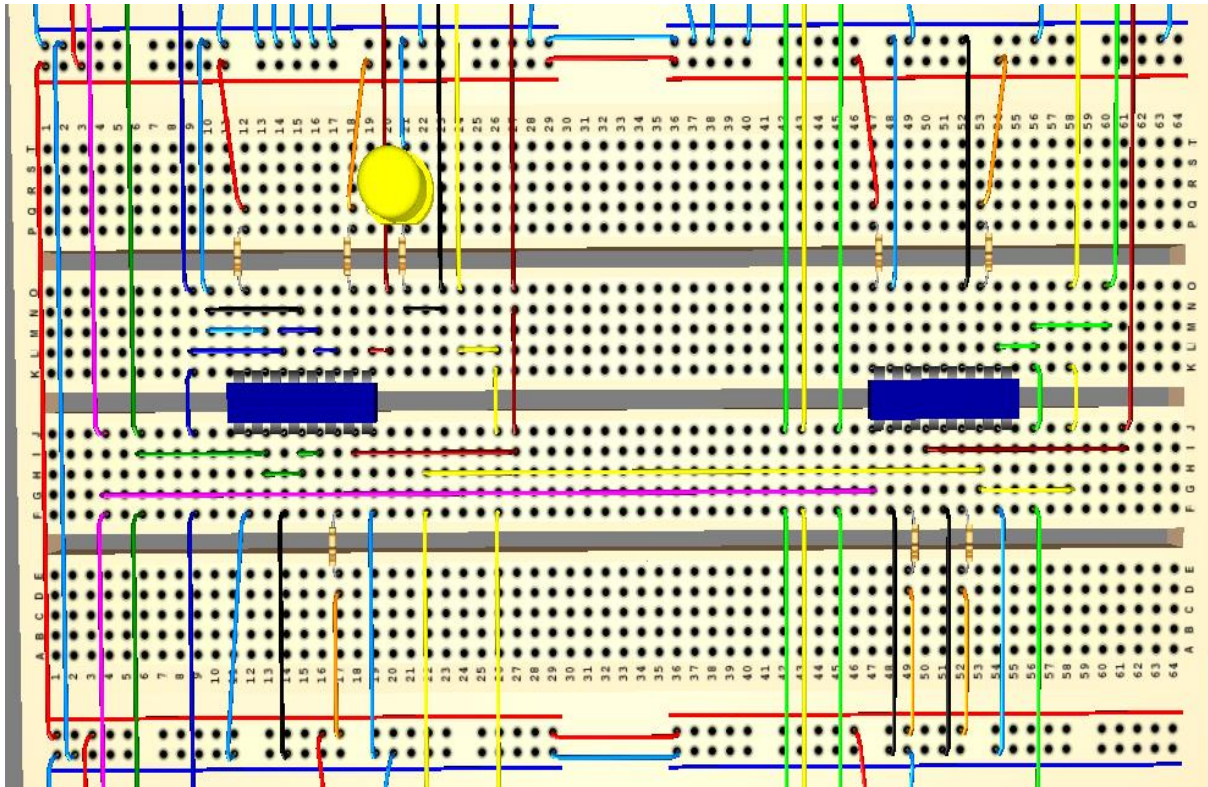


**Fig. 3.3** Distribución del primer slat en protoboard simulado (Operación rotación)

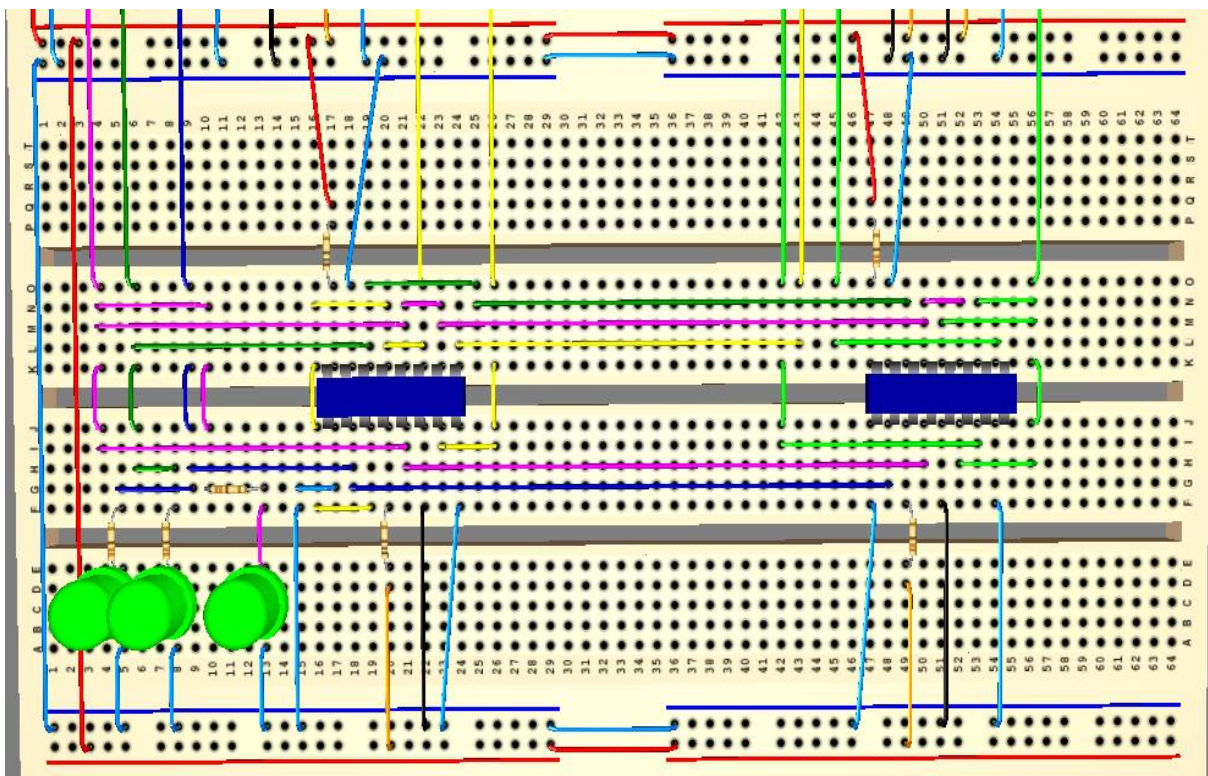


**Fig. 3.4** Distribución del segundo slat en protoboard simulado (DipSwitch de control, Muxes de control y LED's de salida)





**Fig. 3.5** Distribución del tercer slat en protoboard simulado (Operación de complemento; Mux de la derecha 74ls158 también forma parte de las operaciones de incremento y decremento)



**Fig. 3.6** Distribución del cuarto slat en protoboard simulado (Operaciones de incremento y decremento)

## BIBLIOGRAFÍA

-Para la recaudación de información de las bases teóricas se consultaron las siguientes fuentes:

John F. Wakerly "Digital design : principles and practices", Upper Saddle River : Pearson Prentice Hall. 2006

Unidad de codificadores(2020). Retrieved 19 July 2020, from [http://myelectronic.mipropia.com/CICLO\\_2/M11\\_01\\_CODIFICADORES.PDF?i=1](http://myelectronic.mipropia.com/CICLO_2/M11_01_CODIFICADORES.PDF?i=1)

Metodologías para el diseño digital. Universidad de Las Palmas de Gran Canaria, España, 2014.

Daniel D. Gajski. Diseño digital. Edición 1997.

Mano, M., Escalona García, R. and Duchén Sánchez, G., 2003. Diseño Digital. México: Pearson Educación, S.A. Tercera edición.

Técnicas de Diseño, Desarrollo y Montaje de Circuitos. Universidad de Los Andes, Venezuela, 2010.

Multiplexer And Demultiplexer Circuits and Applications. (2020). Retrieved 26 July 2020, from <https://www.elprocus.com/multiplexer-and-demultiplexer/>

Rojas, E. (2020). Reloj Digital. – T-Bem. Retrieved 27 July 2020, from <https://teslabem.com/nivel-intermedio/reloj/>

Vicente, J. (2016) Decodificador 74LS47. Retrieved 16 May 2016 from <https://www.ecured.cu/SN74LS47>

-Para el diseño de la ALU se consultaron las siguientes fuentes:

Mano, M., Escalona García, R. and Duchén Sánchez, G., 2003. Diseño Digital. México: Pearson Educación, S.A. Tercera edición.

Daniel D. Gajski. Diseño digital. Edición 1997.

Trabajos documentales del profesor Dr Craig A. Evans sobre ALU's. Referencia consultable: [https://www.youtube.com/watch?v=chceO4nl9w8&list=PLsHOxmMgAiQwL8X\\_-0KjQLmtKUB67nd3m&index=1](https://www.youtube.com/watch?v=chceO4nl9w8&list=PLsHOxmMgAiQwL8X_-0KjQLmtKUB67nd3m&index=1)