República Bolivariana de Venezuela

Universidad de Carabobo

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica y de Telecomunicaciones

Departamento de Sistemas y Automática

Cátedra de Lógica Digital

**Práctica # 4 – Sesión #01**

**Objetivo: Diseñar e implementar circuitos combinacionales mediante montaje con circuitos integrados de Multiplexores, Demultiplexores y Protoboard**

Sección #05 de Laboratorio

**Integrantes:**

Carlos Hernández

C.I.: 25.829.471

Gianfranco Gasbarri

C.I.: 26.654.860

Fecha de entrega: 05/03/19

**Pre-Laboratorio**

1. **Actividad 2. Implementación de función lógica con multiplexores:** Ver **Anexo 1**
2. **Actividad 3. Implementación de función lógica con demultiplexores:** Ver **Anexo 2**
3. **Actividad 4. Problema.** Ver **Anexo 3**

**Conclusiones**

Se pudo comprobar en la práctica que una de las aplicaciones de utilizar los multiplexores es implementar funciones lógicas. El procedimiento a seguir puede variar dependiendo de la cantidad de entradas del dispositivo con respecto a la cantidad de variables de entrada que posea la función.

Cuando el MUX tiene igual número de entradas que variables de la función, simplemente debemos conectar a Vcc o a tierra cada una de las entradas de datos del dispositivo, tal cual indique la tabla de la verdad. En este caso, resulta mucho menos costoso realizar esta implementación que empleando únicamente compuertas lógicas básicas, tal como se ha estudiado en obras anteriores.

En el caso en el que el MUX tenga menos entradas que variables, se deberá seleccionar las variables de selección de acuerdo a la cantidad que indique el codificador. Una vez hecho esto, se procede a obtener una expresión lógica para cada una de las posibles combinaciones que puedan tomar dichas variables escogidas, esto se hace construyendo tantos Mapas de Karnaugh como combinaciones existan. Finalmente, se arma el circuito de acuerdo a las expresiones obtenidas previamente y se conectan a las correspondientes entradas del multiplexor.

Otra de las aplicaciones de este aparato, es la capacidad de crear circuitos selectores de datos. Si modelamos nuestro problema en particular como una función lógica, podemos crear un selector de datos que de acuerdo a ciertas condiciones (los valores de las variables de selección) podremos escoger un valor u otro. También tenemos la capacidad de controlar si deseamos o no que el dispositivo realice su función para un determinado problema o condición, conectando el **enable** a Vcc.

El otro dispositivo estudiado en el presente informe ha sido el demultiplexor. Con el mismo, tenemos otra alternativa para implementar cualquier función lógica de nuestro interés. Nuevamente, tendremos dos casos de uso de acuerdo a la cantidad de variables residuales. Sin embargo, para cualquiera de las alternativas, debemos tomar una decisión de realizar el diseño en base a los ceros (0’s) o los unos (1’s) de la función.

Si no hay variables residuales, simplemente conectaremos las salidas que corresponden a ceros o unos (dependiendo de nuestra elección) y las agruparemos con una compuerta **AND** o **NAND** (o **NOT** y **OR)** respectivamente.

Cuando existen variables residuales, deberemos obtener expresiones lógicas para cada una de las combinaciones posibles de las variables de control, mediantes Mapas de Karnaugh. Una vez obtenidas, deberemos agruparlas con compuertas **OR** o **AND**, con las salidas asociadas con 0’s o con la salidas invertidas asociadas con 1’s, respectivamente según nuestra elección al momento del diseño. Una vez hecho lo anterior, todas estas combinaciones se combinaran con compuertas **AND** u **OR**, una vez más, respectivamente.

Las implementaciones de las funciones lógicas con multiplexores y demultiplexores tienen características en común, tales como el procedimiento del diseño. Para ambos casos, deberemos escoger unas variables de selección y crear Mapas de Karnaugh para obtener circuitos lógicos asociados con las entradas o las salidas de los dispositivos. Sin embargo, los demultiplexores nos ofrecen una mayor libertad, debido a que nos permiten implementar más de una función lógica a partir del mismo decodificador, cosa que no era posible con su contraparte. Además, podemos escoger si deseamos implementar la función partiendo de los 0’s o los 1´s de la función.

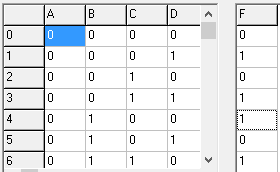
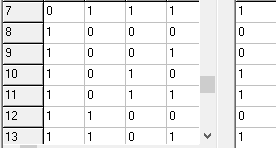
En nuestro problema planteado, utilizamos estos dispositivos y algunos decodificadores que estudiamos en prácticas anteriores. Pudimos dividir el mismo en 3 subproblemas para mayor simplicidad. El primero de ellos, fue el diseño del circuito de selección de los dispositivos de entrada, para el cual utilizamos **4 MUX 4:1**. El segundo, el indicador del dispositivo seleccionado por el anterior diseño a través de un display 7 segmentos. Y el último, el selector de paridad, que indicará si el dispositivo escogido en el primer diseño tiene una cantidad de par de unos en sus entradas, lo cual fue realizado con demultiplexor 1:4.

**ANEXOS**

**Anexo 1**

**Anexo 1.1**

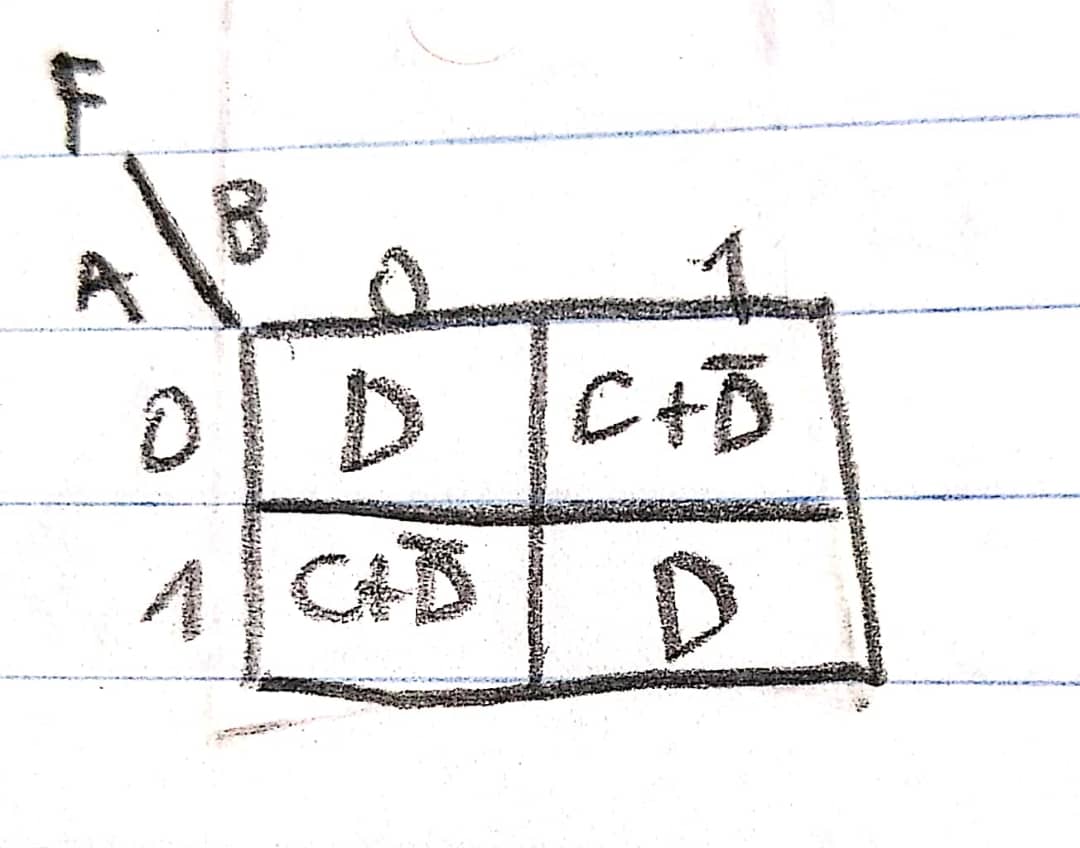
**Tabla de la verdad**





**Anexo 1.2**

**Mapas de Karnaugh**

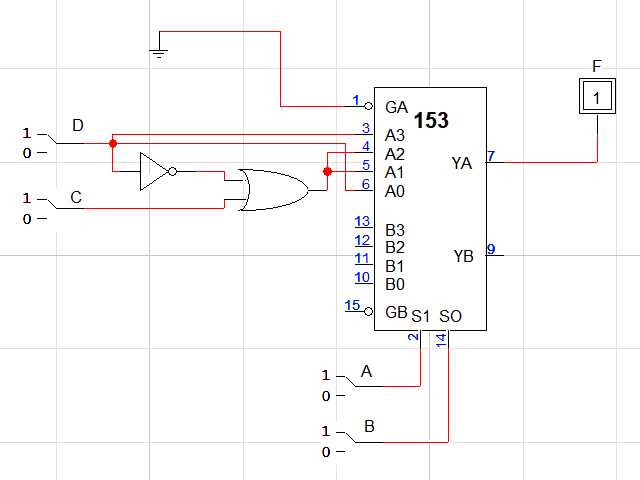


**Anexo 1.3**

**Circuitos**

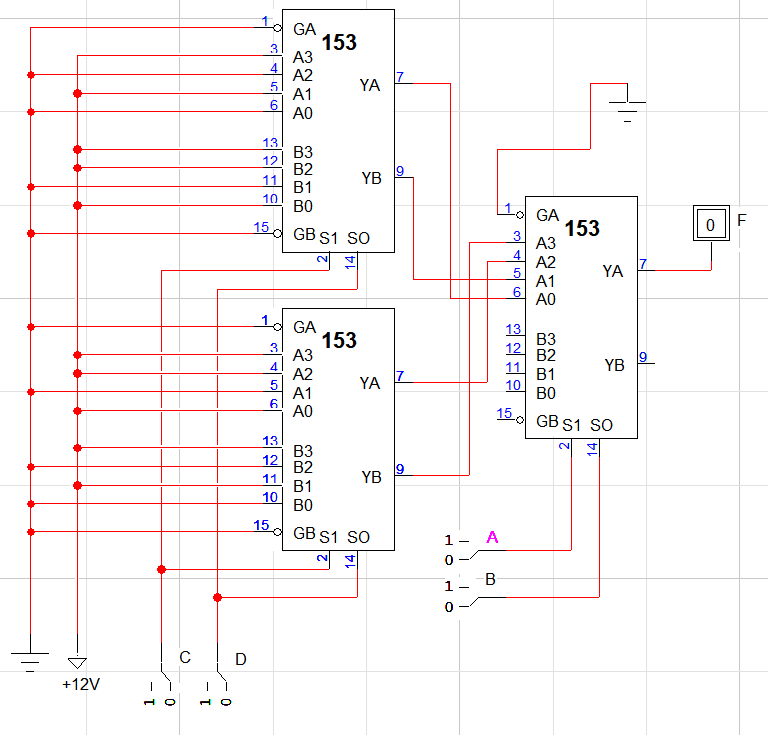
**Anexo 1.3.1**

**Un multiplexor 4:1**



**Anexo 1.3.2**

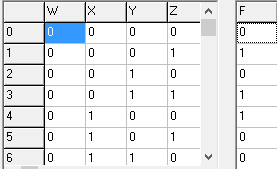
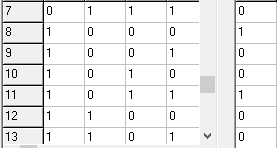
**Árbol de multiplexores 4:1**



**Anexo 2**

**Anexo 2.1**

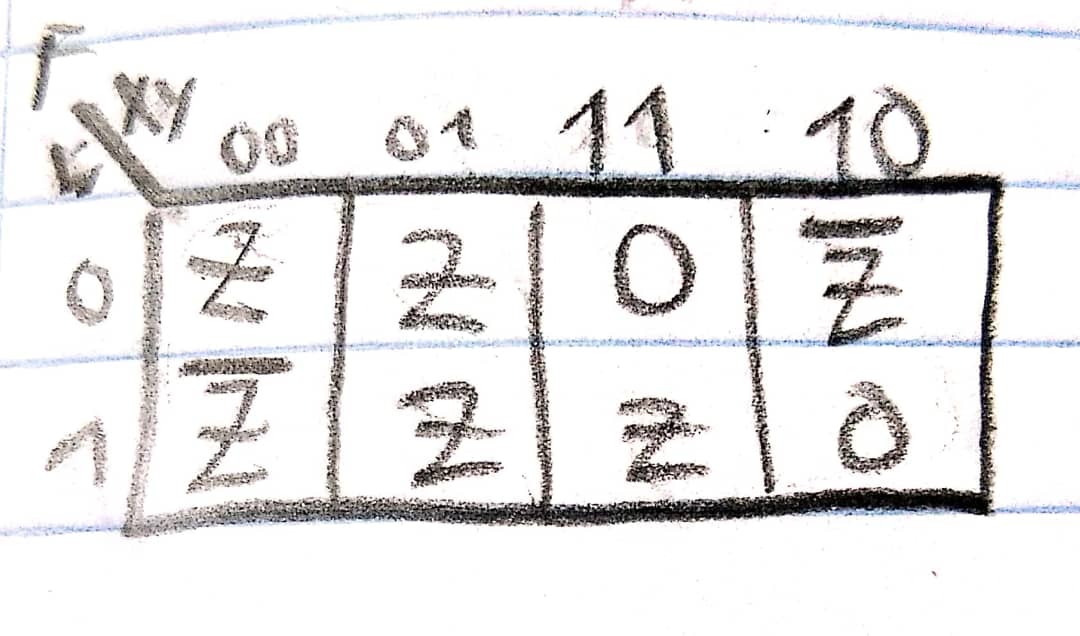
**Tabla de la verdad**





**Anexo 2.2**

**Mapas de Karnaugh**

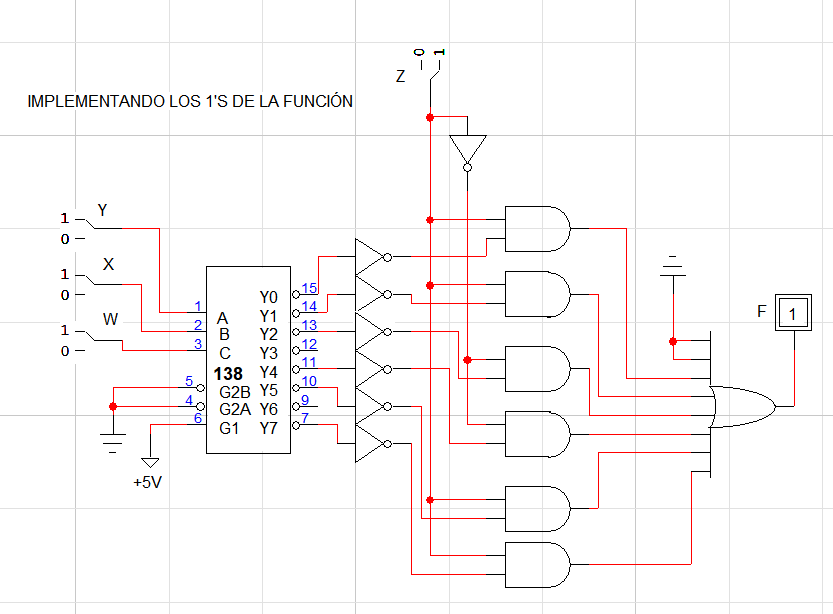
****

**Anexo 2.3**

**Circuitos**

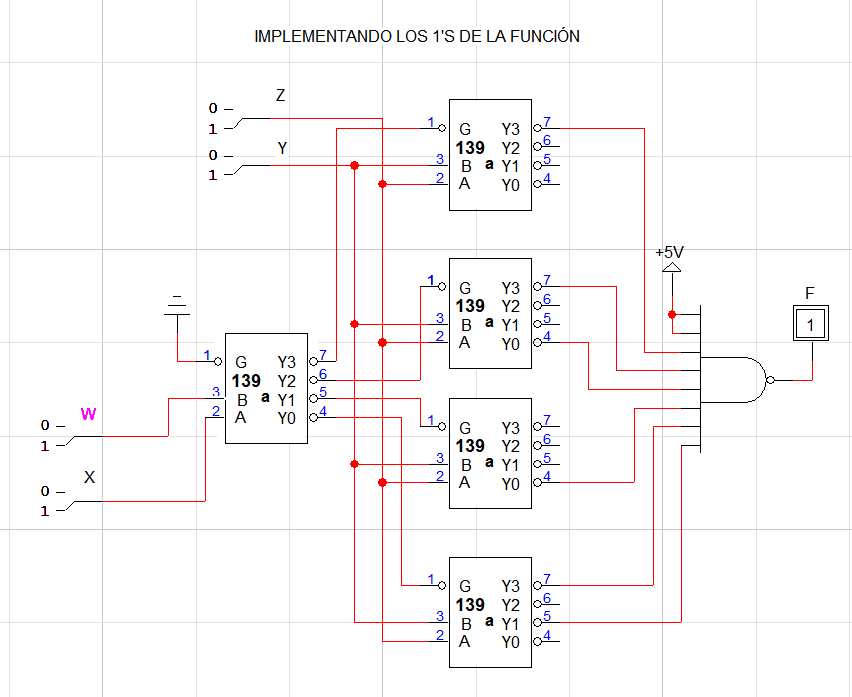
**Anexo 2.3.1**

**Implementación con DEMUX 1:8**



**Anexo 2.3.2**

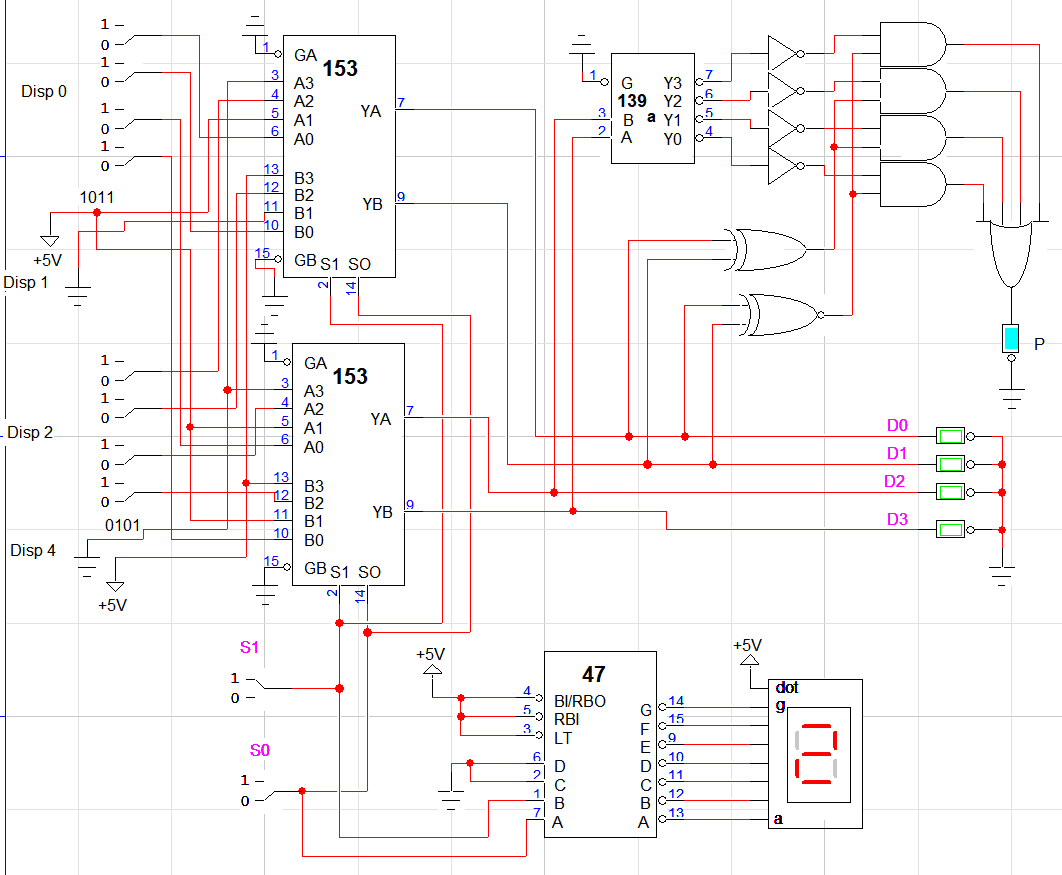
**Implementación con árbol de DEMUX 1:4**



**Anexo 3**

**Anexo 3.1**

**Simulación en LogicWorks**



**Anexo 3.2**

**Simulación virtual del circuito**

