



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



**TECNOLOGICO NACIONAL DE MEXICO**

**INSTITUTO TECNOLÓGICO DE MORELIA**

**INGENIERIA EN SISTEMAS COMPUTACIONALES**

**ARQUITECTURA DE COMPUTADORAS**

**PROFESOR: SERVANDO GONZALEZ HERNANDEZ**

## **Practica 2**

### **Simulación desplazador**

#### **INTEGRANTES**

**URBINA HERNANDEZ RICARDO**

**GONZALEZ LARA SANTIAGO**

**REALIZACION:24/10/24**

**ENTREGA:06/12/2024**

## Material

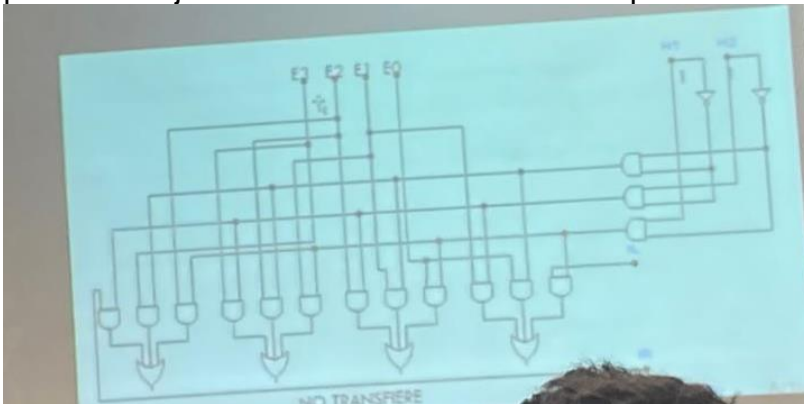
- Simulador proteus
- Compuertas lógicas NOT
- Compuertas lógicas AND
- Compuertas lógicas OR
- LOGIC-STATES proteus(entradas/señales)
- FLIP-FLOPS TIPO D

## Objetivo

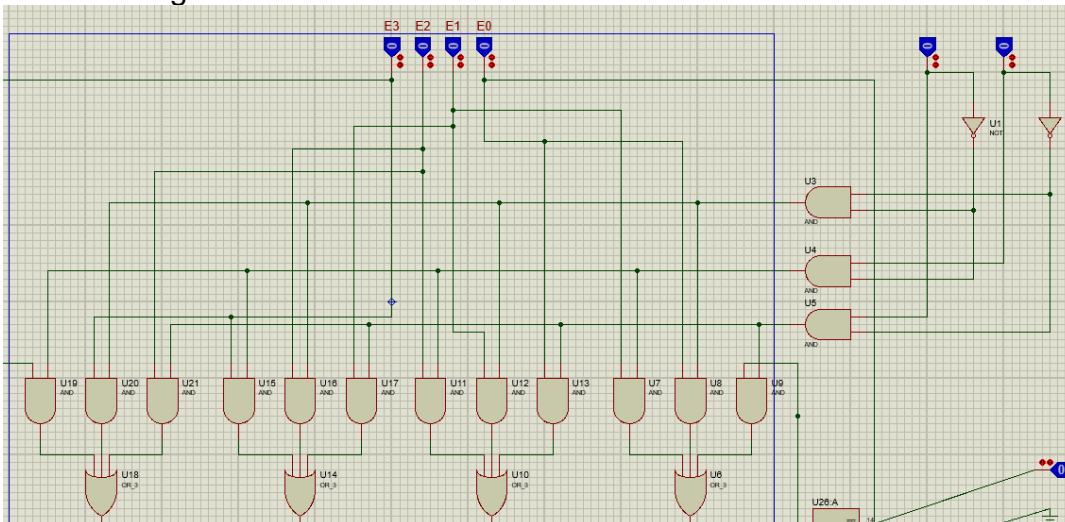
Diseñar y simular un desplazador de 4 bits, incluyendo su lógica de control para implementar diferentes desplazamientos y rotaciones.

## Desarrollo

Trasladamos el siguiente circuito al proteus, que es el de un desplazador, para poder manejarlo e identificar fácilmente sus partes.



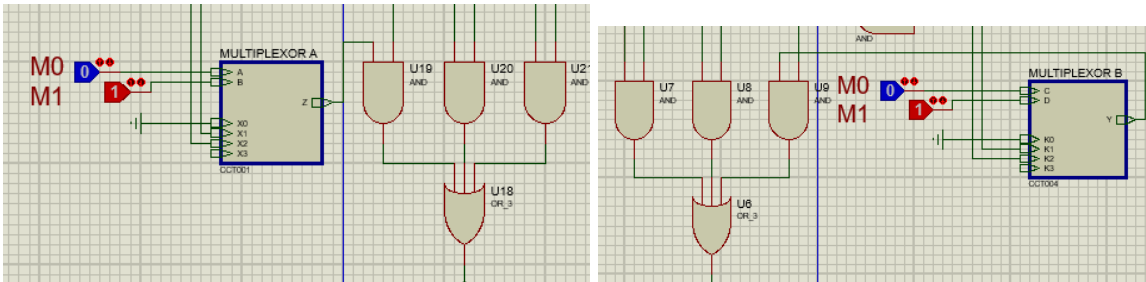
Siendo el siguiente resultado



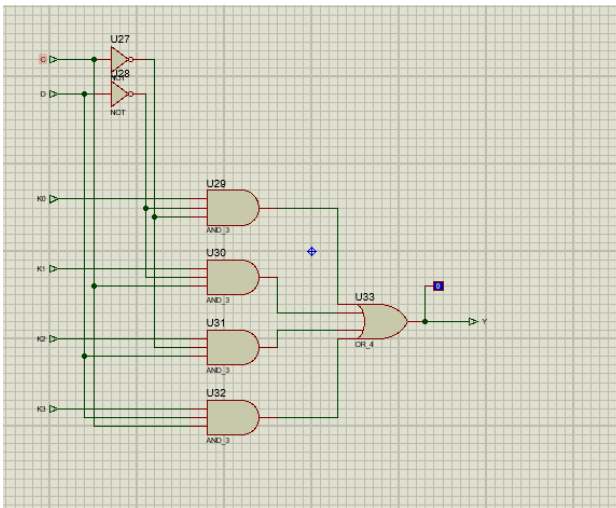
Donde esencialmente distinguimos 3 partes, sus entradas E0, E1, E2 Y E3, un multiplexor con sus bits de selección y lo más importante, el desplazador. Los bits de selección del multiplexor nos ayudan a saber qué operación podemos hacer

H1	H0	ACCION
0	0	Transfiere
0	1	zona izquierda
1	0	zona derecha
0	0	**

Como tal, en este punto la zona derecha y la zona izquierda lo que hacen es hacer un desplazamiento a la derecha e izquierda respectivamente, pero de un modo 'sencillo', y como tal al final de esta practica mas que servirnos como desplazamientos nos servirán para seleccionar más específicamente que desplazamiento (lógico, aritmético o rotación) queremos hacer, ¿Cómo? Se podría decir que al ingresar el h1 y h0 necesario para entrar a cada rotación, lo que hacemos es indicarle hacia que lado de manda la señal(Hacia que compuerta AND de cada triada la manda, si a la primera, segunda o tercera) , que en conjunto con unos DMUX (uno en cada lado) nos ayudarán a seleccionar específicamente que rotación hacer.



Ambos con el mismo interior



2 multiplexores que con sus bits de selección M1 y M0 que, para fines prácticos y de simplicidad, son lo mismo. Pues como ya explicamos anteriormente, con los bits de selección del primer multiplexor seleccionamos hacia qué lado salen las entradas que necesitan ser desplazadas, por lo que, si entra hacia un el lado derecho, el izquierdo queda 'inactivo' y viceversa, así que es indistinto si comparten bits de selección.

Resultándonos pues en la siguiente tabla de verdad:

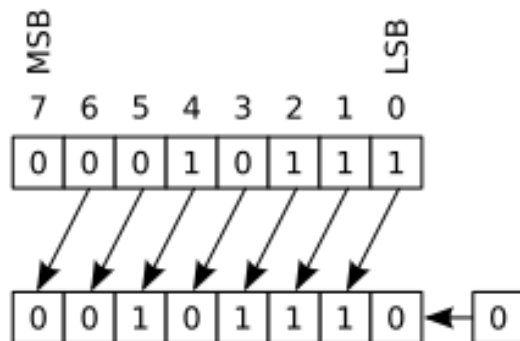
M1	M0	resultado
0	0	LSL – LSR
0	1	ASL – ASR
1	0	ROL - ROR
1	1	IL SERIE – IR SERIE

Ósea, una tabla general como la siguiente

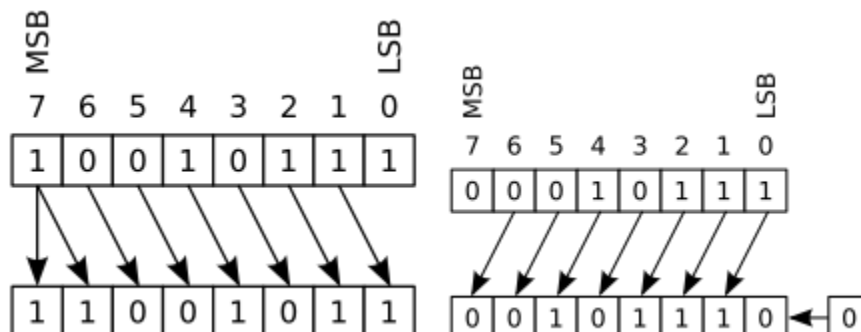
H1	H0	M1	M0	resultado
0	0	*	*	transfiere
1	0	0	0	LSL
		0	1	ASL
		1	0	ROL
		1	1	IL Serie
0	1	0	0	LSR
		0	1	ASR
		1	0	ROR
		1	1	IR Serie

Cada uno lleva a un tipo de desplazamiento

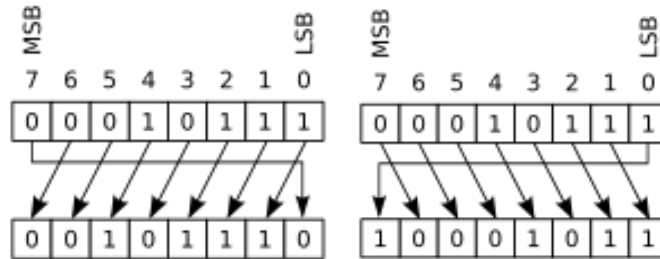
- Desplazamientos lógicos (Logic Shift Right & Logic Shift Left): los bits de un registro son desplazados hacia la dirección que indicada, si hay un bit activo en el borde de la dirección que estemos desplazando, se pierde, y entra un 0 en el registro opuesto a la dirección que se desplazó, como en el siguiente ejemplo. Se suele utilizar para operaciones como multiplicaciones de enteros sin signo, pues al perder el bit de la derecha pierde el signo, o a divisiones donde no nos importe la parte decimal del resultado, pues al perderse el bit menos significativo se pierde la parte decimal del cociente de la division



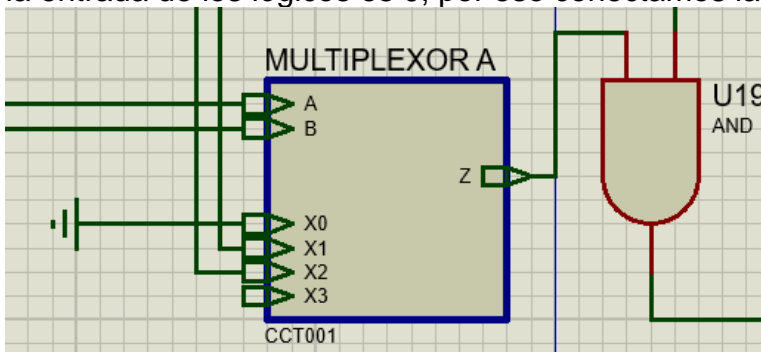
- Desplazamientos aritméticos (Aritmetic shift Left & Aritmetic shift Right): son operaciones que mueven los bits de un número binario hacia la izquierda o derecha, preservando el signo del número, ¿Cómo lo hace? En el caso de la derecha copia el bit más significativo en los espacios vacíos, preservando el signo. En el caso de a la izquierda, mueve los bits hacia la izquierda y llena el espacio vacío con un 0.



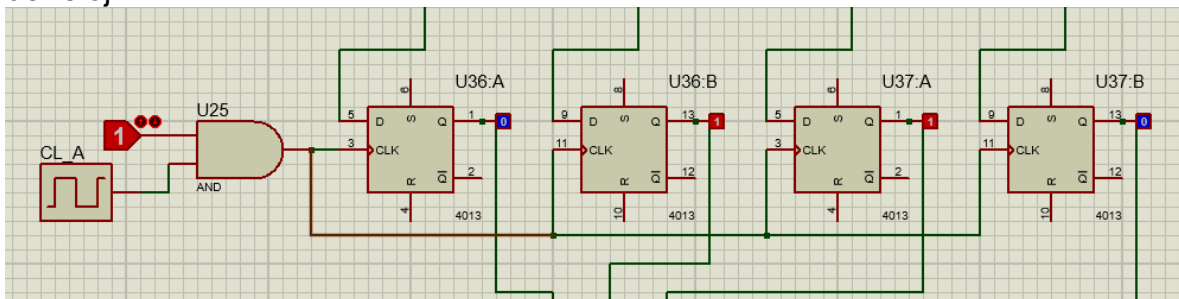
- rotaciones: los bits de un registro se mueven de manera circular, como conectando el bit mas y menos significativo. El bit que sale por el extremo por el de la dirección en la que fue rotado, es copiado en la del bit opuesto a donde se está rotando.



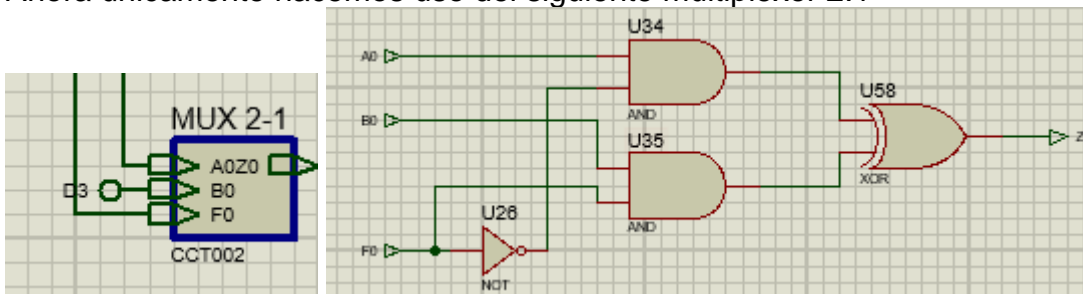
Sabido esto, únicamente conectamos los bits más y menos significativos respectivamente al multiplexor y conectamos su salida al desplazador (por defecto la entrada de los lógicos es 0, por eso conectamos la tierra)



Ahora lo que necesitamos es tener donde guardar las salidas y de donde jalar las entradas, por lo que usaremos flipflops tipo D, metiendo su pulso de reloj y una compuerta and con un logicstate, para controlar cuando usar y no usar este pulso de reloj



Ahora únicamente hacemos uso del siguiente multiplexor 2.1



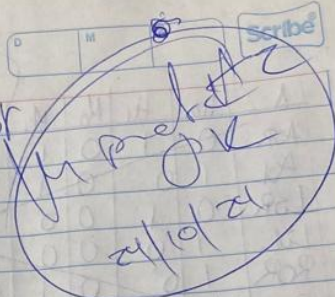
Que con un bit de selección (f0), la salida del desplazamiento (b0) y la entrada inicial (A0Z0), logramos hacer que haga los desplazamientos de manera ciclica



## Bitácora

### Practica 2 - Desplazador

Gratzalez Lara Santiago  
Hernandez Urbina Ricardo



Objetivo  
Diseñar y simular un desplazador de 4 bits, incluyendo la lógica de control para implementar los diferentes desplazamientos y rotaciones.

H <sub>i</sub>	H <sub>o</sub>	N <sub>i</sub>	M <sub>o</sub>	
00	* *			transfiere
10	00			LSL
	01			ASL
	10			Rol
	11			IR serie
01	00			LSP
	01			ASR
	10			ROR
	11			IR Serie