

1. Circuito Secuencial para el Almacenamiento y Lectura de Bytes en Registros en Paralelo FF tipo D (Active Low)

Andreu Lechuga, 202073595-6, andreu.lechuga@usm.cl

1. Resumen

Se desarrollo un circuito secuencial en Logisim capaz de almacenar un conjunto de entradas en binario en forma de registro y luego escoger una de ellas para mostrarla por consola como una secuencia de luces LED. La entrada estuvo compuesta de 4 números binarios de 8 bits cada uno, tres para almacenar y uno para seleccionar.

Se implementaron Flip Flops de tipo D sobre FF SR (Active Low) para almacenamiento y lectura en paralelo, mientras que para la estructura se usaron multi y demultiplexores, splitters y un reloj.

Cada parte del circuito quedo plenamente funcional y a lo largo de este informe se detallará el proceso de concepción y desarrollo del sistema entero, así como la lógica detrás de cada circuito utilizado.

2. Introducción

En este experimento se planteó, conceptualizo y desarrollo un circuito complejo con la capacidad de recibir tres números binarios de 8 bits y almacenarlos en tres registros diferentes, implementados con FF de tipo D (Active Low). Además, el registro a utilizar para cada dato es decidido por un reloj que coordina todo el sistema.

El registro a mostrar es decidido a partir de un cuarto input, numero binario de 8 bits, que identifica el registro como la representación binaria de su valor, del 00 (1) hasta el 10 (3), tratando el resto de los bits como DONT CARE.

Una vez que se almacenaron los 3 registros, el circuito recibe un 4to input en forma de selector, conectado simultáneamente al demultiplexor de entrada y al multiplexor de salida. Esto permite la coordinación del dato almacenado para posteriormente mostrarlo en su formato LED.

Para la estructura del circuito se implementaron:

- 4 entradas de 8 [bit]
- 1 entrada de 3 [bit] (manual)
- 2 Demultiplexor
- 1 Multiplexor
- 1 Reloj

- Múltiples Splitters
- Cable

Un circuito eléctrico es un conjunto de elementos que interactúan y dictan un determinado movimiento para una corriente eléctrica, normalmente con algún propósito útil. En palabras de Gonzalo Ramos (2014), “El circuito eléctrico es el recorrido establecido de antemano que una corriente eléctrica tendrá.”

Dada la naturaleza de la electrónica y los computadores, se utiliza un acercamiento binario matemático conocido como Algebra de Boole con el fin de interactuar con la máquina.

Una compuerta lógica es una representación electrónica y material de un concepto propio de dicha algebra.

Las compuertas lógicas son utilizadas para imponer un comportamiento determinado de la corriente, puesto que “garantizan el flujo y control de los electrones que conforman la electricidad” (Gonzalo Ramos, 2014)

El FF tipo D es un tipo de Flip Flop para almacenamiento, desplazamiento y lectura de 1 bit de información. Su implementación en paralelo o secuencia permite incrementar ese margen, mientras que si se disponen varias secuencias en forma de matriz comienza a considerarse Memoria.

El FF tipo D es una extensión del Flip Flop SR, que a su vez esta construido en base a 2 compuertas NAND. Además, las compuertas FF SR utilizadas en este experimento son de tipo Active Low. Esto significa que, en un estado neutro, los inputs se encuentran siempre activos (1) y que una baja momentánea de la señal (0) provoca el cambio de estado.

3. Desarrollo

Antes de comenzar a tirar cables y quemar fusibles, fue necesario conceptualizar el problema y entender sus exigencias.

Lo primero fue entender la lógica detrás de un registro y decidir, por ejemplo, si convenia implementarlo en paralelo



o en serie. La respuesta fue en paralelo, puesto que el input viene como un bloque (Byte) y no como una corriente de bits. Lo siguiente fue hondar en los Flip Flops, como y porque funciona cada uno y cuáles son sus diferencias. Preámbulos importantes para decidir cual utilizar en función de la naturaleza del problema.

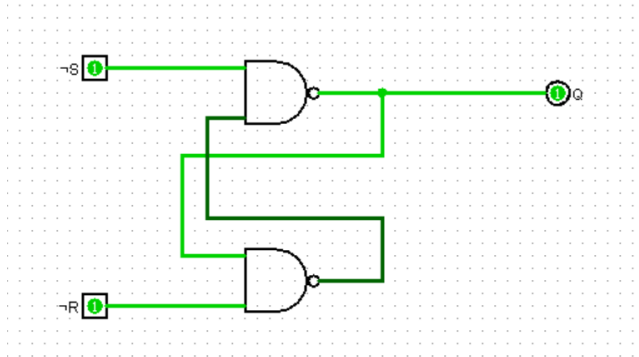


Figura 1: FF SR en estado neutro, almacenando 1

Resultado que por temas de la plataforma utilizada (Logisim) y las propias restricciones del departamento, no fue posible implementar FF de tipo JK ni T. Esto debido a que según la misma documentación del programa (soportada por la comunidad), no es posible construir uno de estos FF desde cables y compuertas, por alguna razón. Ellos recomendaban usar los ya implementados por el programa. Sin embargo, la regulación del departamento prohibía su uso, lo que me llevo a un callejón sin salida. Por otro lado, resultado que el FF que mejor sintonía tuvo con el problema fue el tipo D, siendo capaz de implementar el FF SR (Active Low) sin problemas. El FF SR (Active Low) se conforma de 2 compuertas NAND interconectadas entre sí, además las entradas S (set) y R (reset) más la salida Q, como se muestra en la Figura 1.

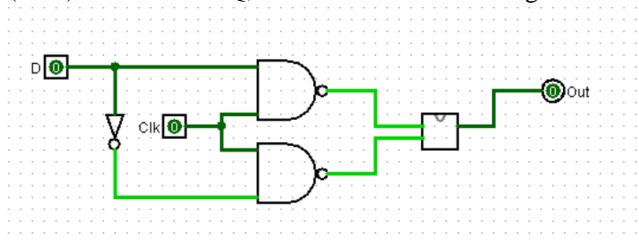


Figura 2: FF tipo D

Una vez realizados los FF tipo D, como se muestran en la Figura 2, se dispusieron en forma de registros para números binarios de 8 bits y coordinados por un reloj (Clk). A este circuito se le llamo Registro.

El circuito Registro permite una entrada de 8 bits que procede a separar en bits individuales, como se muestra en la Figura 3. De esta manera, a la subida del reloj, se cargan en paralelo a una secuencia de 8 FF tipo D. De manera similar, la salida del registro almacenado se vuelve a unir en un Byte y queda disponible para su acceso desde el Multiplexor del circuito principal.

Una vez listos los registros, surgió la necesidad de algún tipo de condicional o control de flujo que hiciera entender al circuito principal que registro correspondía a cada número y que numero mostrar al momento de seleccionar. Para la solución de este problema se utilizaron dos tipos de plexores, Multi y Demultiplexores, para la captura y transmisión de la instrucción correcta respectivamente. Además, se utilizó un Demultiplexor extra para la coordinación el sistema en torno del reloj.

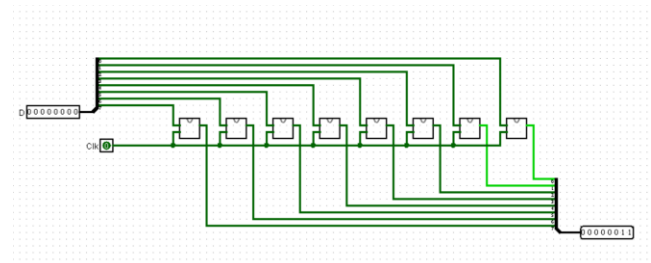


Figura 3: Circuito Registro

La lógica fue permitir que el Demultiplexor de datos recibiera 1 señal de 1 [byte] a la vez. Simultáneamente un registro por la entrada de Selección, por la que ingresa un 3er input manual de 3[bit] en forma de instrucción. De esta forma, el Demultiplexor direccionara recibirá la nueva entrada y la transmitirá por el canal (registro) que corresponda, dado el input de Selección.

El problema con la implementación anterior (One-Hot) fue que, de esa manera, los canales que no estuviesen activos o seleccionados transmitirían un 0 a los registros. Así, al momento de la subida de reloj, todos los registros menos el seleccionado actualizarían a un valor nulo (00000000).

La solución a esto fue implementar un segundo Demultiplexor encargado de coordinar el reloj con el sistema. El segundo Demultiplexor (de reloj) estaría conectado al mismo selector que el primero (de datos), permitiendo la coordinación de canales para la subida de reloj. En otras palabras, al momento de actualizar los valores, el reloj solo



interactuara con el canal seleccionado en ese momento, actualizando solo ese registro sin poder alcanzar el resto.

De esta manera el circuito principal quedo capaz de recibir varios inputs y registrarlos exitosamente por separado. Solamente queda por implementar la lectura e interpretación del 4 input, encargado de seleccionar la salida por consola, y la salida en sí, con el Multiplexor y la secuencia LED.

El Multiplexor de datos recibe 3 canales de información, con solo uno activo a la vez, y transmitirá el canal que corresponda, dado por el input de Selección. Por otro lado, su entrada de Selección corresponde a un numero de 2 bits, extraído del input original de 8 bits. Para esto se utilizaron 2 splitters, el primero para separar la entrada de 1 Byte en 8 bits individuales y la segunda para leer solamente los 2 primeros bits y concatenarlos. El resto de los bits se tratan como DONT CARE, normalmente debiendo ser 0s. Los bits concatenados serán la entrada de Selección para el Multiplexor de datos.

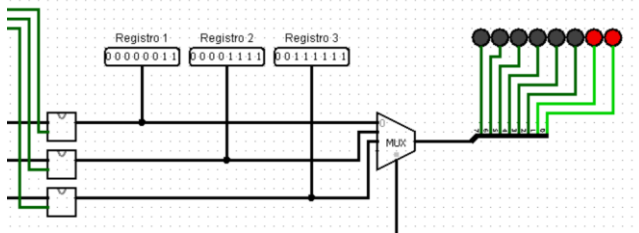


Figura 4: Salida LED del Registro 1

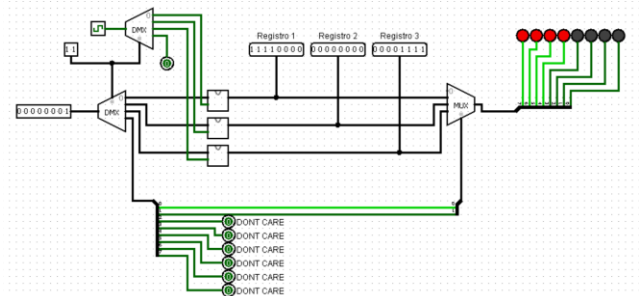
Por ultimo se interpreto la salida de 8 bits en forma de secuencia de luces LED, como se muestra en la Figura 4, dando por terminada la creación del circuito.

4. Resultados

En el siguiente apartado se presentarán distintas entradas que demuestren el correcto funcionamiento del programa, dando paso a un análisis más en detalle en la sección siguiente. Para la demostración me apoyare en los resultados ya verificados dados por el Departamento.

Ejemplo 1

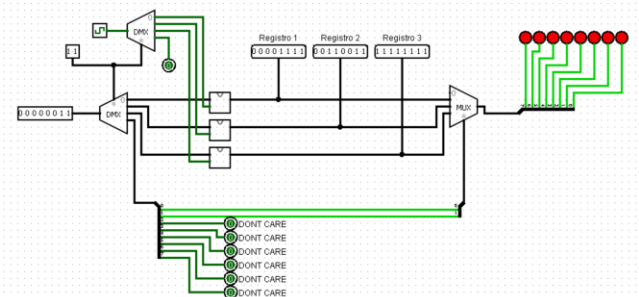
Bin1 – Registro 1: 11110000
Bin2 – Registro 2: 00000000
Bin3 – Registro 3: 00001111
Bin4 – Selección: 00000001



Ejemplo 1: Salida Registro 1

Ejemplo 2

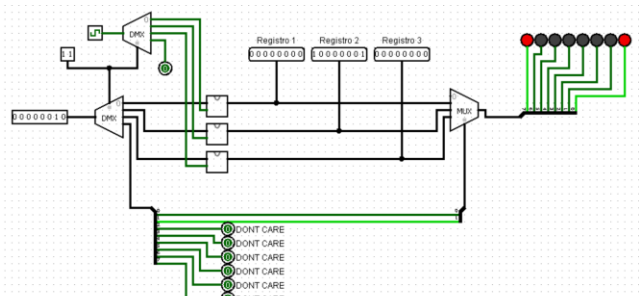
Bin1 – Registro 1: 00001111
Bin2 – Registro 2: 00110011
Bin3 – Registro 3: 11111111
Bin4 – Selección: 00000011



Ejemplo 2: Salida Registro 3

Ejemplo 3

Bin1 – Registro 1: 00000000
Bin2 – Registro 2: 10000001
Bin3 – Registro 3: 00000000
Bin4 – Selección: 00000010



Ejemplo 3: Salida Registro 2

5. Análisis



Siendo honestos, el proceso de realización del proyecto entero, desde los primeros acercamientos hasta el final, fue bastante directo. Diría que la comprensión de que hacer y como hacerlo fue exponencial. Caótico al comienzo, pero una vez entiendes la lógica de cada cosa el trabajo se vuelve bastante lineal. Aclaro esto porque no me encontré con grandes dificultades a excepción de la comprensión teórica primera. Que es un Flip Flop y como se implementa. Como funciona un registro. Tipos de registros. El funcionamiento de las compuertas estaba claro del material visto en clase. Luego de pasar esa barrera, faltó conectar los cables y funciono en su gran mayoría.

Una gran piedra en el camino fue idear una manera en que un único reloj pudiese coordinar todo el sistema sin interferir con demás registros. Finalmente se logró, con la implementación del Demultiplexor para el reloj, como explico en detalle en la sección de Desarrollo.

6. Conclusión

Con el proyecto realizado se comprendieron en profundidad conceptos clave de los pilares de la electrónica, tales como la estructura interna de un circuito, uso de reloj para circuitos secuenciales síncronos y la punta del iceberg sobre la Memoria de un ordenador. En sus bloques más fundamentales, se profundizo en registros a través de Flip Flops, su estructura interna y su funcionamiento, ventajas comparativas (con respecto a otros FF) y más, dentro del ámbito de la Memoria. Al respecto de la estructura interna, se hondo en los usos e interacciones entre compuertas lógicas y manipulación de entradas numéricas, como bloques y por componentes, profundizando en la estrecha relación que guarda la computación con los flujos electrónicos y trascendiendo las barreras de la abstracción a un conocimiento más integrado.

Mediante la superación, más teórica que practica, de distintos imprevistos, mencionados en el apartado de Análisis, se completó con éxito el circuito que se buscaba realizar.

7. Anexos

8. Bibliografía

Gonzalo Ramos. (2014). Definición MX. Recuperado de <https://definicion.mx/circuito-electrico/>