Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Eléctrica

ALU y Shift Register

Laboratorio #3

Integrantes: Ariel Núnez.
Profesor: Ricardo Finger.
Auxiliar: Sergio Saavedra.
Ayudantes de Laboratorio: Alonso Caviedes.
Daniel Vásquez.
Diego Gonzalez.
Néstor Henríquez..

Fecha de entrega: 10 de junio de 2020
Santiago, Chile

1. Introducción

Para esta experiencia de laboratorio se realizará la construcción de una Arithmetic Logic Unit, o ALU, de 8 bits con cuatro operaciones fundamentales a partir de dos entradas A y B: Sumar, comparar entre bits mediante AND y OR, y finalmente el copiar la entrada A; con estas operaciones, a partir de una señal de control se elige cual se presentará en la salida de la ALU. Una vez hecho este proceso, la salida de la ALU se conecta a un Working Register, o WREG, el cual permite tener un registro de las operaciones realizadas en la ALU y poder desplazar los bits de este resultado hacia la derecha en caso de quererse esto.

A partir de este diseño y aplicación del componente, se busca comprender el funcionamiento de la ALU a partir de funciones independientes, las cuales pueden ser escogidas a partir de una entrada de control. Junto a esto, ver como la ALU puede almacenar información en un registro para posteriormente dejarla en una memoria, manipularla mediante el WREG o incorporarla en la entrada de la ALU para poder seguir haciendo sus operaciones a gran velocidad.

Con lo anterior en consideración, se podrá entender de una forma práctica como la ALU de dispositivos más complejos realiza cientos de operaciones en poco tiempo, al igual que retroalimentarse si esta lo desea a partir del registro. De este modo, se puede complementar con componentes ya vistos como una memoria ROM o RAM, almacenando información y presentándola a partir de la cantidad de bits que se tiene. Así, se entenderá el proceso de información y operaciones que tiene, por ejemplo, un computador en su interior.

2. Metodología

En el desarrollo de esta experiencia se utilizó el software SimulID, versión 0.3.12. Este software permite la simulación de componentes lógicos, entre los cuales utilizaremos los componentes fundamentales (AND, OR, NOT, XOR) junto a componentes como el Flip Flop D, Sumador, Multiplexor, BUS y LEDs en barra. Para alimentar estos componentes se utilizaron tanto fuentes de voltaje fijo como una fuente de tipo CLOCK. Estos componentes se utilizaron para la fabricación de una ALU y WREG (Working Register) de 8 bits.

Cabe destacar el uso del BUS en esta experiencia, no usado en las anteriores. El BUS nos permite conectar un cierto número de bits a una salida de un cable, para luego conectar esta salida a otro BUS que va a entregar los bits originales en la posición de la entrada original. Esencialmente hace el rol de un circuito integrado que nos permite simplificar y ordenar el esquema del circuito, lo cual es bastante necesario en una ALU de 8 bits que posee múltiples funciones y se deben conectar muchas cosas, lo cual facilitará el proceso.

2.1. ALU

Para la construcción de la ALU, se diseñó un circuito que cumpliese cuatro funciones distintas: sumador, operación AND en cada bit, operación OR en cada bit y copiar directamente la entrada. Para seleccionar la operación a utilizar se hace uso de Multiplexores cuyas entradas de control están conectadas en paralelo al "ControlBUS", fuentes de voltaje cuya combinación determina la operación que que resultará a partir de la ALU.

La salida de la ALU se presentará mediante un display de LEDs, y está se va a conectar con un "Working Register", o WREG, el cual permite registrar el dato y modificarlo mediante la operación de desplazamiento de los bits a la derecha.

Para la implementación del sumador se utilizaron los bloques Sumadores predeterminados de SimulID, conectando la salida "Carry Out" del primer sumador al "Carry In" de la siguiente. La salida "Carry Out" del Sumador del Most Significant Bit (MSB) se utilizará para realizar un sistema de prevención de errores por Overflow mediante saturación.

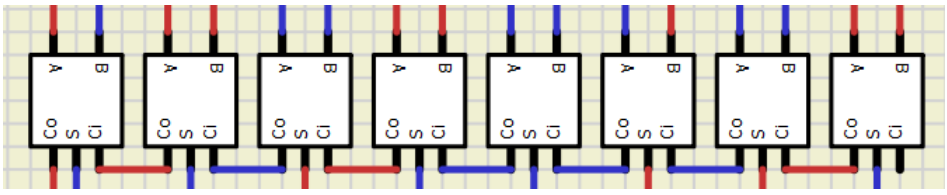

Figura 1: Sumador de 8 bit, con las conexiones requeridas para su funcionamiento en conjunto.

La implementación de la función AND y OR resultan más sencillas, ya que solo debemos comparar entre cada bit utilizando el operador respectivo. Así, la estructura de las funciones AND y

OR resultan iguales, solo modificando el operador. La función de copia es simplemente enviar la señal directamente a los multiplexores.

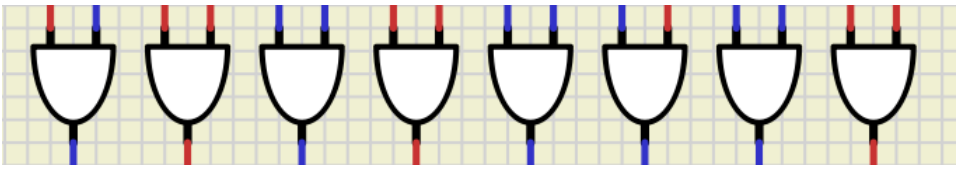

Figura 2: Operador AND entre bits. Para el operador OR, tan solo debe reemplazarse el componente.

Para hacer la elección de la operación a utilizar en la ALU, se conecta la salida de cada bit respectivo de su operación a un multiplexor, con los cuales decidiremos que entregará la ALU mediante el "ControlBUS"; con esto, se utilizarán ocho multiplexores, uno por cada bit, utilizando cuatro de sus entradas de información y solo dos entradas de control. Se podrían usar las tres, pero así se reduce el uso de recursos en fuentes de voltaje. De ser necesario se puede utilizar la tercera entrada de control si se integran más funciones en la ALU, como resta o complementos.

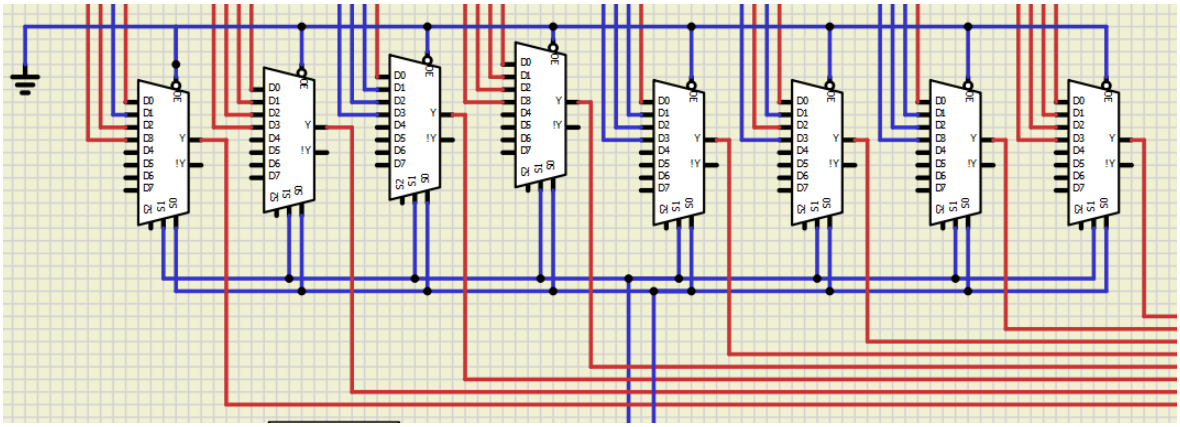

Figura 3: Multiplexores usados en la ALU para definir la operación a utilizar. Atención a las entradas de control conectadas en paralelo para un funcionamiento síncrono.

Con esto, podemos conectar la salida de los multiplexores a un display de LEDs en barra, para poder ver el resultado, y se conectará este resultado al "Working Register" (WREG) para posteriormente manipular la salida de la ALU.

2.2. WREG

El proposito del WREG es ser un registro de los resultados de la ALU, los cuales pueden posteriormente integrarse a una memoria (ROM o RAM), o manipularse mediante una operación "Shift", lo cual hace que los valores de bits resultantes de la ALU se desplacen un bit hacia la derecha, haciendo que el MSB sea 0. Esta operación se realiza al encender una señal "Shift", la cual

si se mantiene encendida hace que el desplazamiento se reitere constantemente, hasta que la señal se apague, donde nuevamente se registrará el resultado actual de la ALU.

Para la incorporación de la operación de desplazamiento se utilizó una modificación del circuito de Shift Register modo PISO visto en cátedra, haciendo uso de operadores AND y OR, mientras que el circuito en cátedra solamente usa componentes NAND. De solo tener componentes NAND o de ser más baratos en una situación de implementación real, se decidiría en el más adecuado para las necesidades de la fabricación de la ALU. El circuito además de los operadores lógicos requiere los componentes Flip Flop D, y una señal de reloj la cual todos los Flip Flop comparten para que se comporten de manera sincronizada.

El diseño de este circuito, conocido como Shift Register en modo PISO, consiste en tomar una señal que empiece el desplazamiento a la derecha de la señal registrada y a partir de esta, enviar la señal del MSB al siguiente bit, y asi sucesivamente. La forma en que ocurre este proceso es mediante la lógica de los operadores, en el cual solo se debería dar una señal alta de entrada para el Flip Flop si es que el bit registrado del Flip Flop anterior es alto.

Un detalle a considerar, es que en el esquema original, si el MSB era un 1, se iba a propagar un 1 al desplazar a la derecha, y caso análogo con un 0, propagando el 0 en el resto de los bits. Para solucionar este problema, simplemente se agregó un nuevo elemento a la cadena (esencialmente es un Shift Register modo PISO de 9 bits) en el cual el MSB siempre es 0 conectando a tierra el Flip Flop asociado, y considerando el resto para el display del WREG.

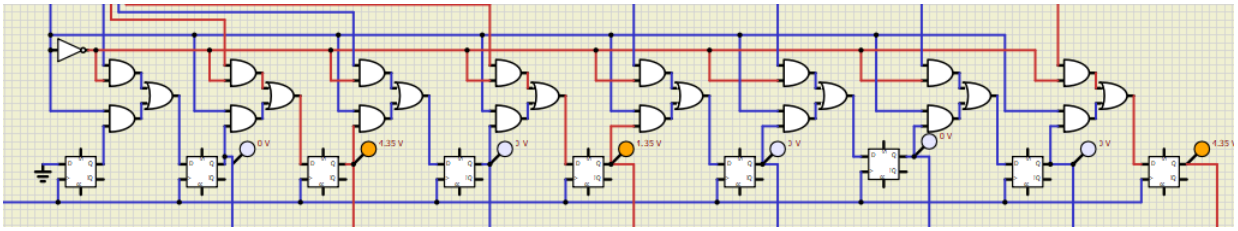

Figura 4: WREG del circuito, utilizando un Shift Register modo PISO. Notese como la primera entrada del Flip Flop esta conectada a tierra y el resto asociada a los bits provenientes de la ALU y que saldrán al display.

Finalmente, este debe tener un display de LEDs para poder ver los resultados del registro. Para esto se conectan las salidas de cada Flip Flop en el orden deseado a una barra de LEDs como se ha hecho previamente. Cabe destacar que en el diseño de este circuito, se presentará un display de LEDs para la ALU y el WREG, para así comparar su comportamiento al mismo tiempo y como se sincroniza el registro mediante la señal “Shift”.

3. Análisis

3.1. ALU

Al implementar la ALU, se observó un correcto funcionamiento individual en cada componente. La forma en que se evidencia esto es a partir de la propiedad de SimulID de evidenciar salidas con señales altas o bajas mediante el color. Una señal alta presenta un cable de color rojo en el cual se envía, mientras que una señal baja presenta un cable de color azul. Con esto, podemos observar el comportamiento de las salidas sin necesitar de modificar el circuito de una forma significativa.

El sumador funcionó de forma correcta al vincular las entradas "Carry in" y "Carry out", logrando realizar las sumas en 8 bits de la forma esperada. Además, al detectar esta salida "Carry out" del último Flip Flop, el sumador correctamente realiza una corrección por saturación del overflow del circuito. Así, al ingresar la suma "11010001 + 01010101" el resultado es el número más cercano a 100000000 que podemos hacer en 8 bits, 11111111.

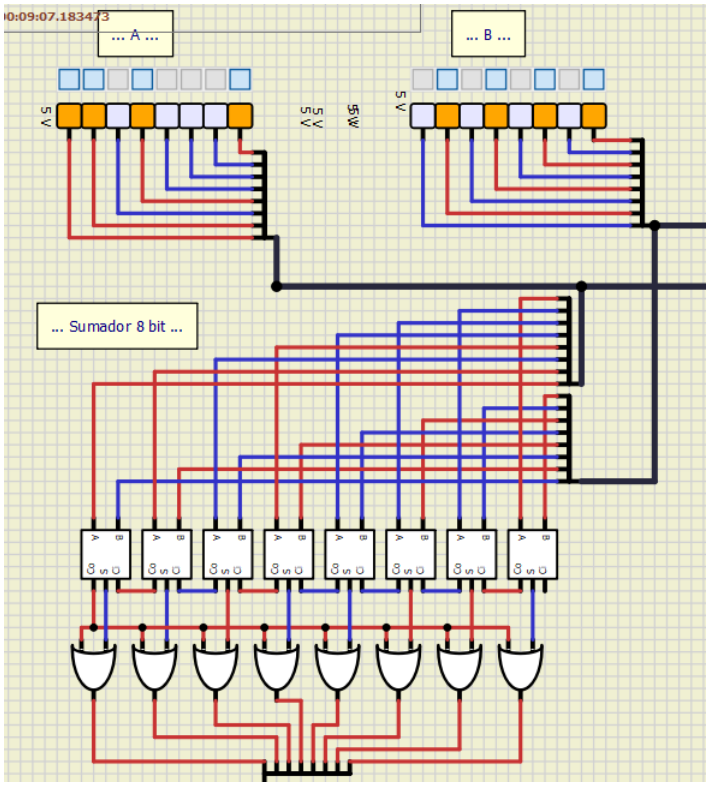

Figura 5: Resultado del sumador, utilizando un método de corrección de overflow por saturación.

El funcionamiento de los comparadores mediante la operación AND y OR resultó correcto y facil de comprobar. Notamos el comportamiento a partir de las animaciones de SimulID, el cual nos permite observar las señales altas y bajas que entran al operador lógico, y la salida de esto hacia el BUS del resultado. Observamos entonces como opera en AND y OR.

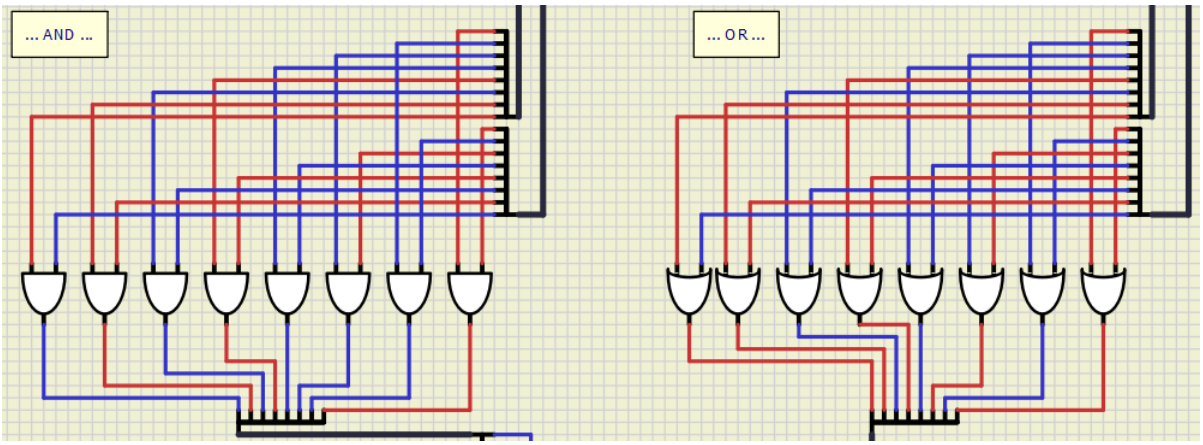

Figura 6: Resultado de los comparadores lógicos entre bits de AND y OR respectivamente.

Finalmente, para la operación restante solamente es necesario tomar la entrada directa desde la señal original y transmitirla mediante un BUS, para mantener orden, a la sección de multiplexores de la ALU. La sección de multiplexores funciona efectivamente también, ya que al proveer la señal con entradas que tendrían distintos resultados para las diversas operaciones de la ALU, al cambiar los valores del ControlBUS, la salida de LEDs correspondiente a la ALU presenta resultados distintos (y correctos según la operación asociada al ControlBUS) para las combinaciones de control conectadas a los multiplexores.

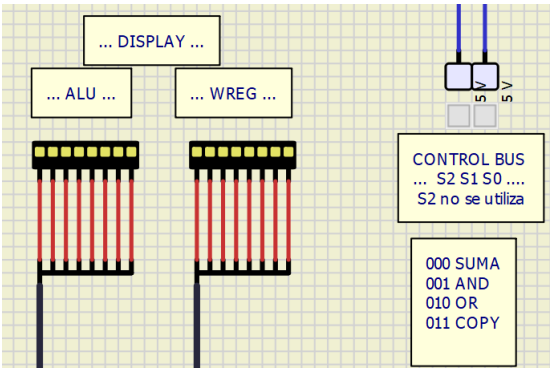

Figura 7: Resultado del sumador, visto en el display de LEDs, escogido a partir de la combinación 00 del ControlBUS.

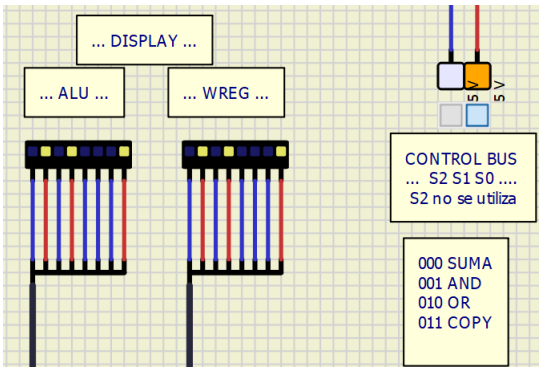

Figura 8: Resultado del comparador entre bits AND, visto en el display de LEDs, escogido a partir de la combinación 01 del ControlBUS.

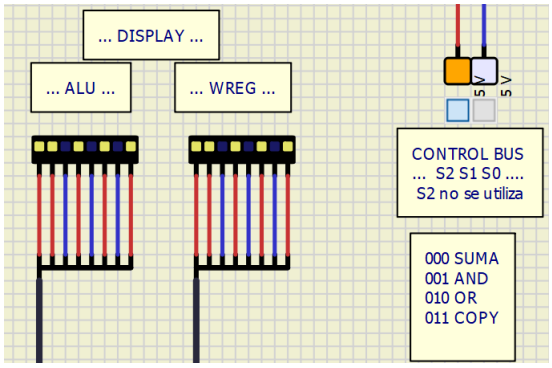

Figura 9: Resultado del comparador entre bits OR, visto en el display de LEDs, escogido a partir de la combinación 10 del ControlBUS.

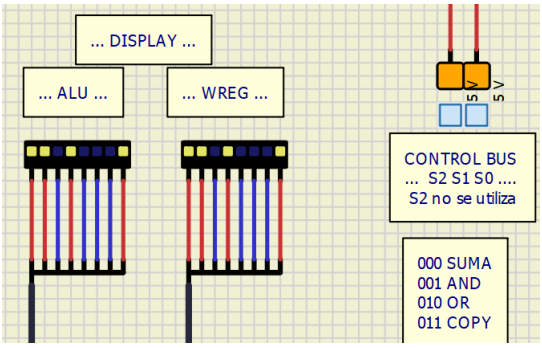

Figura 10: Resultado del copy a la entrada A, visto en el display de LEDs, escogido a partir de la combinación 11 del ControlBUS.

3.2. Working Register

Inicialmente, el Working Register sin integrar la señal de “Shift” debería funcionar de la misma forma que el display de la ALU. Corroboramos el correcto funcionamiento de este al comparar el

display de la ALU con el WREG, y ver la igualdad entre estos. Posterior a esto, al activar la señal "Shift", empieza el proceso de desplazamiento cuando se sincroniza el tick del reloj. Notamos como el primer flip flop siempre emitirá un 0 al siguiente, lo cuales el funcionamiento que esperamos. Así, si dejamos esta función seguir indefinidamente eventualmente iterará infinitamente una señal 00000000 en el display del WREG. Una vez se apaga la señal "Shift", el display presentará la señal en la salida de la ALU en ese momento.

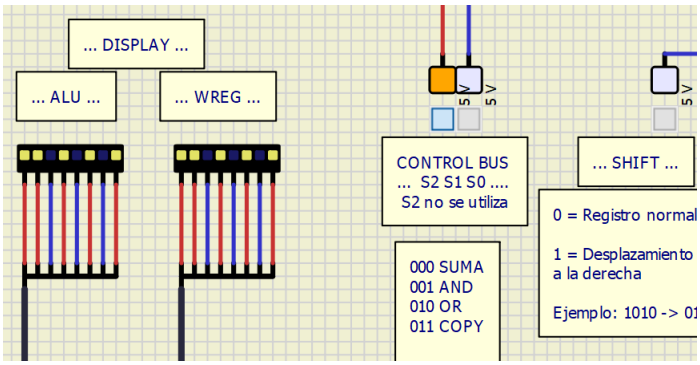

Figura 11: Resultado del sumador, utilizando un método de corrección de overflow por saturación.

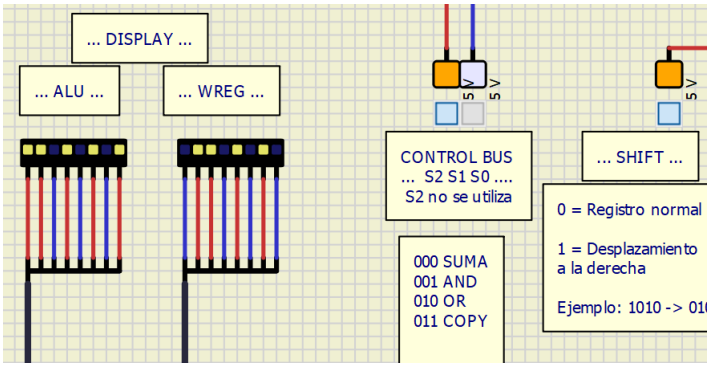

Figura 12: Resultado del sumador, utilizando un método de corrección de overflow por saturación.

Cabe destacar que en este circuito, al usar dos displays de LEDs con las condiciones por defecto se producen bajas de voltaje (de 5 V a 2.41 V) en ciertas secciones del circuito, en específico en las entradas de los bits del WREG. Por lo tanto, se modificaron los valores de detección de los operadores AND del WREG para que pudiese detectar estos voltajes más bajos, los cuales fueron de basados a partir de un componente real, el AND de dos compuertas SN54/74LS08, con un voltaje de detección alto de 2 V. Utilizando estas medidas basadas en componentes reales en la simulación, el circuito funciono como se esperaba, así que no requiere mayores cambios en otros lugares (la resistencia de los LED o prescindir del display de la ALU). Sin embargo, para la presentación correcta del circuito y el funcionamiento de los LED se modifico el valor de resistencia a 120 Ohms.

4. Conclusiones

A partir de los resultados obtenidos, la experiencia de laboratorio resultó exitosa, logrando correctamente el diseño de una ALU con cuatro operaciones al igual que un Working Register que posee la operación de un Shift Register, cuyo funcionamiento tanto para la ALU como el WREG resultaron correctos a partir de las simulaciones realizadas.

Con respecto a la ALU, notamos en su construcción como se conforma por distintas funciones que realizan su labor independientemente, y a partir de multiplexores podemos elegir la función indicada. Con esto, si quisiéramos podríamos incluir muchas más funciones en nuestra ALU, sin embargo el costo de estas podría ser más del necesario, o el tiempo en que demora realizar las operaciones.

Al mismo tiempo, con el WREG consideramos la importancia del MSB en la operación, en el cual este es el que conforma el valor del bit que se va a propagar al desplazar los bits del números hacia la derecha. Así, en la creación de un circuito secuencial se debe dar importancia a como los bits iniciales afectan a los siguientes en la cadena, en especial el MSB. Como aparte, sería de interés la creación de un Shift Register de modo PISO el cual no necesite un eslabon extra de la cadena de Flip-Flops.