LABORATORIO SORTING NETWORK

Estudiantes

Karol Daniela Alzate Mejía 1192735374
Sergio Josue Rodríguez Taborda 1007398380

Profesor

Fredy Alexander Rivera Vélez

Arquitectura de Computadores y Laboratorio
Universidad de Antioquia
Ingeniería de Sistemas
2020-1

Introducción

En este informe se mostrará la solución del laboratorio de Arquitectura de computadores en el cual se realiza un circuito capaz de organizar de manera ascendente un conjunto de 8 números de 8 bits cada uno, los cuales estarán representados en el complemento a 2.

Lo primero que se realizó fue el comparador de dos números de un 1 bit para luego incorporarlo por medio de un diseño jerárquico en el comparador de dos números de 8 bits, allí se devolvía si un número era mayor, menor o igual que el otro. Ya con esto, se implementó un componente que pudiera ayudar a hacer los cambios en la red de ordenamiento, en otras palabras, que ayudara a colocar los números de manera ascendente. Para esto se creó un componente que debía dejar pasar ciertos valores dependiendo del valor de otra de las variables, se realizó inicialmente para números de un solo bit, pero, este se reutilizó para crear uno de 4 bits hasta llegar a los 8 bits. Luego de esto se unió el comparador de dos números de 8 bits con el nuevo componente para así crear otro componente encargado de hacer los cambios y por último, acomodarlos en otros circuitos para crear la red de ordenamiento.

Metodología

Para esta práctica se realizó un análisis de cómo decir si un número era mayor, menor o igual que otro teniendo en cuenta que los números negativos estaban expresados en el complemento a 2. Por lo tanto, se hicieron diferentes ejemplos para encontrar la mejor manera para solucionarlos. A continuación se podrá ver un ejemplo de los números expresados en bits, decimales sin signo (como la máquina los verá) y en el complemento a 2 (como se verán para este ejercicio).

Bits	Decimal	Complemento a 2
000	0	0
001	1	1
010	2	2
011	3	3
100	4	-4
101	5	-3
110	6	-2
111	7	-1

Con la tabla anterior se harán ejemplos de comparaciones con los números en complemento a 2 para observar qué métodos usar en la práctica para saber si un número es mayor, menor o igual que el otro.

Para números de igual signo

- Si los números son negativos: se tomarán como ejemplo el número -1 y el -3 para realizar la comparación.

Bits	Decimal	Complemento a 2
101	5	-3
111	7	-1

En complemento a 2: -1 **>** -3
En decimal, que sería su otra representación: 7 **>** 5

Concluyendo así, que, cualquier conjunto de números negativos expresados en complemento a 2 que se comparen entre sí, se pueden comparar igual a que si estos fueran números representados en un su forma decimal.

- Si los números son positivos: se tomarán como ejemplo el número 0 y el 2 para realizar la comparación.

Bits	Decimal	Complemento a 2
000	0	0
011	3	3

En complemento a 2: 0 **<** 3
En decimal: 0 **<** 3

Con este conjunto de datos pasa lo mismo que el anterior, al número representarse de igual manera en ambas representaciones, se puede realizar la comparación desde el punto de vista decimal.

Teniendo en cuenta los resultados obtenidos de las comparaciones anteriores, se concluye que, si los números a compararse tienen el mismo signo (sea negativo o positivo), se puede usar el mismo procedimiento para ambos casos, es decir, compararlos como si estos estuvieran representados en su forma decimal.

Para números de diferente signo

Un número es automáticamente menor que otro cuando tiene su bit más significativo es un uno. Por ejemplo:

Bits	Decimal	Complemento a 2
001	1	1
100	4	-4

Aquí se nota claramente como, al número -4 al tener su bit más significativo igual a 1, lo hace menor que el número 1, que tiene su bit más significativo igual a 0.

Comparador de dos números de un bit

Dado que las condiciones que pueden dar como resultado de este comparador es si un número es mayor, igual o menor que otro, se diseñó la siguiente semántica para expresar estas condiciones:

C1o	C2o	Significado
0	0	A = B
1	0	A > B
0	1	A < B
1	1	No aplica

A y B representa los números que se van a comparar en este circuito y C1o y C2o serán las salidas de este.

Para este componente se eligió evaluar los números desde su bit menos significativo hasta el más significativo, entonces, el circuito aparte de tener las entradas de los números A y B, tendrá también las entradas C1 y C2 que se encargarán de decirnos si el número que estaba anterior a ese era mayor, menor o igual. Se explicará mejor con la siguiente tabla:

A	B	C1	C2	C1o	C2o
1	0	X	X	1	0
0	1	X	X	0	1
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0

1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
0	0	1	1	X	X
1	1	1	1	X	X

Cuando en la tabla A es igual a 1 y B es igual a 0, no importa el valor que traigan C1 y C2, la salida será con C1o igual a 1 y C2o igual a 0, lo que en la tabla de definiciones significa que A > B, esto se da dado que, si el bit que está más a la izquierda es igual a 1, automáticamente ese número se convierte en el mayor. Se ilustrará mejor con un ejemplo:

N°1: 00111
N°2: 01000

Si se analizan los números anteriores con base en la tabla anterior empezando desde el bit menos significativo de derecha a izquierda. Cuando se llega al bit 2, N°1 > N°2, pero cuando se analiza el bit número 3, se ve que N°2 > N°1, dado que, así todos los números del bit 2 a la derecha sean igual a 1, nunca será igual o mayor a otro número que tenga el bit 3 igual a 1. En este caso, el N°1 es igual a 7, mientras que el bit N°2 es igual a 8, mostrando así lo que afirmamos anteriormente de que N°2 > N°1.

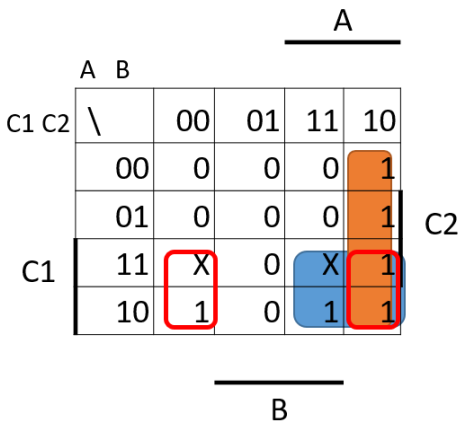
Para el siguiente caso en el que A es igual a 0 y B es igual a 1, sucede exactamente los mismo que el anterior, así A sea mayor que B en los bits anteriores, al B ser igual a 1 en el siguiente bit, B pasa a ser mayor que A.

Para los casos en que A y B son iguales a 0 o ambos son iguales a 1, es decir, que ambos bits tienen los mismos valores, se lleva el valor que tengan C1 y C2 a C1o y C2o, es decir si, por ejemplo, C1 es igual a 1 y C2 es igual a 0, C1o tomará el valor de C1, que sería en este caso 1 y C2 tomará el valor de C2, es decir, 0.

Hay una excepción a esta regla y es cuando C1 y C2 son ambos iguales a 1, este es el caso en el que en la tabla de definición que planteamos al principio, es un valor que no aplica, es decir, es un valor que de cierta manera, C1 y C2 nunca van a devolver, por lo tanto, a estas dos situaciones se les aplica *don't care* a las salidas de C1o y C2o.

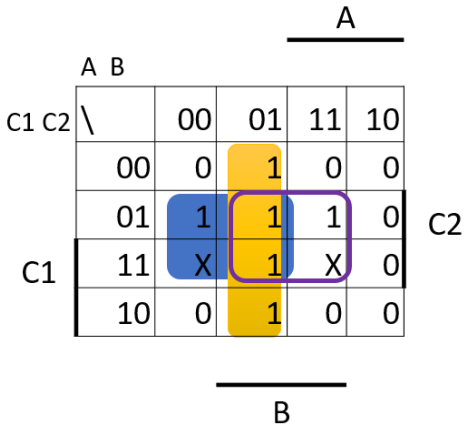
Mapas de Karnaugh

Para C1o



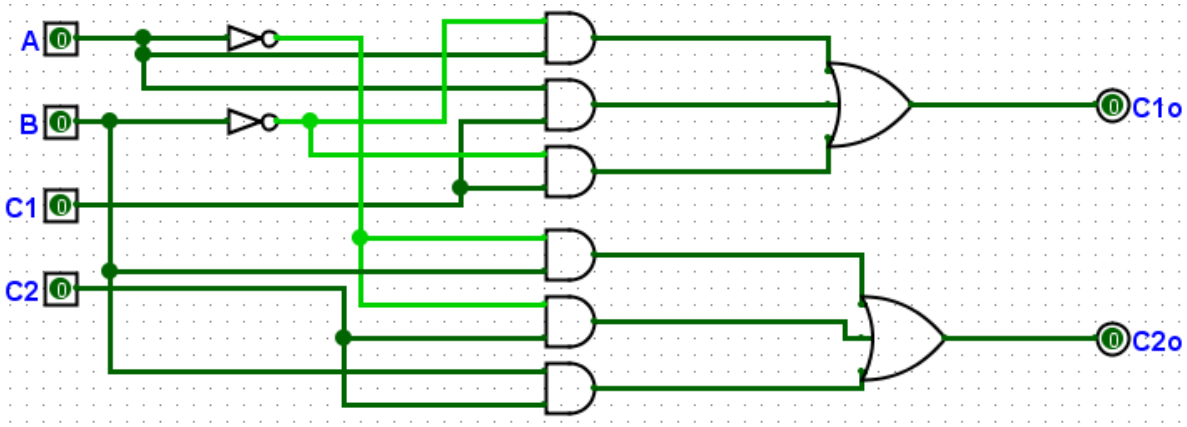
C1o = AB’ + AC1 + B’C1

Para C2o



C2o = A’B + A’C2 + BC2

Circuito

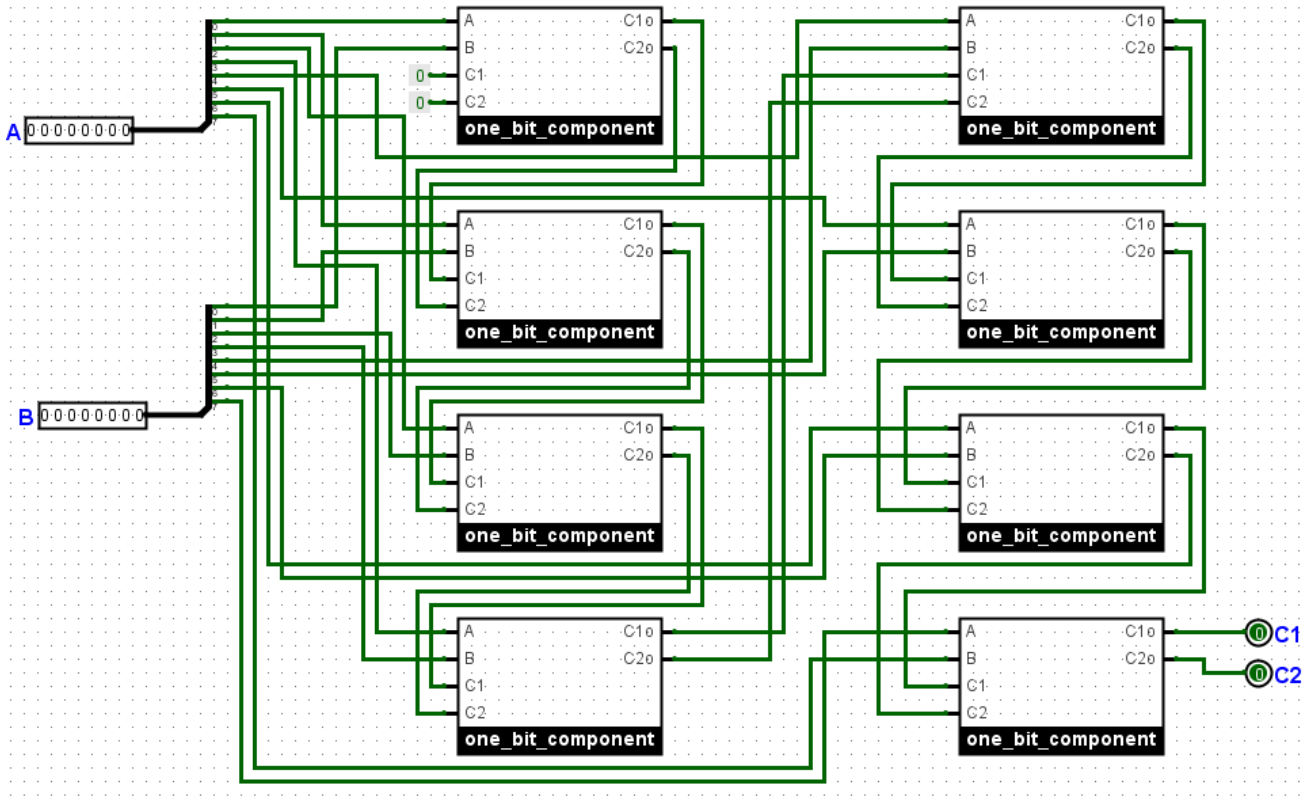


Comparador de dos números de ocho bits

Para realizar el comparador de dos números de ocho bits, se utilizaron 8 componentes del comparador de dos números de un bit siguiendo el esquema que ya habíamos mencionado, se avanza desde el bit menos significativo hasta el más significativo, es decir, de derecha a izquierda, este componente genera una salida C1o y C2o que indica cuál número es mayor, menor o igual al otro, ya con esto, avanzamos al siguiente bit y estas salidas serán las entradas C1 y C2 de este nuevo bit que estamos comparando.

Sólo hay una diferencia que se presenta y es que, como se mencionó en la parte de la metodología, cuando los números tienen diferentes signos, es menor aquel que tenga el bit significativo igual a 1. Según la lógica que se implementó en el comparador de dos números de un bit, esto sería al revés, aquel que tenga el número significativo igual a 1, sería el mayor de los dos números, por lo tanto, cuando se va a comparar el último bit con este componente se intercambian los valores de los bits de A y B para que ahora sí devuelva como mayor al que tenga el bit más significativo igual a 0.

Circuito



Red de ordenamiento

Para la red de ordenamiento se siguió las indicaciones según como se ilustraba en la práctica que mostraba una red de ordenamiento que acarrea cuatro valores, esta se utilizó dos veces (se especificará más adelante) y se crearon otro dos componentes para hacer posible el cambio de los valores en la red y estos puedan quedar ordenados de manera ascendente.

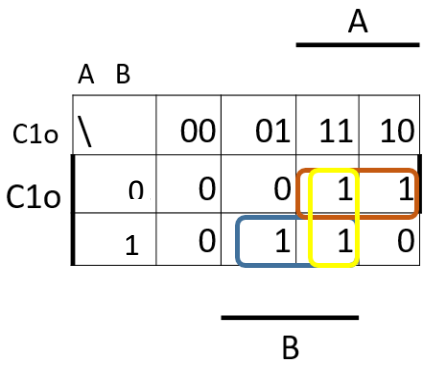
Componente de retorno de un bit

En este componente, se tomó los valores que devolvía el comparador de 8 bits, es decir, C1o y C2o que son los valores que nos indican si un número es mayor, menor o igual, pero cuando se analizaron estos valores se notó que no era necesario usar ambos ya que, si los números son iguales, no hay que cambiarlos de lugar al igual que si A < B, en estas dos opciones en las que no hay que realizar cambios, se notó que C1o era igual a 0, y cuando A > B, que sí hay que realizar el cambio, C1o era igual a 1. Por lo tanto, sólo se usó la salida de C1o para realizar el componente.

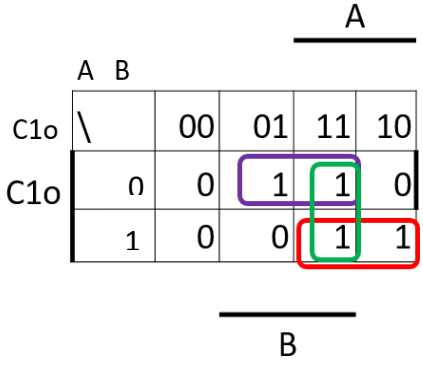
Por otro lado, a la salida del componente se le aplicó una lógica sencilla en la que, si los números eran iguales o A < B, que es lo mismo a decir que C1o es igual a 0, lo que hará es devolver esos valores en el mismo orden que entraron y, cuando C1o es igual a 1, que es cuando hay que realizar el cambio, se devuelven los valores de A y B intercambiados, para esto se hizo una tabla de verdad en la que se podrá visualizar mejor esto.

C1o	A	B	LD	GD
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

Mapas de Karnaugh

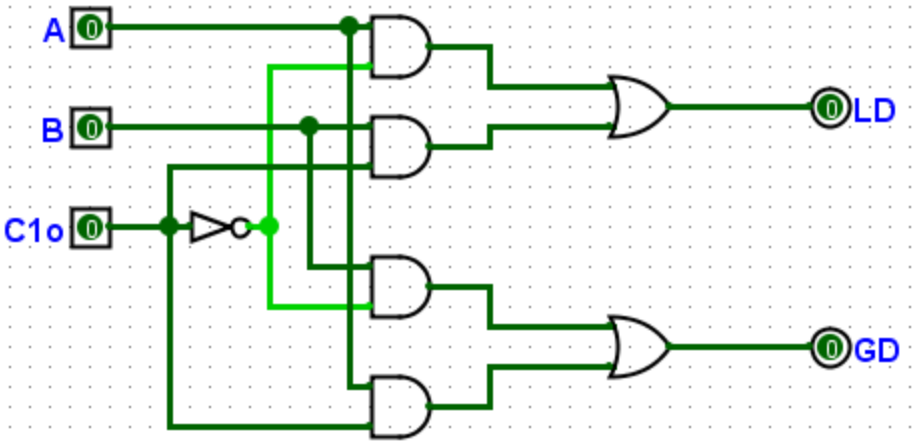


LD = AC1o' + BC1o



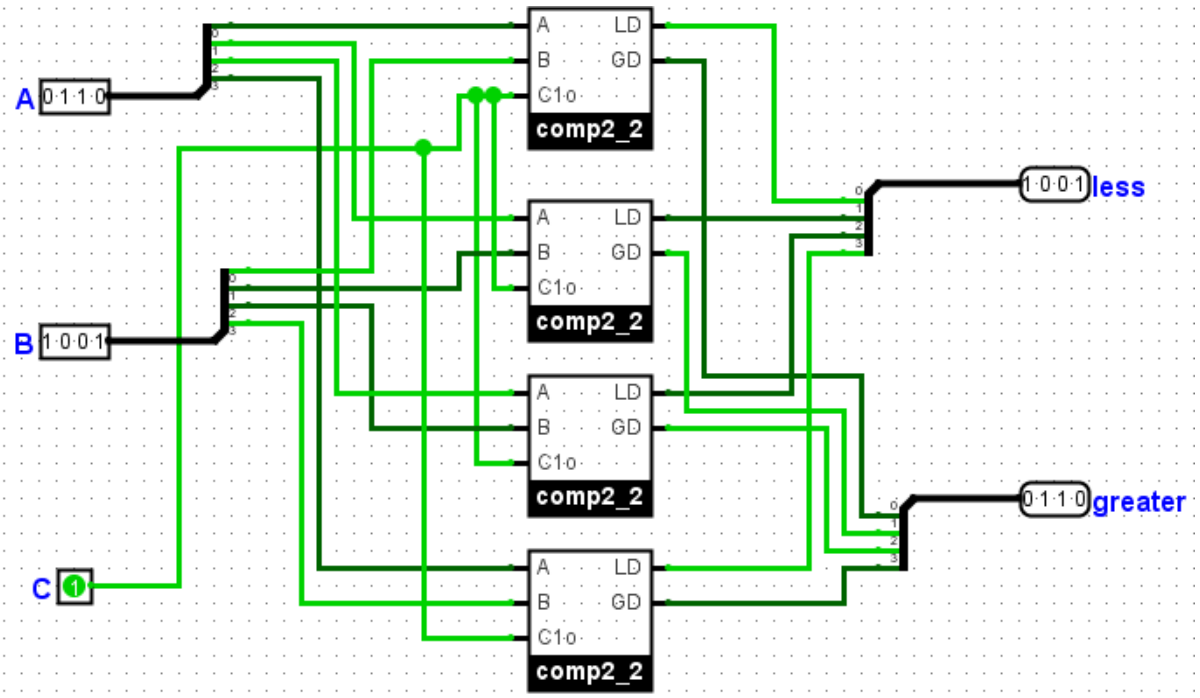
GD = BC1o' + AC1o

Circuito



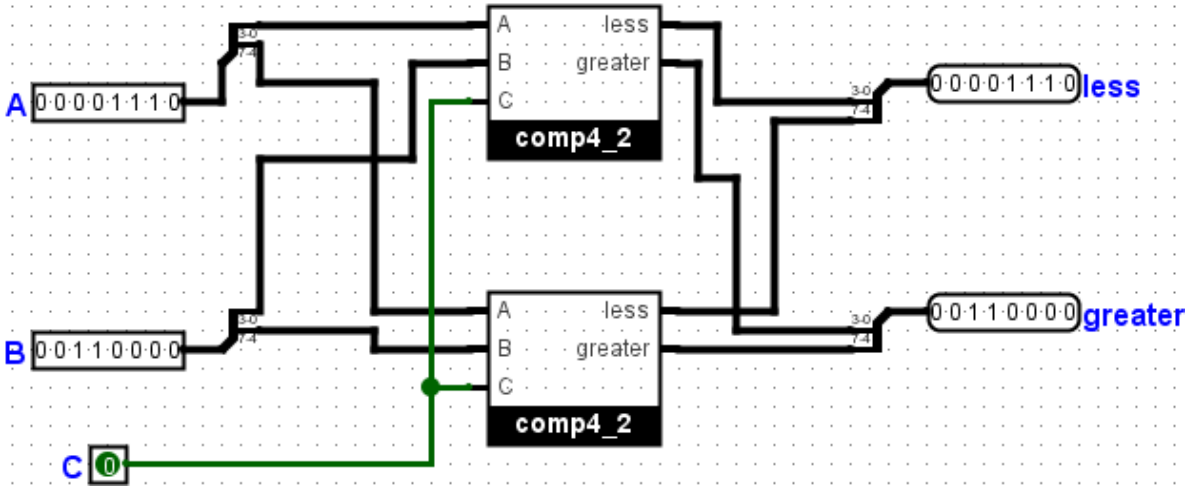
Componente de retorno de cuatro bits

Este componente lo que hará es tomar cuatro bits de nuestros dos números y utilizará el componente para retornar los bits, mirando el bit 0 del número A y el número B de acuerdo al valor que tenga C, y hace esto con todos los bits, devolviendo entonces, por ejemplo, el bit 0 del número que es menor y bit 0 del número que es mayor, y así sucesivamente, luego, estos bits se juntan en el orden adecuado para devolver los cuatros bits del número que es menor y los otros cuatro del número que es mayor.



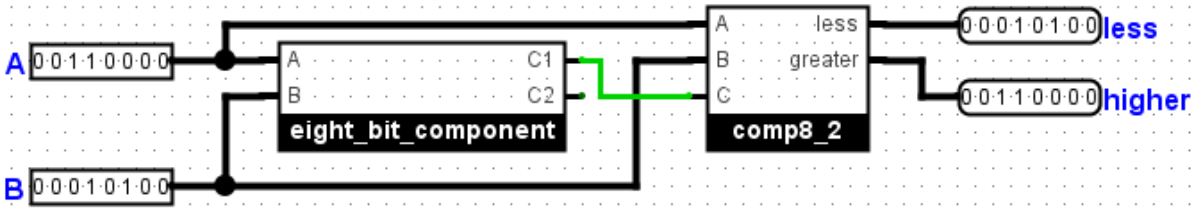
Componente de retorno de ocho bits

Este componente lo único que hace es llamar al componente de retorno de cuatro bits, lo hace mandando los cuatro primeros bits de los dos números a uno, y luego le manda los cuatro últimos bits de cada número a otro, luego se organizan los bits del número que es menor y del número que es menor.



Componente switch

Este componente une el componente del comparador de ocho bits con el componente de retorno de ocho bits, ya que este último necesita la salida C1, que es la que nos dice cuál número es mayor, menor o igual que el otro, para luego devolvernos los números del número que es mayor y del número que es menor.



Componente de ordenamiento de cuatro números

Para este componente se replicó exactamente la red de ordenamiento que acarrea cuatro valores, en donde las líneas donde se comparan los números, se colocaron los componentes de switch que son los encargados de definir cuál número es mayor o menor. A continuación se mostrará la red de ordenamiento y el circuito.

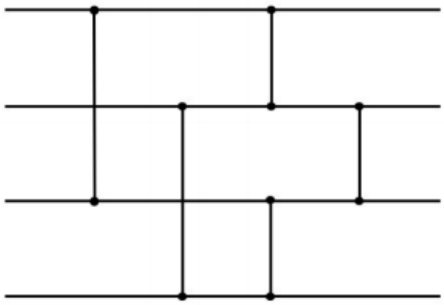
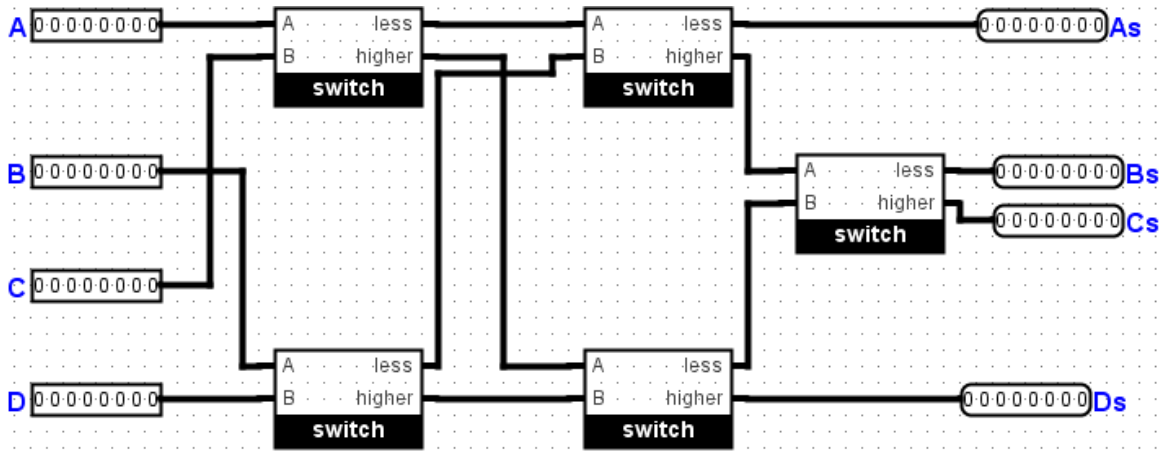


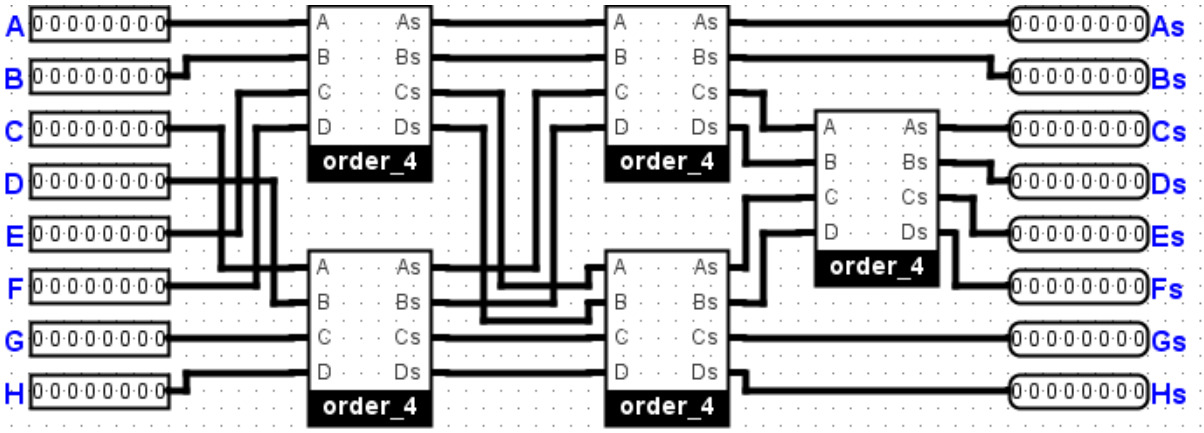
Figura 1. Red de ordenamiento que acarrea cuatro valores

Circuito



Componente de ordenamiento de 8 números

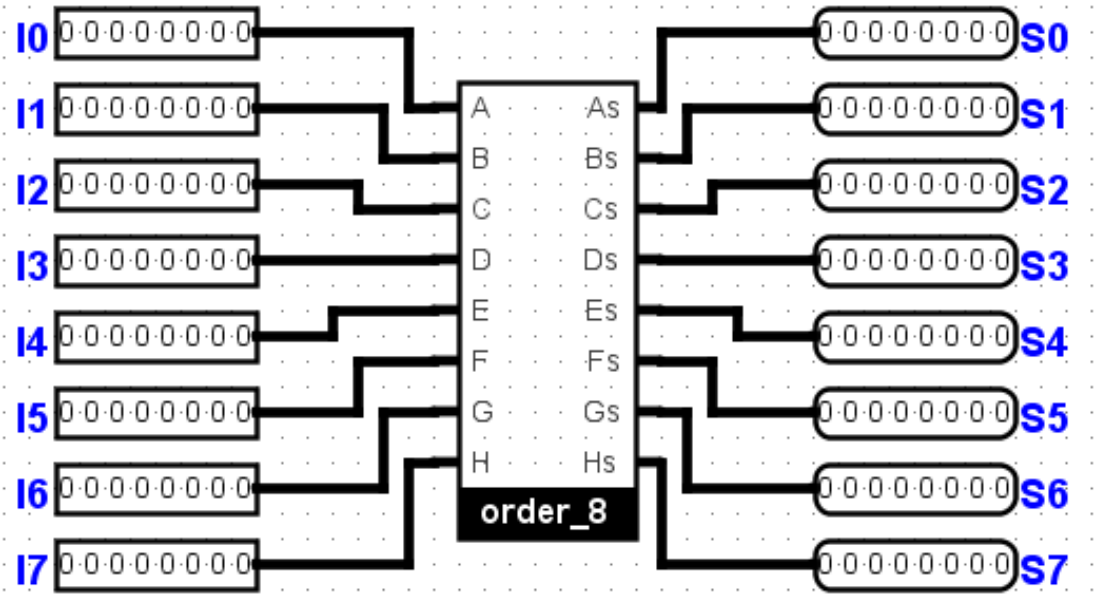
Para este componente se realizó un análisis y se notó que la red de ordenamiento que acarrea cuatro valores también se podía aplicar para ordenar los ocho números utilizando el componente de ordenamiento de 4 números.



Se sigue el mismo orden que la red de ordenamiento, sólo que mandando los valores por pares, y como la salida de los componentes son los números ya ordenados no hay problema en pensar que por mandar dos números en vez de uno, entre estos no se hará el ordenamiento, no, porque justamente el componente de ordenamiento de 4 números, los está organizando todos, al ya tener los valores en orden se siguen mandando de igual manera por pares al resto de componentes que a su vez, vuelven a sacar los valores organizados hasta que termina el ordenamiento.

Resultado

Ya llamando al componente de ordenamiento de ocho números se puede sólo definir las entradas y salidas, y el mostrará los números ordenados de manera ascendente.

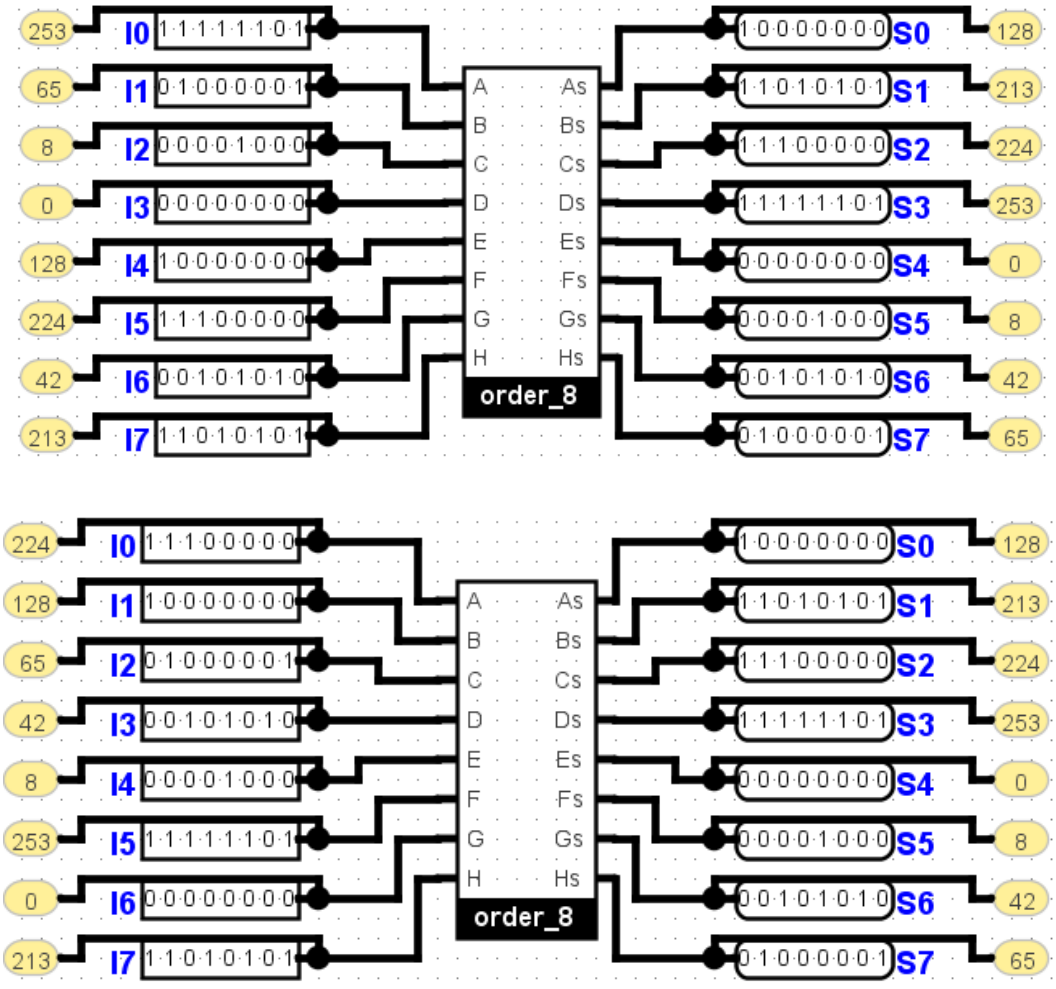


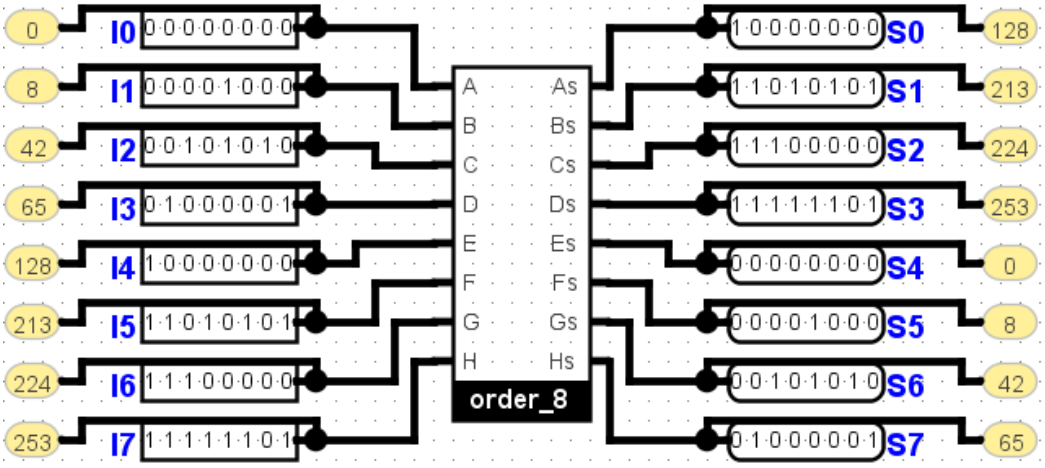
Simulación del sistema

Para las simulaciones se utilizará el la herramienta de *ver* de logisim con los números expresados sin signo para poder mostrar lo que habíamos probado en la tabla de la metodología. Pondremos los siguientes números en diferentes órdenes de entrada y nos debería devolver lo mismo en cada uno de ellos.

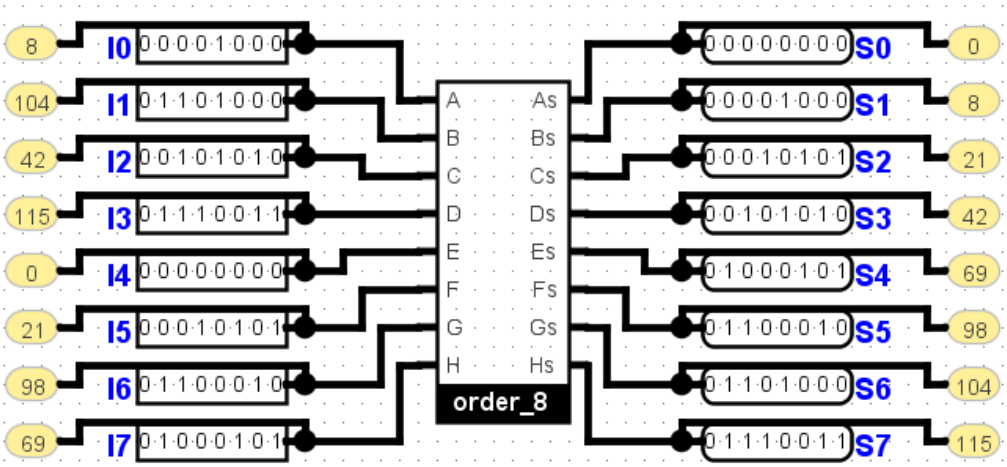
Bits	Decimal	Complemento a 2	Ordenados complemento a 2	Como aparecen en logisim
00000000	0	0	-128	128
00001000	8	8	-43	213
00101010	42	42	-32	224
01000001	65	65	-3	253
10000000	128	-128	0	0
11010101	213	-43	8	8
11100000	224	-32	42	42
11111101	253	-3	65	65

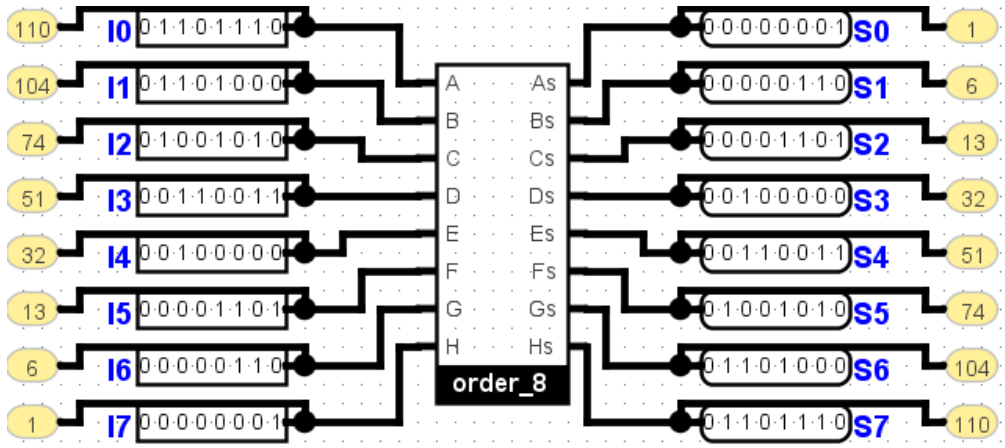
Comparaciones





Como se puede visualizar, a pesar de cambiar el orden de entrada de los datos, siempre se devolvieron los datos en orden tal como estaba definido en la tabla anterior. Ahora haremos lo mismo pero con números únicamente positivos. No hay necesidad de la tabla ya que los números en complemento a 2 si son positivos son iguales a los decimales.

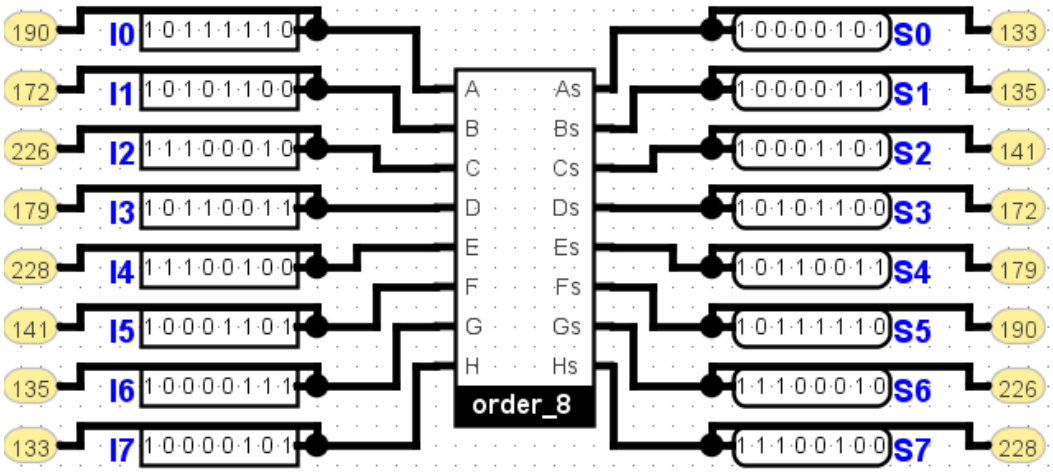


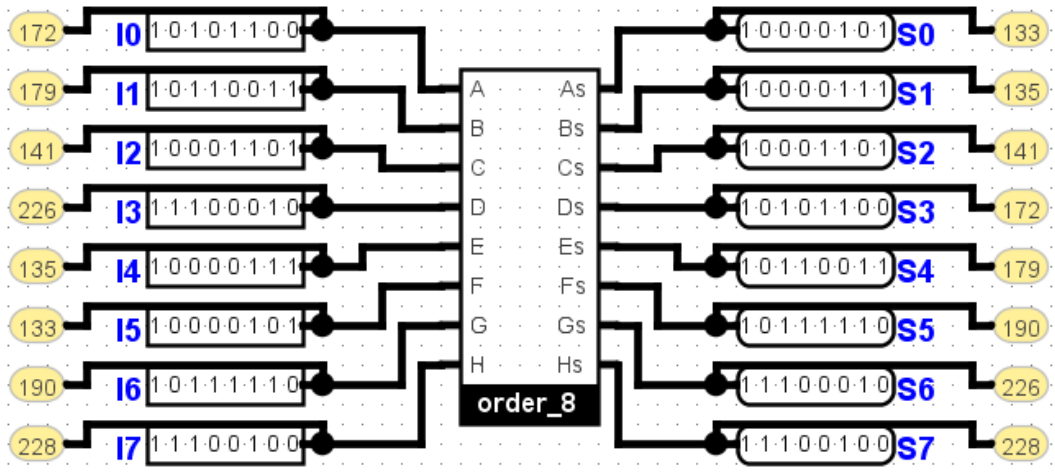


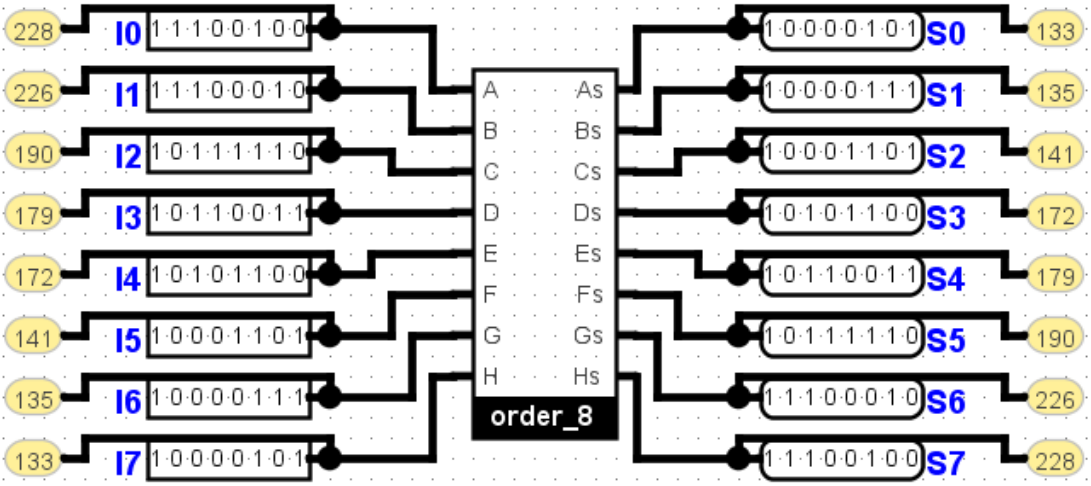
Ahora se realizará con valores únicamente negativos en diferentes órdenes y se hará de acuerdo a los números en la siguiente tabla.

Bits	Decimal	Complemento a 2	Ordenados complemento a 2	Como aparecen en logisim
10000101	133	-123	-123	133
10000111	135	-121	-121	135
10001101	141	-115	-115	141
10101100	172	-84	-84	172
10110011	179	-77	-77	179
10111110	190	-66	-66	190
11100010	226	-30	-30	226
11100100	228	-28	-28	228

Comparaciones







Análisis y conclusiones

- Al realizar un análisis en cómo los datos son representados en la manera en la que la máquina lo ve y en cómo se necesita para el ejercicio, en este caso complemento a 2, se encontró que la solución se podía hacer de manera más simple ya que se encontraron similitudes a cómo se podían comparar los datos en decimal y en el complemento, por lo que no había que hacerles mucho procesamiento.
- Cuando se aborda el problema desde niveles más profundos en el que por ejemplo sólo se compara o se retorna un bit en vez de muchos al mismo tiempo, es más sencillo implementar estos componentes para crear soluciones a problemas más complejos haciendo uso de ellos ya que se puede ver de una manera más clara cómo se pueden utilizar.
- Así como la red de ordenamiento que acarrea cuatro valores se pudo utilizar también para realizar el ordenamiento de los ocho números, es necesario analizar la lógica que manejan estos componentes porque pueden servir de ayuda para implementar la misma solución en niveles más altos sin necesidad de tener que aplicar una solución diferente.