# INSTITUTO TECNOLÓGICO DE COSTA RICA

## ESCUELA DE INGENIERÍA ELECTRÓNICA

## DISEÑO LÓGICO

## II SEMESTRE

# TAREA 2


**PROFESOR: PABLO DANIEL MENDOZA PONCE**

**ESTUDIANTE: JOSÉ JULIÁN CAMACHO HERNÁNDEZ**

**CARNÉ: 2019201459**


**1 DE OCTUBRE DE 2021**

# 1

sábado, 25 de septiembre de 2021 19:01

José Julián Camacho Hernández
2019201459

1. Basado en lo visto sobre codificadores: obtenga las ecuaciones booleanas simplificadas (por medio de algebra booleana y mapas K) que describen un codificador de prioridad de 8 a 3.

Tabla de verdad:

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ | $Q_3$ | $Q_2$ | $Q_1$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | X | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | X | X | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | X | X | X | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | X | X | X | X | 1 | 0 | 0 |
| 0 | 0 | 1 | X | X | X | X | X | 1 | 0 | 1 |
| 0 | 1 | X | X | X | X | X | X | 1 | 1 | 0 |
| 1 | X | X | X | X | X | X | X | 1 | 1 | 1 |

Los términos X no se toman en cuenta porque no afectan el resultado:

$$X \equiv (D+\bar{D}) \equiv 1 \quad \Rightarrow \quad DX \equiv D$$

$$Q_3 = \bar{D_7}\,\bar{D_6}\,\bar{D_5}\,D_4 + \bar{D_7}\,\bar{D_6}\,D_5 + \bar{D_7}\,D_6 + D_7$$

$$= \bar{D_7}\left(\bar{D_6}\left(\underbrace{\bar{D_5}\,D_4 + D_5}\right) + D_6\right) + D_7 \qquad \text{T8}$$

$$= \bar{D_7}\left(\bar{D_6}\left(D_4 + D_5\right) + D_6\right) + D_7 \qquad \text{Simplificación}$$

$$= \bar{D_7}\left(\left(D_4 + D_5\right) + D_6\right) + D_7 \qquad \text{Simplificación}$$

$$= \left(\left(D_4 + D_5\right) + D_6\right) + D_7 \qquad \text{Simplificación}$$

$$\boxed{\therefore\ Q_3 = D_4 + D_5 + D_6 + D_7}$$

$$Q_2 = \overline{D_7}\,\overline{D_6}\,\overline{D_5}\,\overline{D_4}\,\overline{D_3}\,D_2 + \overline{D_7}\,\overline{D_6}\,\overline{D_5}\,\overline{D_4}\,D_3 + \overline{D_7}\,D_6 + D_7$$

$$= \overline{D_7}\left\{\overline{D_6}\left[\overline{D_5}\,\overline{D_4}\,(\underline{\overline{D_3}\,D_2 + D_3})\right] + D_6\right\} + D_7 \qquad \text{T8}$$

$$= \overline{D_7}\left\{\underline{\overline{D_6}\left[\overline{D_5}\,\overline{D_4}\,(D_2 + D_3)\right] + D_6}\right\} + D_7 \qquad \text{Simplificación}$$

$$= \overline{D_7}\left\{\left[\overline{D_5}\,\overline{D_4}\,(D_2 + D_3)\right] + D_6\right\} + D_7 \qquad \text{Simplificación}$$

$$= \left\{\overline{D_5}\,\overline{D_4}\,(D_2 + D_3) + D_6\right\} + D_7 \qquad \text{Simplificación}$$

$$\boxed{\therefore\ Q_2 = \overline{D_5}\,\overline{D_4}\,D_2 + \overline{D_5}\,\overline{D_4}\,D_3 + D_6 + D_7}$$

$$Q_1 = \overline{D_7}\,\overline{D_6}\,\overline{D_5}\,\overline{D_4}\,\overline{D_3}\,\overline{D_2}\,D_1 + \overline{D_7}\,\overline{D_6}\,\overline{D_5}\,\overline{D_4}\,D_3 + \overline{D_7}\,\overline{D_6}\,D_5 + D_7$$

$$= \overline{D_7}\left\{\overline{D_6}\left[\overline{D_5}\left(\overline{D_4}\left(\underline{\overline{D_3}\,\overline{D_2}\,D_1 + D_3}\right)\right) + D_5\right]\right\} + D_7 \qquad \text{T8}$$

$$= \overline{D_7}\left\{\overline{D_6}\left[\overline{D_5}\left(\overline{D_4}\left(\overline{D_2}\,D_1 + D_3\right)\right) + D_5\right]\right\} + D_7 \qquad \text{Simplificación}$$

$$= \overline{D_7}\left\{\overline{D_6}\left[\left(\overline{D_4}\left(\overline{D_2}\,D_1 + D_3\right)\right) + D_5\right]\right\} + D_7 \qquad \text{Simplificación}$$

$$= \left\{\overline{D_6}\left[\left(\overline{D_4}\left(\overline{D_2}\,D_1 + D_3\right)\right) + D_5\right]\right\} + D_7 \qquad \text{Simplificación}$$

$$= \overline{D_6}\left(\overline{D_4}\left(\overline{D_2}\,D_1 + D_3\right)\right) + \overline{D_6}\,D_5 + D_7$$

$$\boxed{\therefore\ Q_1 = \overline{D_6}\,\overline{D_4}\,\overline{D_2}\,D_1 + \overline{D_6}\,\overline{D_4}\,D_3 + \overline{D_6}\,D_5 + D_7}$$

# 2

sábado, 25 de septiembre de 2021 19:01

José Julián Camacho Hernández
2019201459

2. Desarrolle un sumador de números en formato "signo y magnitud" de 4 bits. Haga uso de un circuito de suma de riso (ripple-carry-adder) como base. ¿Qué componentes deben añadirse para poder realizar la suma en este formato?

| | $A_0$ | $A_1$ | $A_2$ | $A_3$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| -1 | 1 | 0 | 0 | 1 |
| -2 | 1 | 0 | 1 | 0 |
| -3 | 1 | 0 | 1 | 1 |
| -4 | 1 | 1 | 0 | 0 |
| -5 | 1 | 1 | 0 | 1 |
| -6 | 1 | 1 | 1 | 0 |
| -7 | 1 | 1 | 1 | 1 |

Considera casos:

$A + B$
$(-A) + B$
$A + (-B)$
$(-A) + (-B)$

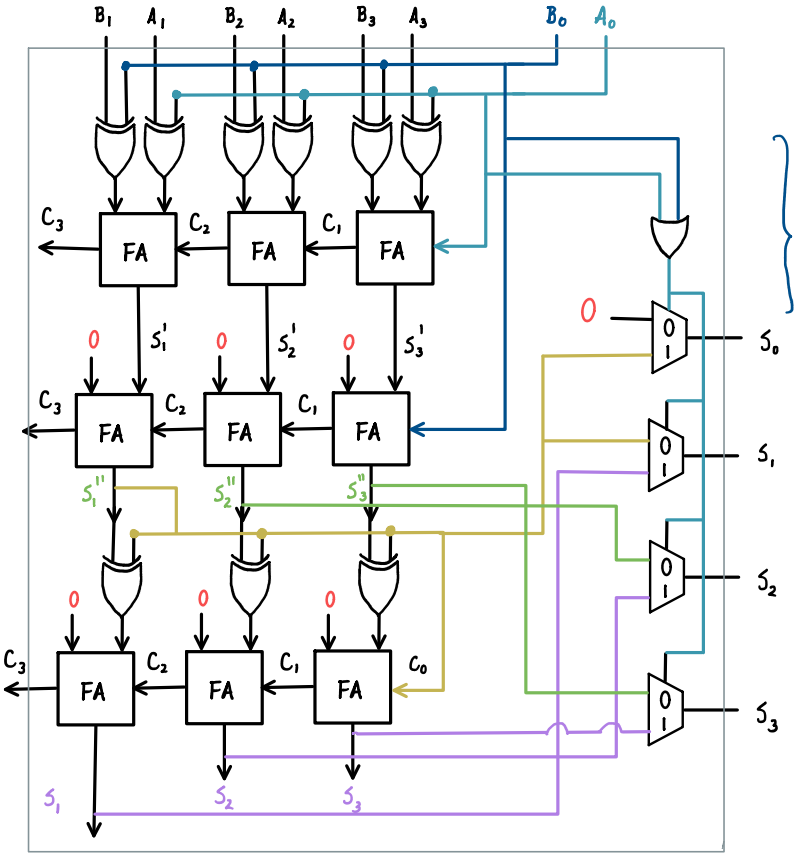
Sin overflow

En esta sección, si el MSB del número es 1 se convierte a complemento a dos, sino se pasa el número normal.

Se realiza una suma extra para considerar si los dos sumandos se convirtieron, por lo que ambos aportan $C_{in} = 1$ (Sino suma cero y no afecta el resultado)

$S_1''$, $S_2''$ y $S_3''$ son el resultado normal o en compl. a dos.

Se hace esta última sección para pasarlo de nuevo a signo y magnitud. (Se invierte y se le suma 1)




El OR y los mux se encargan de considerar el caso en que los dos sumandos son positivos. (Para dar el resultado sin hacer la última sección)

# 3

sábado, 25 de septiembre de 2021 19:01

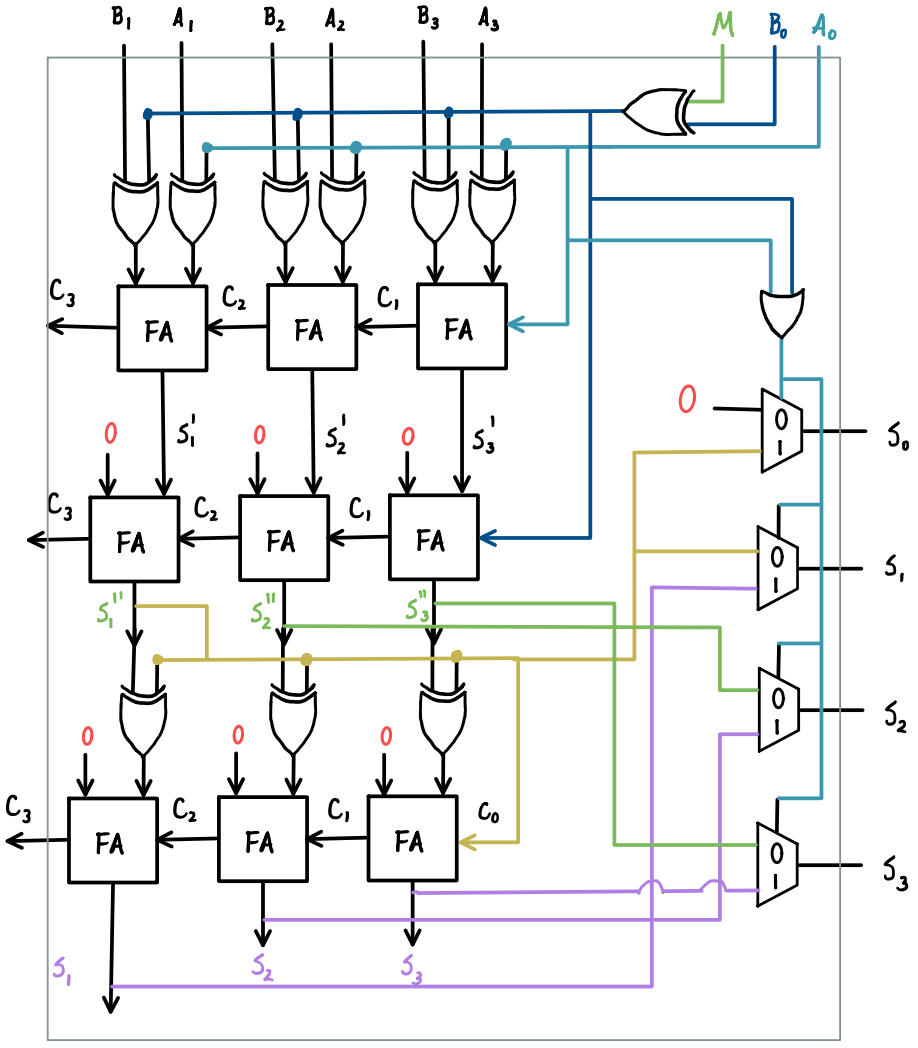
José Julián Camacho Hernández
2019201459

3. Agregue la lógica necesaria para poder obtener un sumador/restador de datos con signo y magnitud.

Considera casos:

$$A - B = A + (-B)$$
$$(-A) - B = (-A) + (-B)$$
$$A - (-B) = A + B$$
$$(-A) - (-B) = (-A) + B$$

Para que también sea un restador se debe cambiar el signo de B. Como son números en compl. a dos se invierte el MSB ($A_0$, $B_0$).




Para convertirlo en un sumador/restador se agrega la señal de selección M.

Si M es 0:
Pasa $B_0$ por lo que es un sumador.

Si M es 1:
Pasa $\overline{B_0}$ por lo que es un restador.

# 4

sábado, 25 de septiembre de 2021 19:01

José Julián Camacho Hernández
2019201459

4. Describa en que casos se generan elementos de memoria NO deseados en diseños de lógica combinacional en verilog.

En lógica combinacional, se pueden generar "latches" o elementos de memoria no deseados en ciertos casos donde las salidas no dependen únicamente de las entradas. Por ejemplo, en un demux, si no se tiene el cuidado de tener solo un salida en alto, tras cambios en las entradas es posible que se guarde el valor de una salida; por lo que es necesario poner en bajo todas las salidas cada vez que ocurran cambios en las entradas.

Otro de los casos donde esto puede suceder es cuando en un bloque condicional no se contempla alguno de los posibles casos. Si la condición coincide con el caso que no se especificó, no se realiza una acción de cambio al variar las entradas y se almacena el último valor, por lo que se crea un elemento de memoria no deseado.

# 5, 7, 8

sábado, 25 de septiembre de 2021 19:01

José Julián Camacho Hernández
2019201459

5. ¿Cuál es la importancia de la "lista de sensitividad" en verilog?

La lista de sensitividad es una lista de señales. Generalmente, se utiliza en la implementación de procesos. Al describir un componente combinacional, se deben incluir en ella todas las entradas del dispositivo. Su importancia radica en que detecta cualquier cambio en las señales que contiene, y con esto es posible evaluar las acciones dentro del procedimiento cada vez que esto suceda.

Ej:

always @$\underbrace{(A \text{ or } B \text{ or } C)}_{\text{lista de sensitividad}}$ begin

7. En que casos se usan los "wires" en verilog.

Los "wires" se utilizan para designar y representar los cables físicos e internos del componente. No almacenan su valor, ya que carecen de memoria. Pueden ser utilizados en los inputs y outputs de un módulo. Además, pueden ser utilizados para representar el valor de un nodo dentro del componente y se pueden operar por medio de álgebra booleana para darle un valor a otro wire o a una salida. Dentro de un procedimiento se puede leer su valor, pero no asignarlo.

8. En que casos se usan los "reg" en verilog para lógica combinacional.

En lógica combinacional los "reg" se utilizan como un tipo básico de señal similar al wire. Pueden usarse para designar las entradas de un bloque. Además, proveen la posibilidad de leer su valor o asignarles uno dentro de un procedimiento, por lo que se utilizan para designar las salidas de un componente si esta fue asignada en un proceso como el always.

# 9

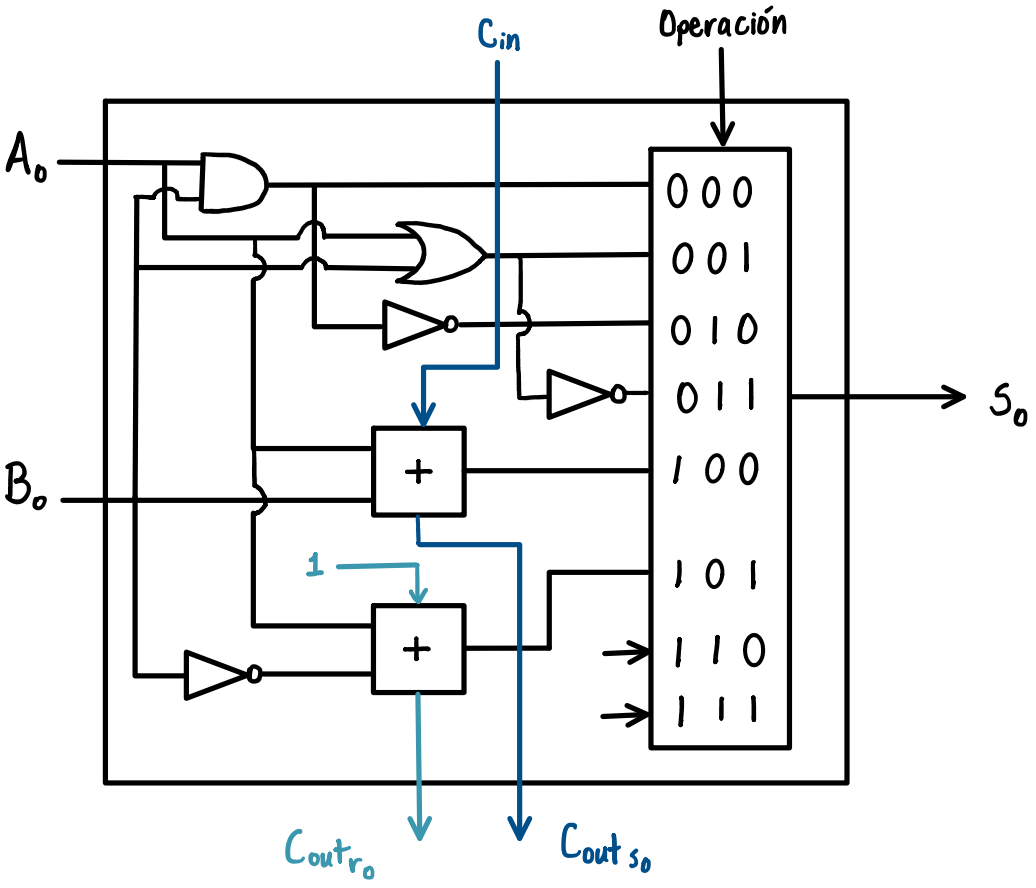
sábado, 25 de septiembre de 2021 19:01

José Julián Camacho Hernández
2019201459

9. Desarrolle una ALU que haga uso de una señal de control de 3 bits. Los datos de entrada son dos operandos A y B, cada uno de 4 bits, la salida también es en 4 bits. De esta manera, las operaciones posibles de dicha ALU (basadas en la señal de control) serían:

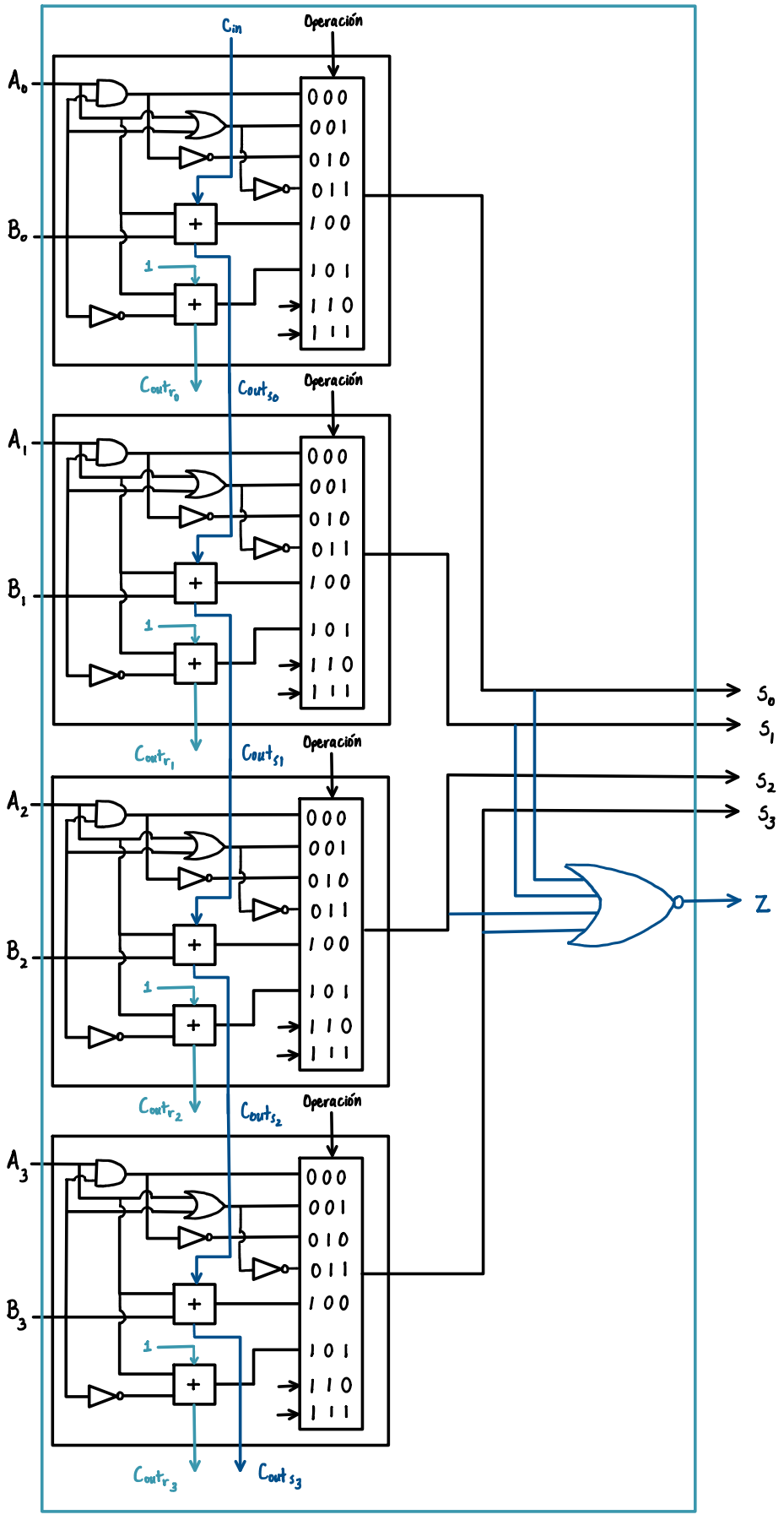
   0: AND
   1: OR
   2: NAND
   3: NOR
   4: SUMA
   5: RESTA

   Además, la ALU posee una bandera de salida cero (cuando el resultado es cero). ¿Qué estrategias de simplificación puede aplicar?



# 9.2

sábado, 25 de septiembre de 2021 19:01

José Julián Camacho Hernández
2019201459

Para hacer la ALU de 4 bits se unen 4 de la ALU de 1 bit anterior: