

# Implementación de una calculadora en una placa FPGA - 2019

## Práctica de laboratorio N°1 Electrónica Digital

Michel Gartner  
michelgartner8@gmail.com

19 de agosto de 2019

### 1. Especificación de requerimientos del sistema

Implementación de una calculadora en una placa Nexys3 Spartan6 XC6LX16-CS324, que permita realizar operaciones de suma, resta y multiplicación de dos números de 4 bits en representación signo y magnitud.

Los números de entrada se mapean a los switches  $sw(3:0)=[M8, V9, T9, T10]$  y  $sw(7:4)=[T5, V8, U8, N8]$ . La salida se representa utilizando el led 'U16' para mostrar el signo, mientras que la magnitud se representa en representación hexadecimal utilizando el módulo 7 segmentos 'P17'. El led encendido indica un número negativo mientras que el led apagado indica un número positivo.

La elección de la operación se realiza utilizando los push-buttons 'C4', 'B8' Y 'D9' para suma, resta y multiplicación respectivamente. Si se presionan más de dos push buttons al mismo tiempo, se producirá una señal de error.

Dado que los números son de 4 bits en representación signo y magnitud, cada número de entrada está contenido en el rango  $[-7, 7]$ . Dado que la salida es solo un dígito hexadecimal, solo se puede representar hasta la magnitud 15. Cualquier número que supere en magnitud a 15 producirá una señal de error.

La señal de error se representa encendiendo los 4 módulos siete segmentos en simultáneo (módulos N16, N15, P18 y P17). En esta condición, el número que se muestra en los módulos siete segmentos y la señal del led no son significativos, dado que se está en una condición de entrada no válida.

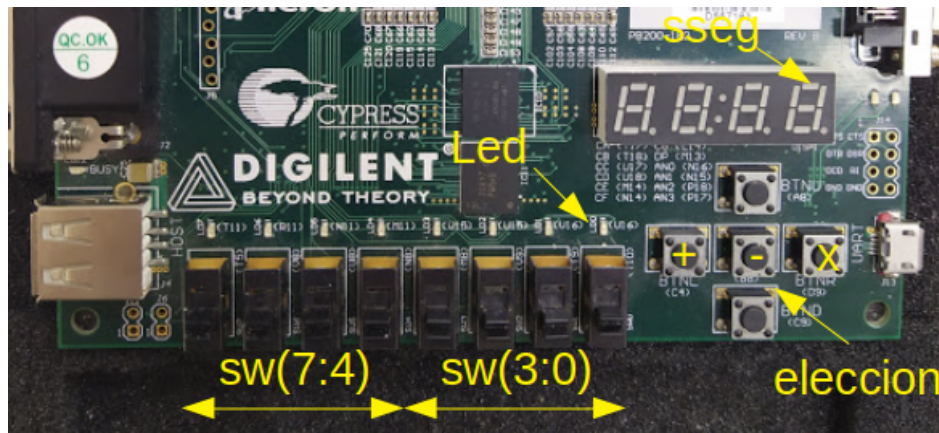


Figura 1: Ubicación de las partes más significativas de la placa Nexys3 para la realización de la calculadora.

En la figura 1 se muestra la ubicación de los switches, los push buttons, los siete segmentos y el led mencionados anteriormente.

## 2. Diseño del módulo TOP

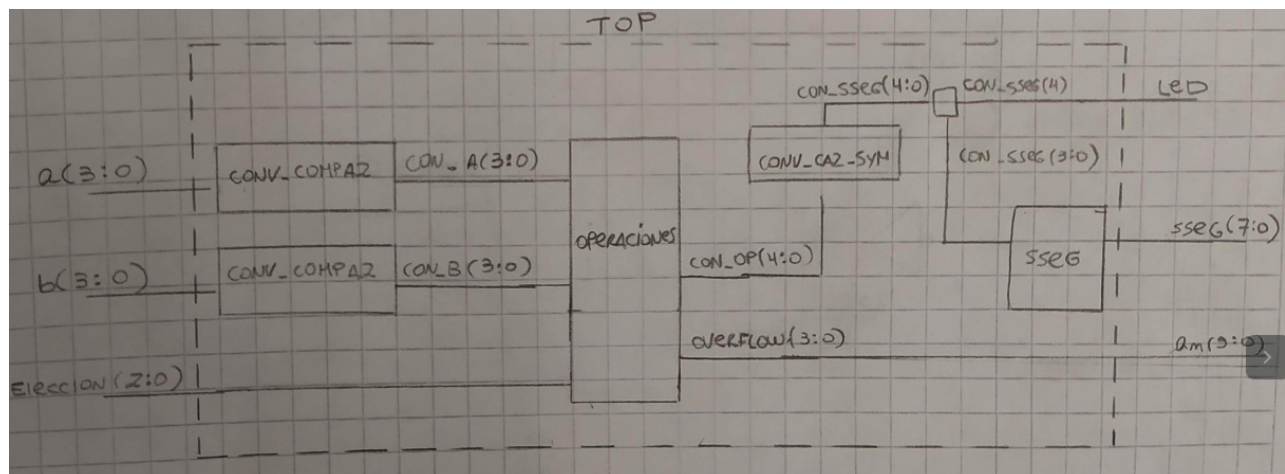


Figura 2: Diagrama de bloques de la lógica del módulo 'top'.

En la figura 2 se muestra esquemáticamente en diagrama de bloques la lógica implementada en el módulo 'top' de la calculadora.

Dado que los números de entrada 'a' y 'b' se ingresan con representación signo y magnitud, el primer paso de este módulo es convertir ambos números a su representación complemento a 2 utilizando el sub-módulo 'conv compa2'. Luego, estos números ingresan al sub-módulo 'operaciones' que realiza la suma, resta o multiplicación utilizando las operaciones nativas de VHDL. Una de las dos salidas del sub-módulo 'operaciones' es el resultado de algunas de las tres operaciones posibles,

que se elige con la entrada 'eleccion'. La otra salida es la señal de overflow que sucede para cualquier resultado de magnitud mayor a 15 y es la salida del módulo top que indica que displays de siete segmentos se encienden. El resultado de la operación pasa por el sub-módulo 'conv ca2 sym' y se convierte a su representación en signo y magnitud. El bit más significativo de éste indica el signo del resultado y va hacia la salida 'led'. Los 4 bits menos significativos indican la magnitud de la operación e ingresan al sub-módulo 'sseg', cuyo resultado es un vector de 8 bits que representan la magnitud del resultado en el 7 segmentos.

El módulo siete segmentos se enciende con un bit '0', los segmentos se encienden con un bit '0' y el led se enciende con un bit '1'.

### 3. Sub-modulo Conversor signo y magnitud a complemento a 2

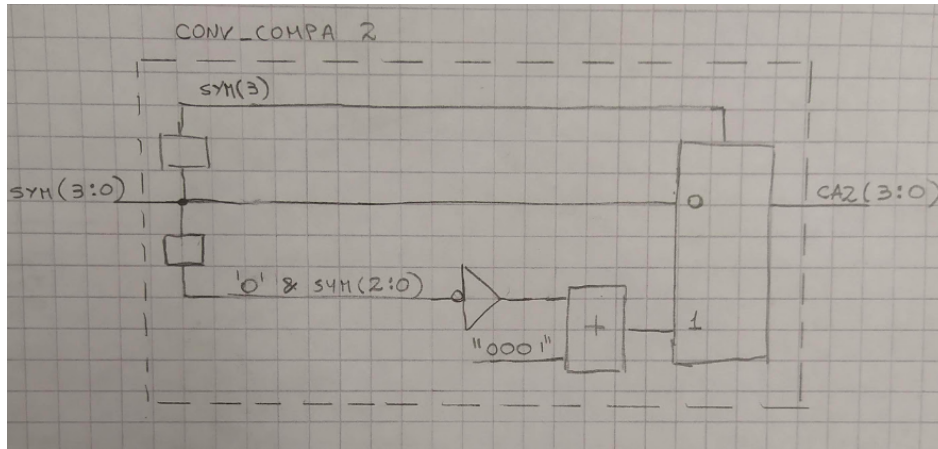


Figura 3: Diagrama de bloques de la lógica del sub-módulo 'conv compa2'.

En la figura 3 se muestra esquemáticamente en diagrama de bloques la lógica implementada en el sub-módulo 'conv compa2' de la calculadora. El bit más significativo de 'sym' representa el signo y se utiliza para elegir la salida del multiplexor, donde '0' indica un número positivo y '1' indica un número negativo. En caso que sea positivo, el valor de entrada es igual al valor al de salida. En caso que sea negativo, se reemplaza el bit más significativo del número por un '0', luego se niega dicho valor y se le suma '0001', obteniendo así la representación en complemento a 2 de un número negativo.

#### 3.1. Implementacion en VHDL

La implementación en VHDL del submódulo operaciones se encuentra en el archivo 'conv\_compa2.vhd'.

### 3.2. Simulación y evaluación

Se realizó un test-bench en el cuál se verificó el correcto resultado para todas las posibles combinaciones de entradas utilizando ciclos 'for'. La implementación del test-bench se encuentra en el archivo 'test\_conv\_compa2.vhd'.

## 4. Sub-módulo Complemento a 2 a signo y magnitud

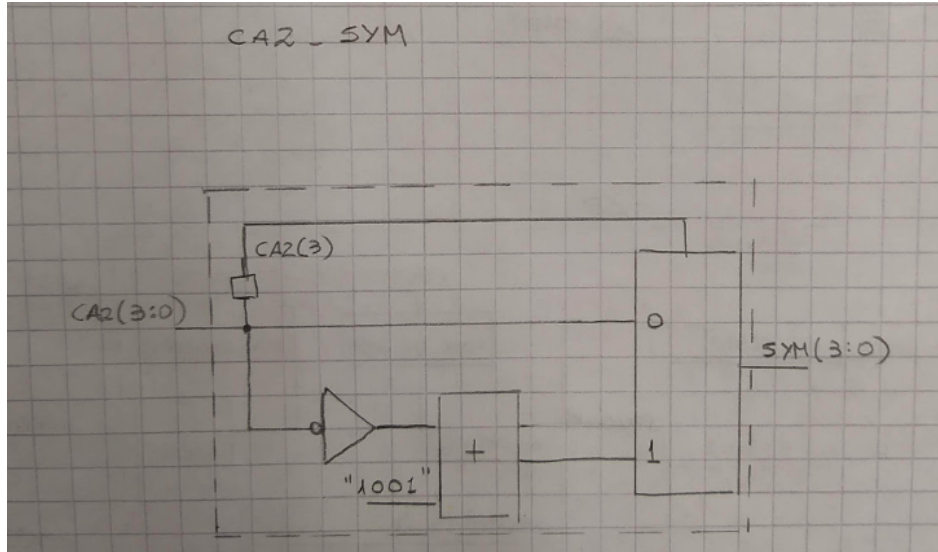


Figura 4: Diagrama de bloques de la lógica del sub-módulo 'conv ca2 sym'.

En la figura 4 se muestra esquemáticamente en diagrama de bloques la lógica implementada en el sub-módulo 'conv compa2' de la calculadora. El bit más significativo de 'ca2' representa el signo y se utiliza para elegir la salida del multiplexor, donde '0' indica un número positivo y '1' indica un número negativo. En caso que sea positivo, el valor de entrada es igual al valor al de salida. En caso que sea negativo, se niega la entrada y se le suma "1001" utilizando la operación de suma nativa de VHDL, obteniendo así la representación en signo y magnitud para un número negativo.

### 4.1. Implementacion en VHDL

La implementación en VHDL del submódulo operaciones se encuentra en el archivo 'conv\_ca2\_sym.vhd'.

### 4.2. Simulación y evaluación

Se realizó un test-bench en el cuál se verificó el correcto resultado para todas las posibles combinaciones de entradas utilizando ciclos 'for'. La implementación del test-bench se encuentra en el archivo 'test\_conv\_ca2\_sym.vhd'.

## 5. Sub-modulo operacion

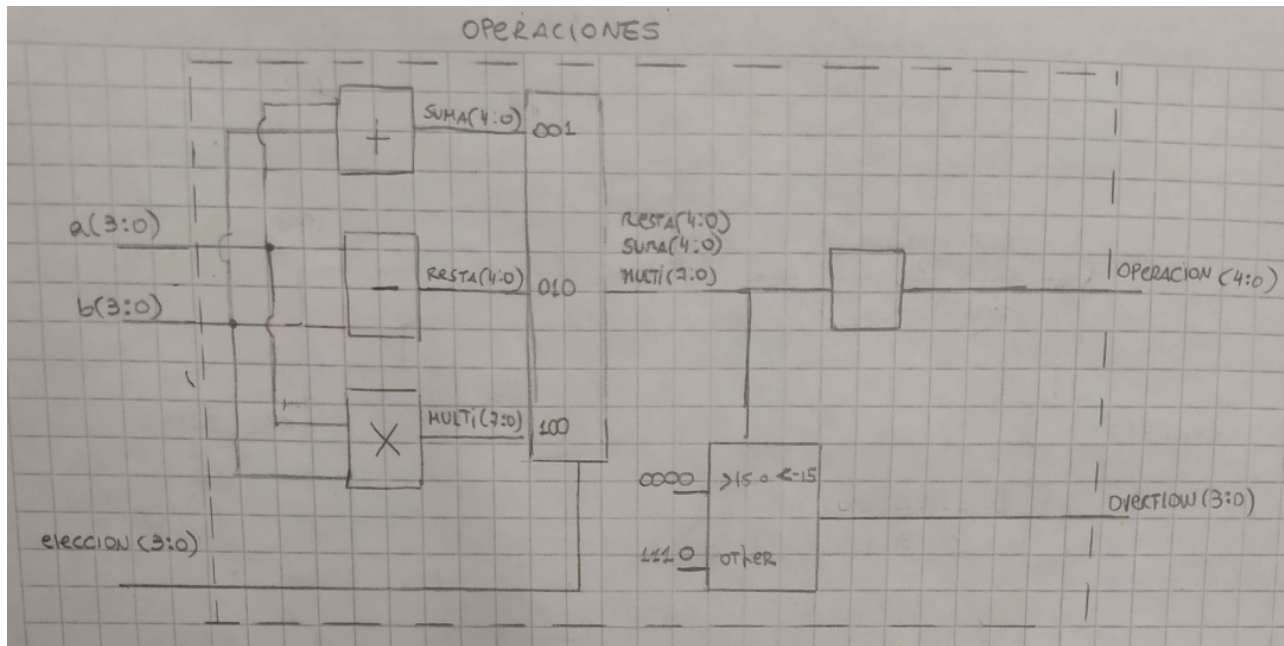


Figura 5: Diagrama de bloques de la lógica del sub-módulo 'operaciones'.

En la figura 5 se muestra esquemáticamente en diagrama de bloques la lógica implementada en el sub-módulo 'operaciones' de la calculadora. La operación a realizar está definida por la entrada 'elección', donde '001' indica la selección de la señal suma de 5 bits, '010' indica la señal resta de 5 bits y '100' indica la señal multiplicación de 8 bits. Las operaciones se realizan utilizando las operaciones nativas de VHDL. La señal seleccionada se utiliza para chequear si hay overflow, donde '0000' es la salida para overflow y '1110' la salida para no overflow. Por otro lado, la señal seleccionada se castea a un número de 5 bits y constituye la salida del 'operación' del submódulo.

### 5.1. Implementación en VHDL

La implementación en VHDL del submódulo operaciones se encuentra en el archivo 'operaciones.vhd'.

### 5.2. Simulación y evaluación

Se realizó un test-bench para el sub-módulo 'operaciones'. En primer lugar se fija el valor de la entrada elección como '001' y se verifican las salidas 'overflow' y 'operacion' para todas las posibles entradas utilizando loops for. Es importante destacar que la señal 'overflow' es más significativa y en caso que 'overflow' sea '0000' el valor de 'operación' es irrelevante. Luego se repite lo mismo para la entrada elección como '010' y '100'. Por último, se verifica que se produce señal de overflow para el resto de los valores posibles de 'eleccion'. La implementación del test-bench se encuentra en el

archivo 'test\_conv\_ca2\_sym.vhd'. Se verificaron los correctos resultados para todas las posibles combinaciones de entradas.

## 6. Sub-modulo sseg

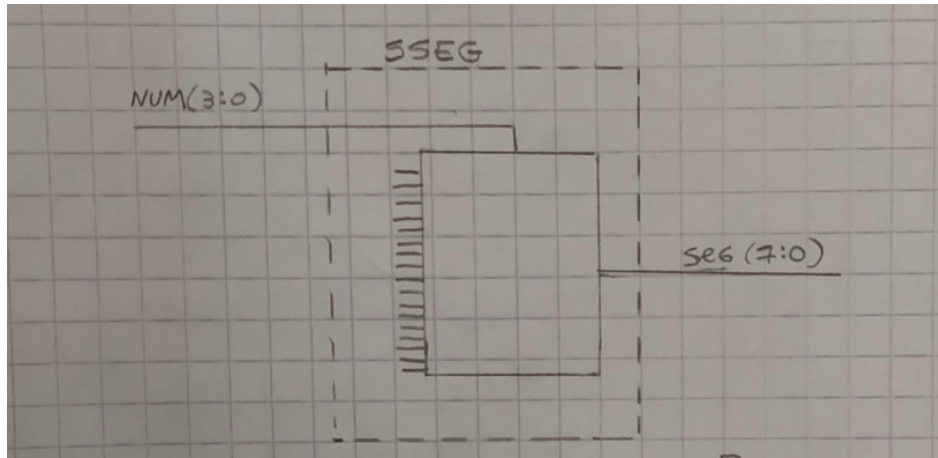


Figura 6: Diagrama de bloques de la lógica del sub-módulo 'sseg'.

En la figura 6 se muestra esquemáticamente en diagrama de bloques la lógica implementada en el sub-módulo 'sseg' de la calculadora. Como entrada se tiene un vector de 4 bits que representa la magnitud del resultado. Esta entrada ingresa a un multiplexor y elige la combinación adecuada de segmentos encendidos y apagados para representar tal magnitud. La combinación de estos segmentos es un vector de 8 bits y está contenido en la salida 'seg'.

### 6.1. Implementación en VHDL

La implementación en VHDL del submódulo operaciones se encuentra en el archivo 'conv\_compa2.vhd'.

### 6.2. Simulación y evaluación

Dada la simplicidad del sub-módulo no se realizó test-bench. La verificación de este sub-módulo está implícita en el test-bench del módulo top.

## 7. Integración de los módulos en VHDL

La implementación en VHDL del módulo 'top' se encuentra en el archivo 'top.vhd'. La asignación de las señales de entrada y salida a la placa Nexys se encuentra en el archivo 'Nexys3\_Master.ucf'.

## 8. Simulacion del módulo TOP

Se realizó un test-bench para el módulo top. En primer lugar se fija el valor de la entrada elección como '001' y se verifican las salidas 'an' para todas las combinaciones posibles de 'a' y 'b'. En caso que 'an' solo encienda un led (no hay falla), se verifica la salida del led (representa el signo) y el siete segmentos (representa la magnitud). Se repite lo mismo para la entrada elección en '010' y '100'. Por último, se verifica que se produce señal de overflow para el resto de los valores posibles de 'eleccion'. La implementación del test-bench se encuentra en el archivo 'test\_top.vhd'. Se verificaron los correctos resultados para todas las posibles combinaciones de entradas.

## 9. Conclusiones

Se implementó en una placa Nexys3 una calculadora para números de 4 bits en representación signo y magnitud para realizar operaciones de suma, resta y multiplicación. Se codificó en lenguaje VHDL utilizando compuertas lógicas y bloques combinacionales. Se realizaron los test-bench correspondientes para verificar el comportamiento del módulo.