Diseño de una Unidad aritmético lógica (ALU) de 4 bits con VHDL

Una ALU es un circuito digital que se utiliza para realizar operaciones aritméticas y lógicas. Es uno de los componentes esenciales en los microcontroladores. Es capaz de realizar las siguientes operaciones:

- Operaciones Aritméticas: suma, resta, multiplicación y división.
- Operaciones Lógicas: AND (Y), OR (O), NOT (NO) y XOR (O exclusivo).

El uso de FPGAs permite diseñar ALUs personalizadas que se ajustan a las necesidades específicas de una aplicación. Esto se hace utilizando lenguajes de descripción de hardware como VHDL. En la figura 1 se muestra la representación de la ALU de 4 bits que tiene como entrada dos parámetros (A y B) y como salida se obtiene el resultado (Result) de la operación que se le indicó realizar.

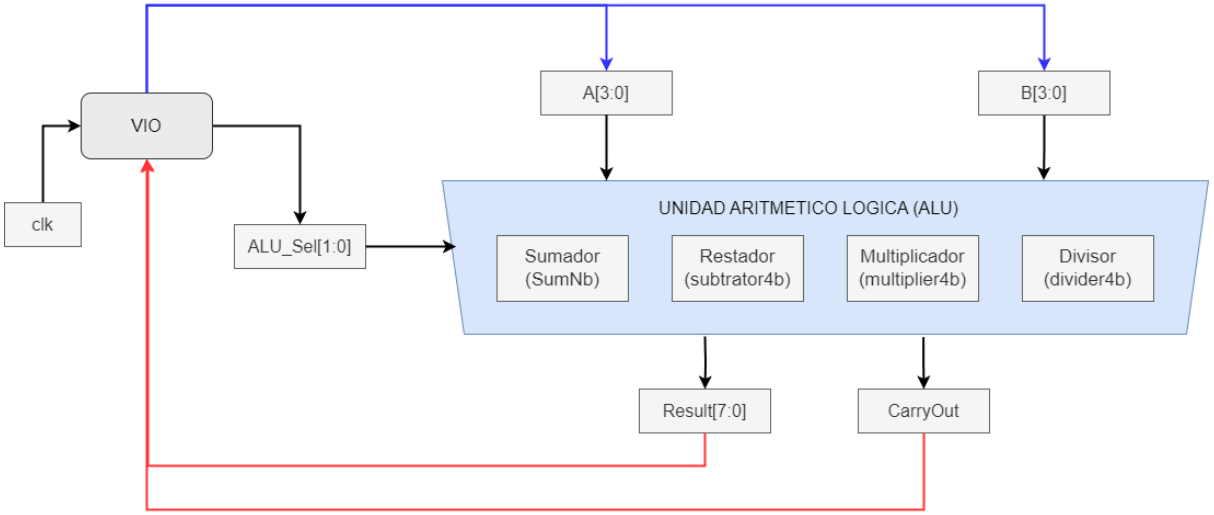

Figura 1. Diagrama de bloques ALU de 4 bits.

Para seleccionar qué operación aritmética se quiere realizar se utiliza un número de operación (ALU_Sel). Las operaciones que podemos realizar según el valor indicado son:

Tabla 1. Operaciones que realiza la ALU de 4 bits.

ALU_Sel	Result
0 0	A + B
0 1	A - B
1 0	A * B
11	A / B

La salida CarryOut se utiliza para la operación de la suma con el valor de 1 o 0 según sea el caso.

Componentes Principales de la ALU de 4 bits:

1. **Suma de 4 bits:**

Realiza la suma de dos números de 4 bits con acarreo.

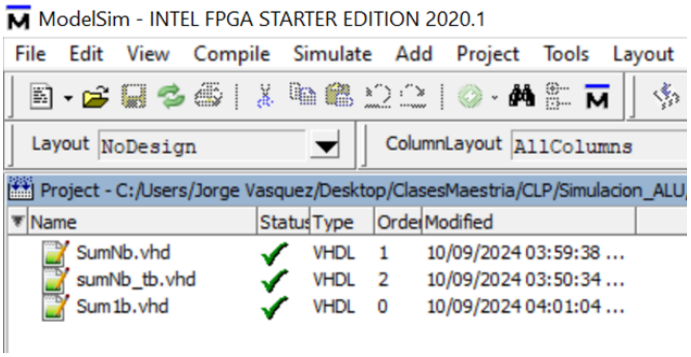

Figura 2. Test bench de la suma de 4 bits realizado con la herramienta modelsim.

SumNB es el sumador de N bits, utiliza múltiples instancias del sumador de 1 bits (sum1b) para sumar dos números, manejando los acarreos de manera adecuada entre los bits.

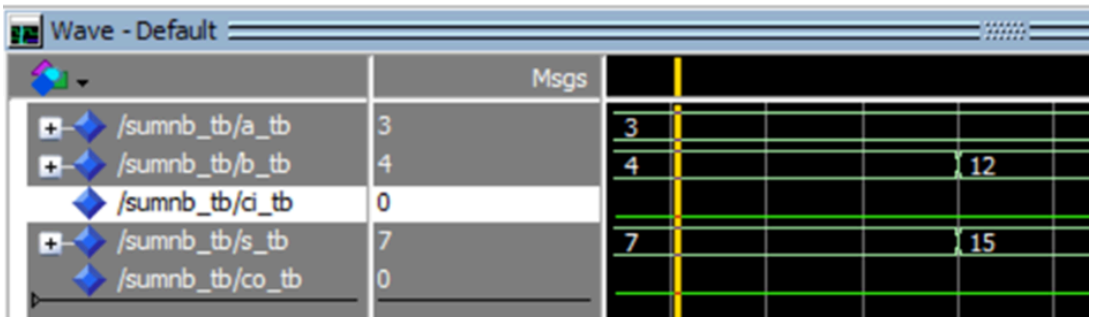

Figura 3. Simulación del componente SumNb.

2. **Resta de 4 bits:**

Realiza la resta entre dos números de 4 bits.

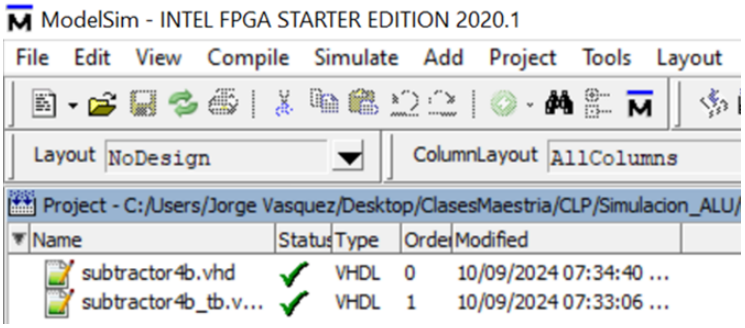

Figura 4. Test bench de la resta de 4 bits.

Subtractror4b es el restador de 4 bits, está diseñado para realizar la operación de resta entre dos números y gestionar el préstamo en caso de que el sustraendo (b) sea mayor que el minuendo (a). Utiliza la lógica de complemento a dos para llevar a cabo la resta. El código maneja esto extendiendo ambos números a 5 bits antes de realizar la resta, lo que permite detectar si hubo un préstamo.

Lógica para realizar la resta:

- Complemento a 2: Se calcula el complemento a 2 del sustraendo “b” para realizar la resta utilizando la suma. Esto implica invertir los bits de “b” (not b) y luego sumar 1 ("0001").
- Suma con Préstamo: Se realiza la suma del minuendo “a” y el complemento a 2 de “b”. El resultado es almacenado en una señal que tiene 5 bits para incluir el posible bit de préstamo.

Resultado y Préstamo:

- En diff se asigna a los 4 bits menos significativos del resultado de la resta.
- En borrow se asigna el bit más significativo del resultado de la resta, indicando si hubo un préstamo. Si hubo préstamo borrow vale 0.

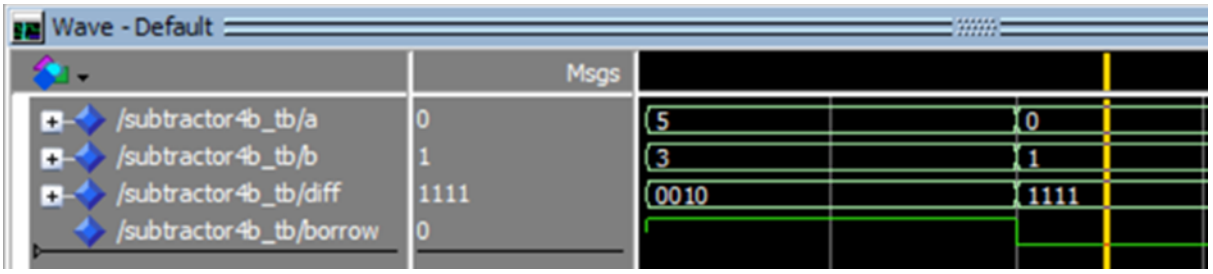

Figura 5. Simulación del componente subtractor4b.

3. **Multiplicación de 4 bits:**

Realiza el producto de dos números de 4 bits.

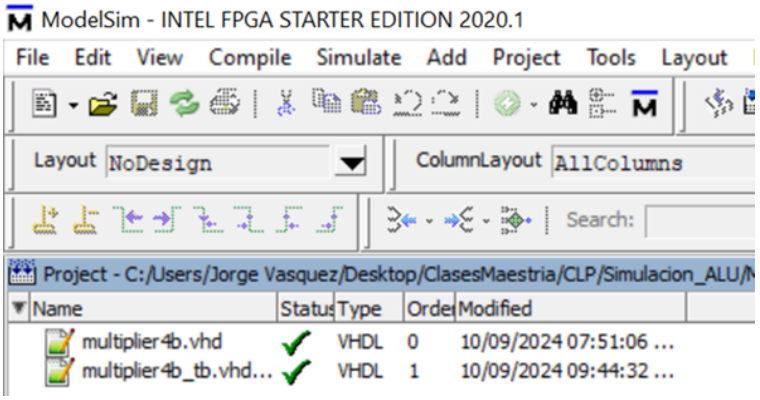

Figura 6. Test bench de la multiplicación de 4 bits.

La multiplicación binaria se realiza sumando productos parciales. Cada producto parcial se obtiene desplazando el multiplicando “a” a la izquierda por el número de posiciones correspondiente al bit actual del multiplicador “b”. Si el bit de “b” es 1, se suma el producto parcial al producto temporal. Este proceso se repite para cada bit de “b”.

Ejemplo:

Valores Iniciales

- a = 0011 (3 en binario)
- b = 0010 (2 en binario)

bit 0 de b es = 0	No se suma nada
bit 1 de b es = 1	Producto parcial: 0011 desplazado una posición: 0110 (6 en decimal). Producto temporal: 00000000 + 00000110 = 00000110 (6 en decimal)
bit 2 de b es = 0	No se suma nada
bit 3 de b es = 0	No se suma nada

Resultado Final

El resultado de la multiplicación es 00000110, que es 6 en decimal.

Otro ejemplo:

Valores Iniciales

- a = 0111 (7 en binario)
- b = 0101 (5 en binario)

bit 0 de b es = 1	Producto parcial: 0111 (7 en decimal) Producto temporal: 00000000 + 00000111 = 00000111 (7 en decimal)
bit 1 de b es = 0	No se suma nada
bit 2 de b es = 1	Producto parcial: 0111 desplazado 2 posiciones: 011100 (28 en decimal) Producto temporal: 00000111 + 00011100 = 00100011 (35 en decimal)
bit 3 de b es = 0	No se suma nada

Resultado Final

- El resultado de la multiplicación es 00100011, que es 35 en decimal.

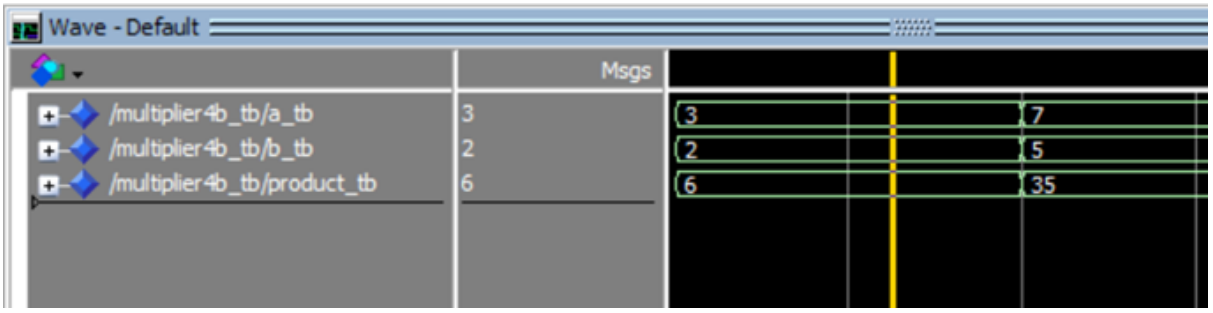

Figura 7. Simulación del componente multiplier4b.

4. **División de 4 bits:**

Realiza la división entre dos números de 4 bits y produce cociente y resto.

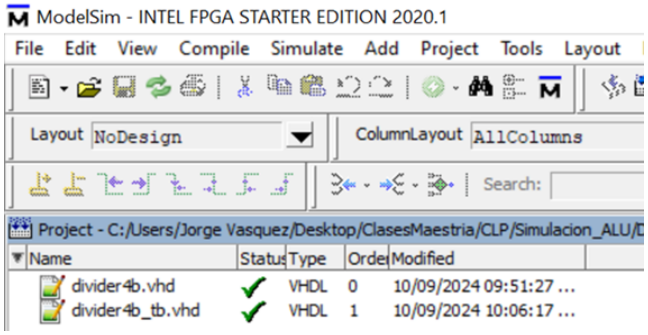

Figura 8. Test bench de la división de 4 bits.

La división binaria se realiza utilizando las operaciones aritméticas estándar de división y resto. El código maneja la división por cero estableciendo el cociente en un valor de error (1111) y el resto en el valor del dividendo.

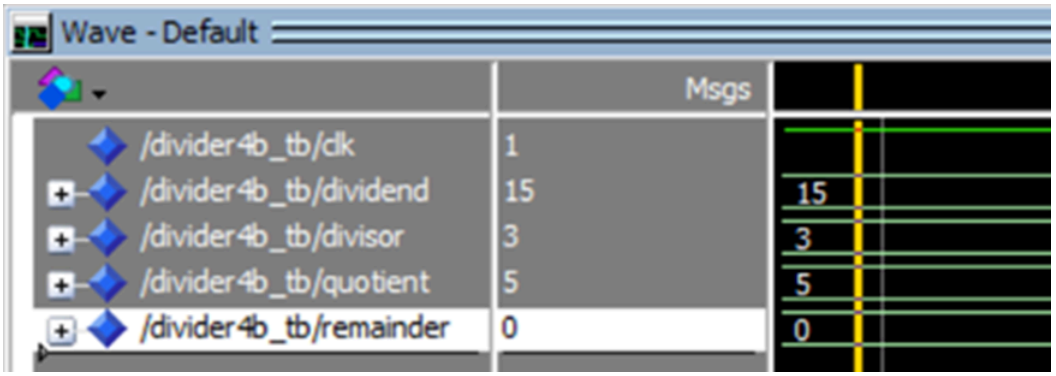

Figura 9. Simulación del componente divider4b.

Bloque principal de la ALU de 4 bits:

A continuación se muestra el test bench del proyecto completo.

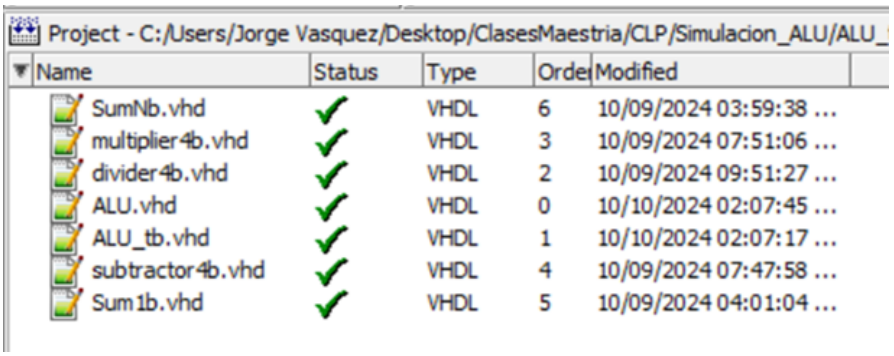

Figura 10. Test bench de la ALU de 4 bits.

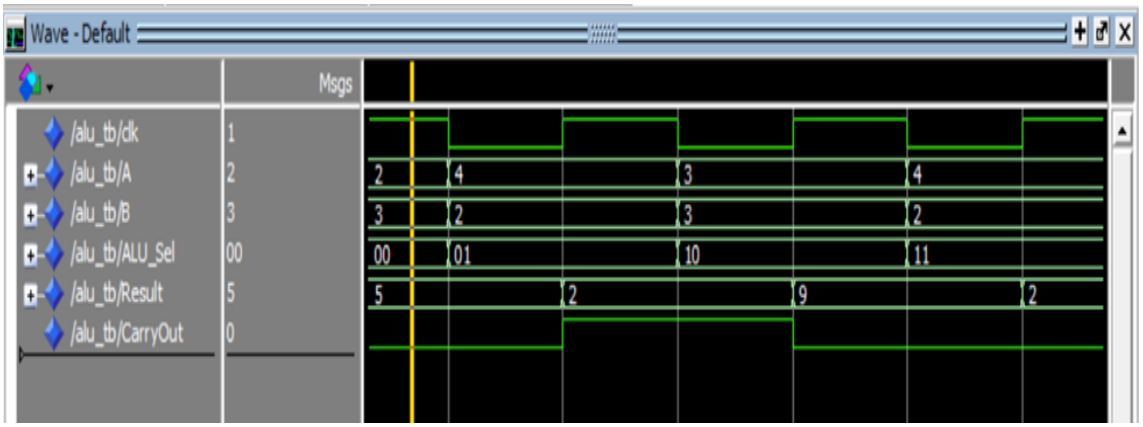

Figura 11. Simulación de todo el proyecto.

Síntesis e implementación de la ALU de 4 bits:

En las siguientes figuras se muestra las tablas de uso de recursos, esquemático y ruteo de las etapas de síntesis e implementación del proyecto en el software vivado.

Utilization — **Post-Synthesis** | Post-Implementation

Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	57	17600	0.32
FF	34	35200	0.10
IO	10	100	10.00
BUFG	1	32	3.13

Figura 12. Tabla de uso de recursos post-synthesis.

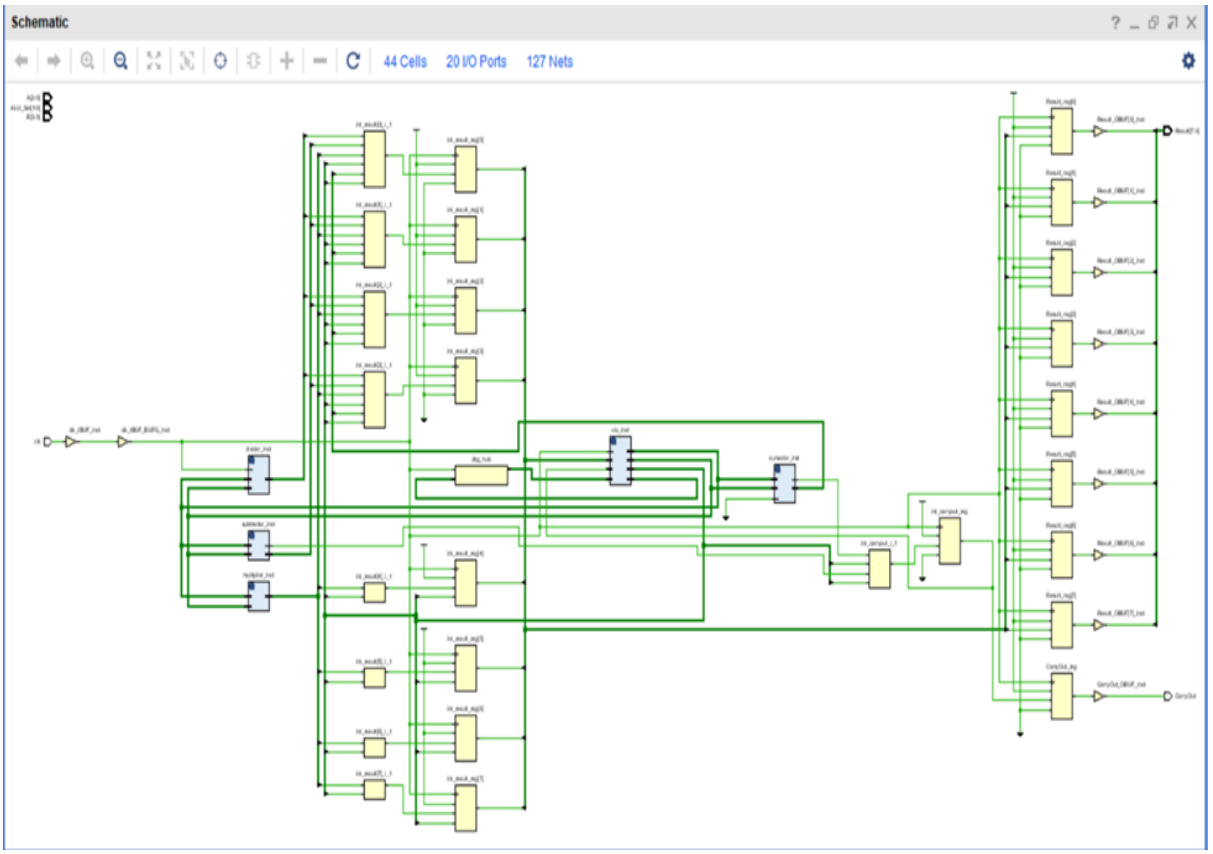

Figura 13. Sistema sintetizado.

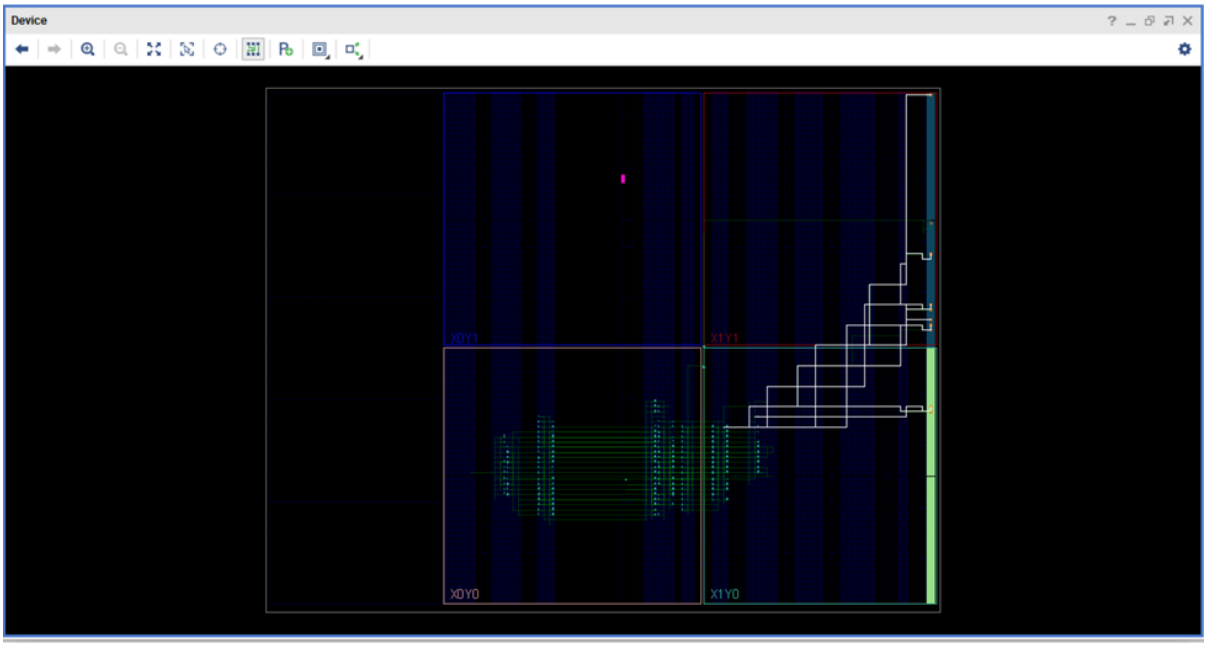

Figura 14. Vista del ruteo realizado.

Utilization — Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	658	17600	3.74
LUTRAM	24	6000	0.40
FF	1077	35200	3.06
IO	10	100	10.00
BUFG	2	32	6.25

Figura 15. Tabla de uso de recursos post-implementation.

Resultado final del proyecto:

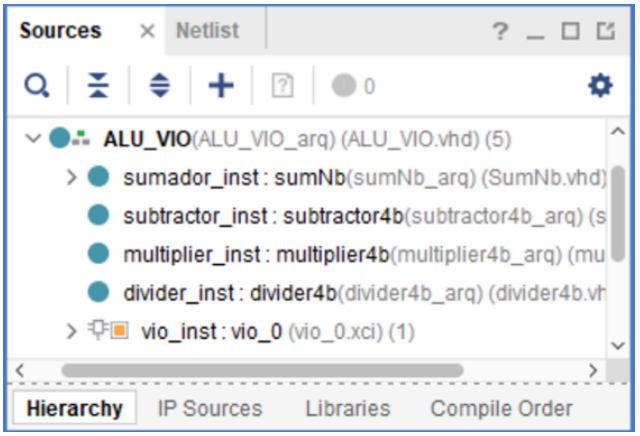

Figura 16. Jerarquía de archivos del proyecto ALU.

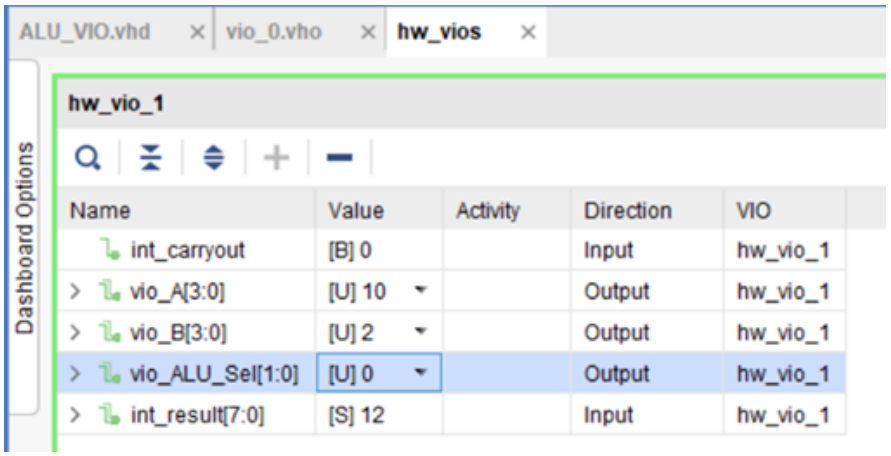

Figura 17. Se realiza la suma de dos números.

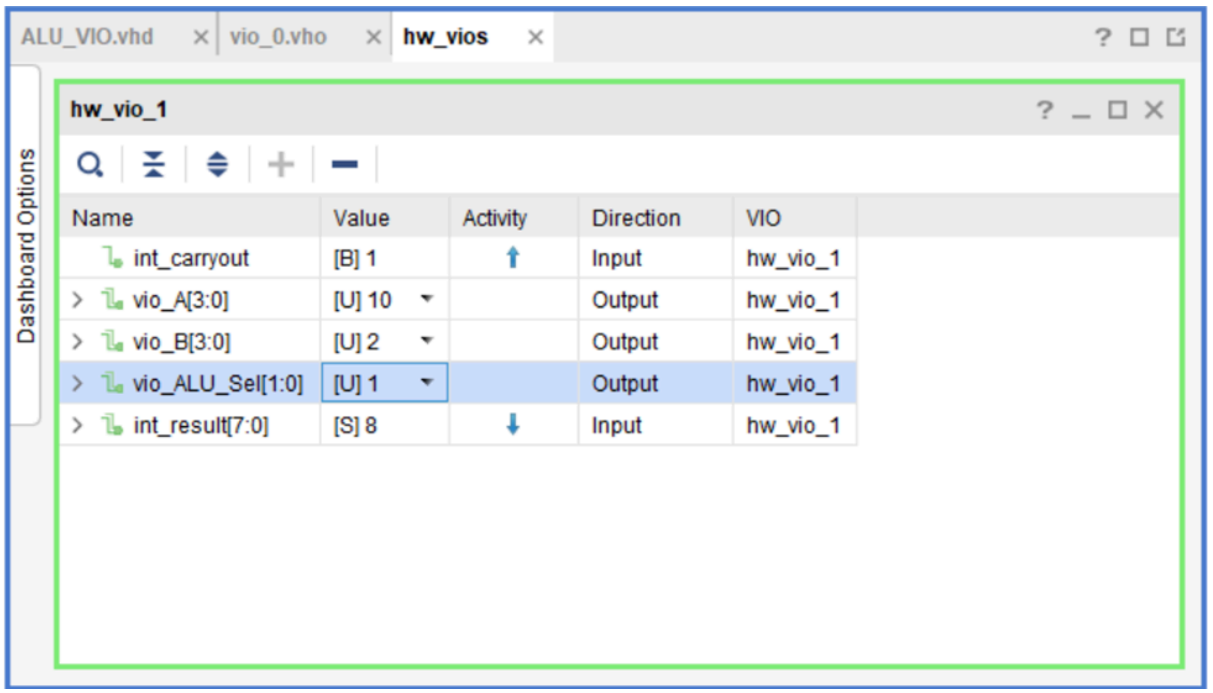

Figura 18. Se realiza la resta de dos números.

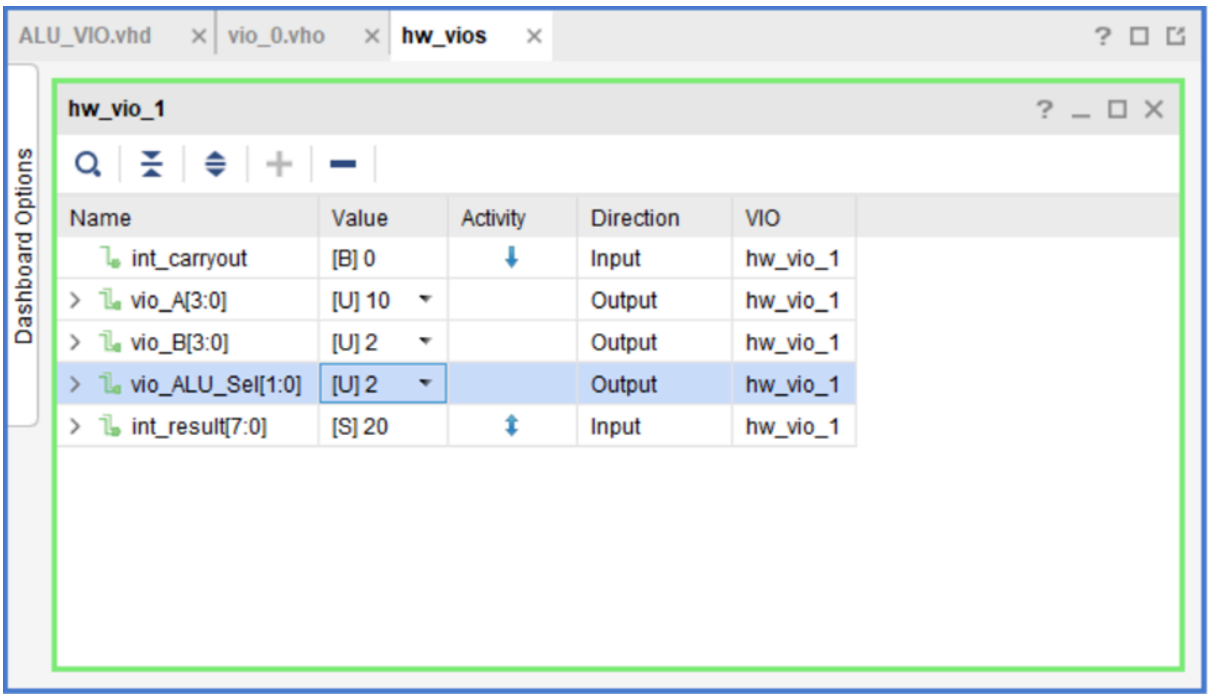

Figura 19. Se realiza la multiplicación de dos números.

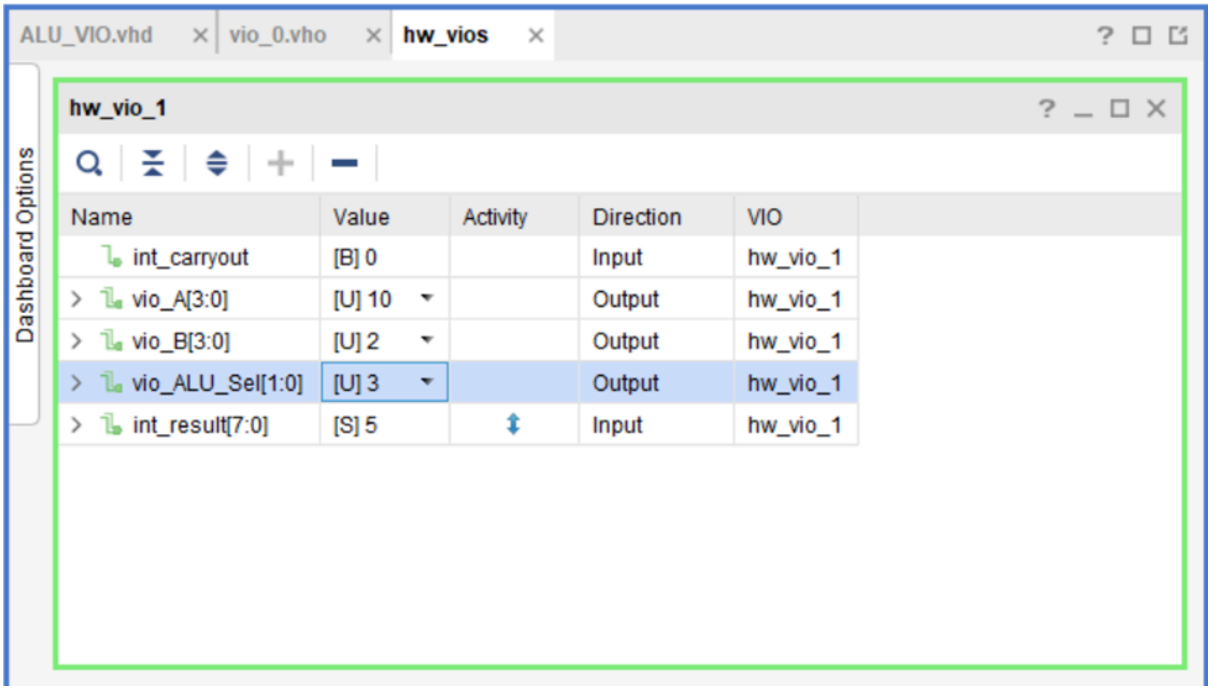

Figura 20. Se realiza la división de dos números.