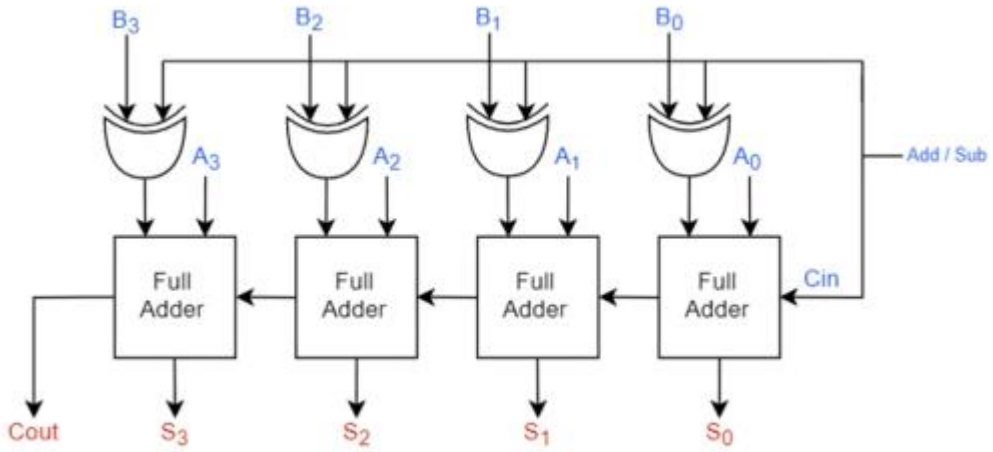
SUMADOR/RESTADOR Y MULTIPLICADOR

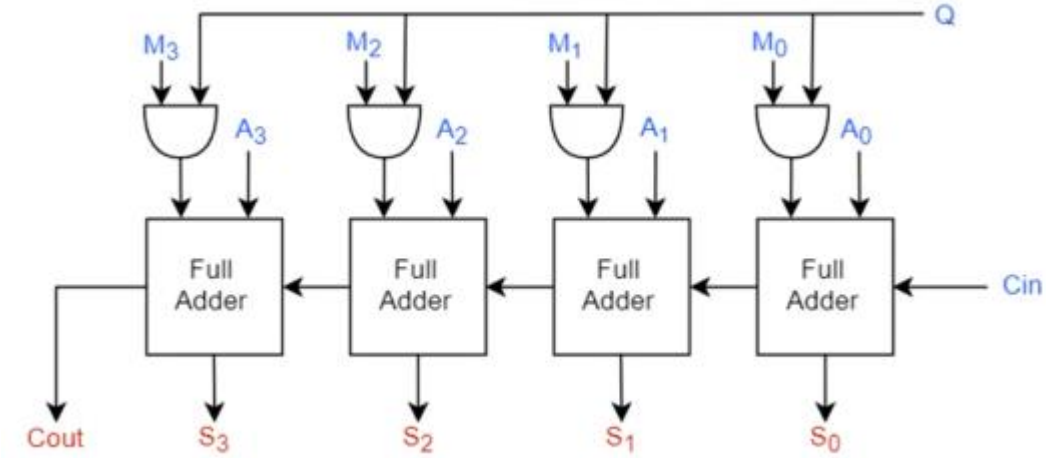
Objetivo

Implementar en VHDL un sumador restador y un multiplicador, ambos de 4 bits, instanciando un sumador de 4 bit, con las siguientes configuraciones:

Sumador restador de 4 bits

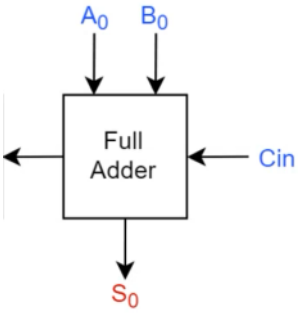


Multiplicador

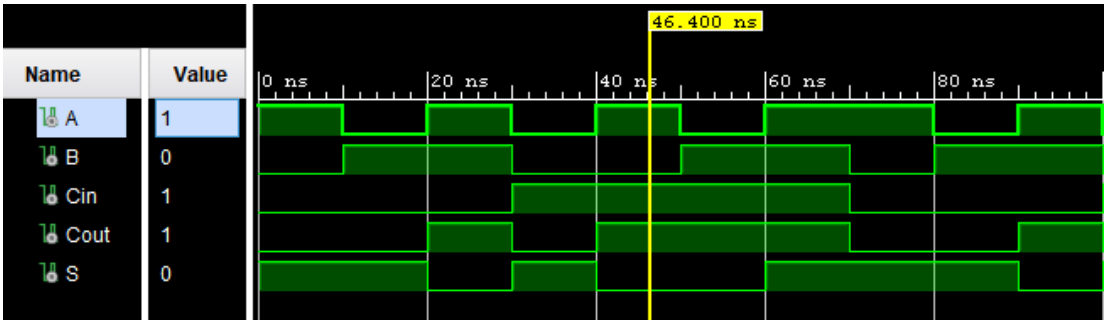


Procedimiento

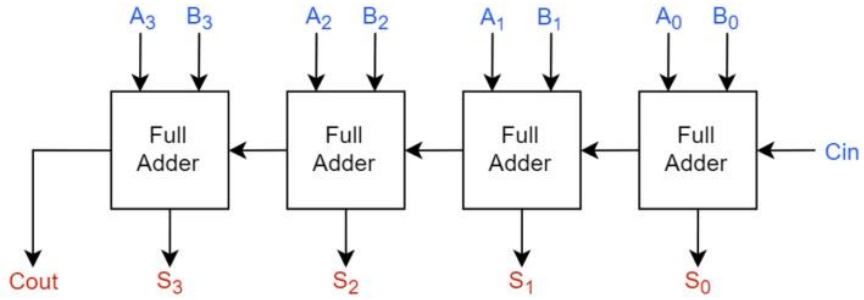
Para la implementación del sumador y restador, así como el multiplicador se realizó la implementación de bloques de menor nivel jerárquico para posteriormente poder instanciarlos. Primeramente, se implementó un sumador completo binario

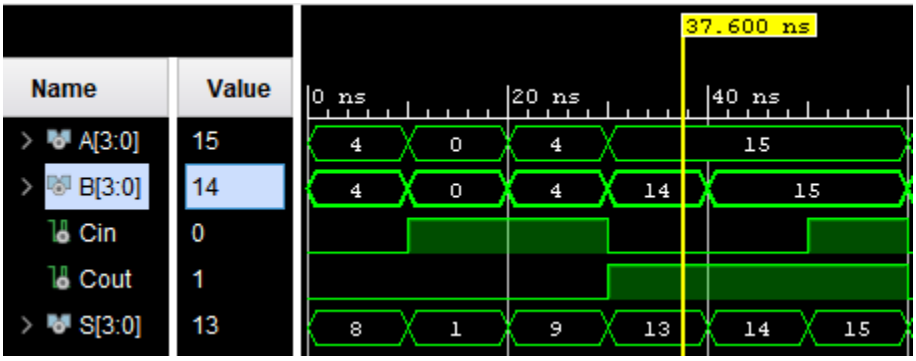



```
S <= A XOR B XOR Cin ;
Cout <= (A AND B) OR (Cin AND A) OR (Cin AND B) ;
```



Después se implementó un sumador de 4 bits



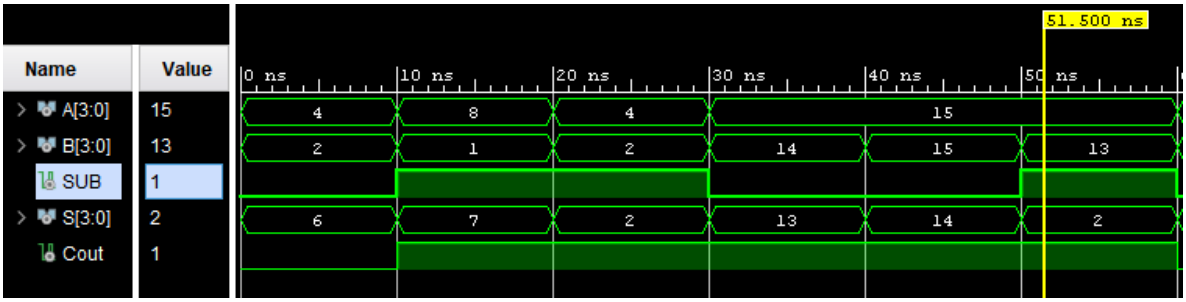



```
CA1: adder
    PORT MAP(
        A => A(0),
        B => B(0),
        Cin => Cin,
        Cout => c1,
        S => S(0)
    );
CA2: adder
    PORT MAP(
        A => A(1),
        B => B(1),
        Cin => c1,
        Cout => c2,
        S => S(1)
    );
CA3: adder
    PORT MAP(
        A => A(2),
        B => B(2),
        Cin => C2,
        Cout => c3,
        S => S(2)
    );
CA4: adder
    PORT MAP(
        A => A(3),
        B => B(3),
        Cin => C3,
        Cout => Cout,
        S => S(3)
    );
```

Con esta implementación ya se contaba con la base para poder crear el sumador-restador y el multiplicador únicamente realizando una operación XOR en el caso del sumador-restador y una AND para el multiplicador.

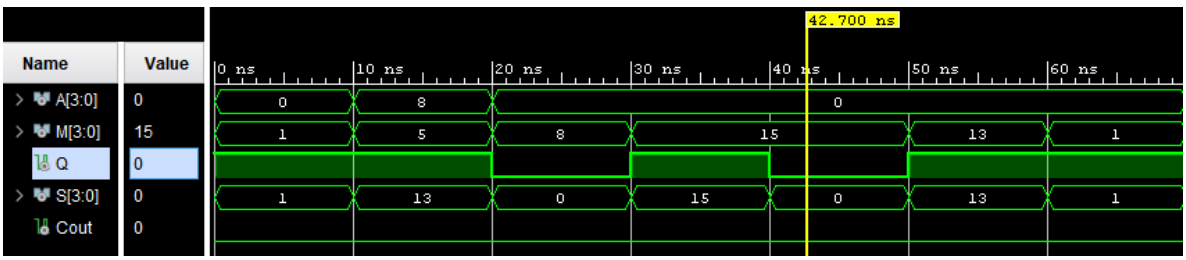
Sumador/Restador

```
B_A <= (SUB,SUB,SUB,SUB) XOR B;
--Instanciar componente
CFA1: four_bit_adder
    PORT MAP(
        A => A,
        B => (B_A),
        Cin => SUB,
        Cout => Cout,
        S => S
    );
```



Multiplicador

```
M_A <= (Q,Q,Q,Q) AND M;
--Instanciar componente
CFA1: four_bit_adder
    PORT MAP(
        A => A,
        B => (M_A),
        Cin => '0',
        Cout => Cout,
        S => S
    );
```



Se puede observar el buen funcionamiento de el sumador/restador y el multiplicador, así como sus dos subsistemas.