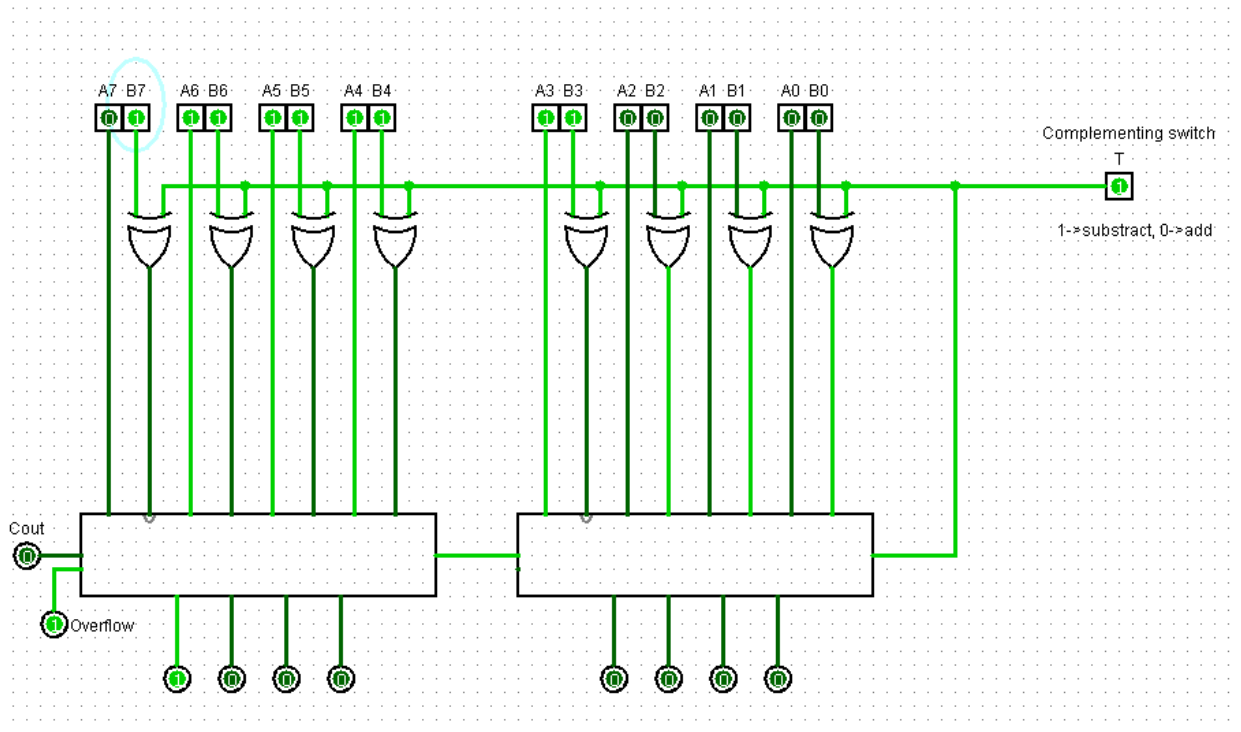
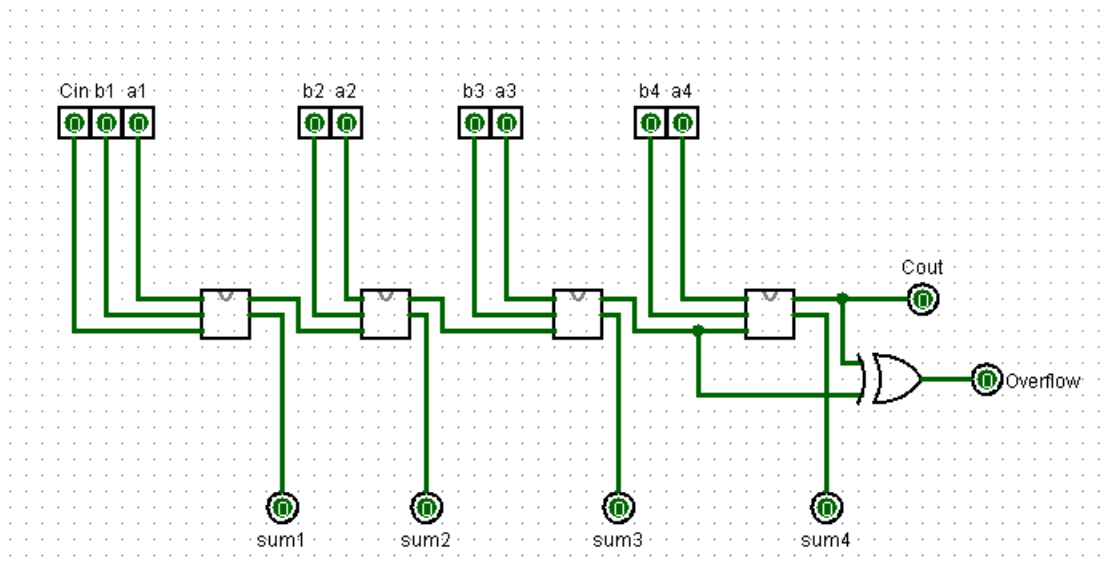


Esquemático

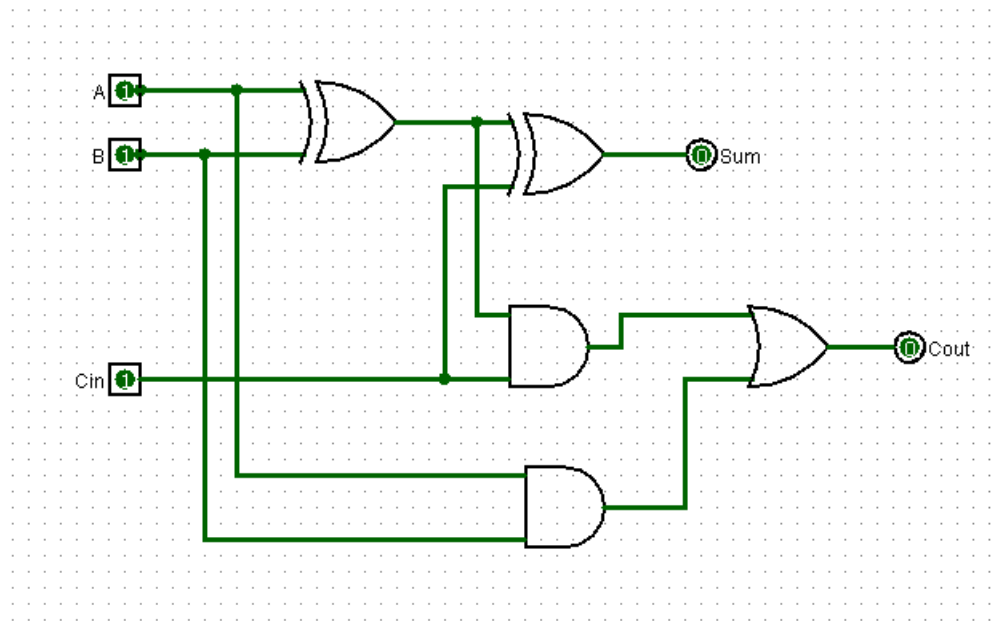
Eight-bit full adder: Conformado por dos Full-Adder de 4 bits.



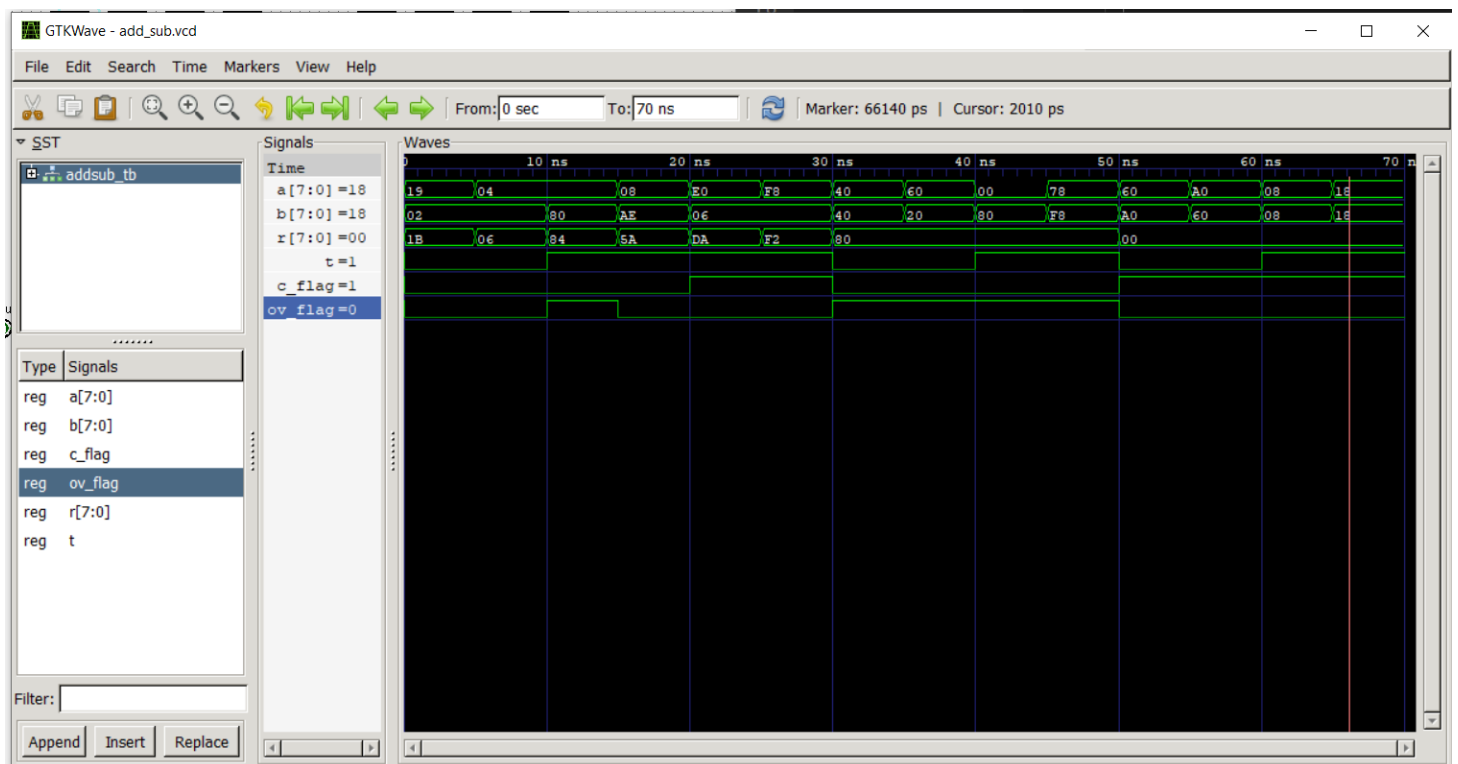
Four-bit full adder: Hecho con 4 Full adders



Full adder:



Simulación:



Eightbit_Adder_Subtractor:

Casos:

```
-- Put initialisation code here
--2 casos de suma de números positivos
A <= "00011001";
B <= "00000010";
T <= '0';
wait for 5 ns;
A <= "00000100";
B <= "00000010";
T <= '0';
wait for 5 ns;
--2 casos de suma de número positivo y número negativo
A <= "00000100";
B <= "10000000";
T <= '1';
wait for 5 ns;
A <= "00001000";
B <= "10101110";
T <= '1';
-- 2 casos de suma de número negativo y número positivo
wait for 5 ns;
A <= "11100000";
B <= "00000110";
T <= '1';
wait for 5 ns;
A <= "11111000";
B <= "00000110";
T <= '1';
```

```
-- 2 casos de overflow suma
wait for 5 ns;
A <= "01000000";
B <= "01000000";
T <= '0';
wait for 5 ns;
A <= "01100000";
B <= "00100000";
T <= '0';
-- 2 casos de overflow resta
wait for 5 ns;
A <= "00000000";
B <= "10000000";
T <= '1';
wait for 5 ns;
A <= "01111000";
B <= "11111000";
T <= '1';
```

```

-- 2 casos con carry out suma
wait for 5 ns;
A <= "01100000";
B <= "10100000";
T <= '0';
wait for 5 ns;
A <= "10100000";
B <= "01100000";
T <= '0';

-- 2 casos con carry out resta
wait for 5 ns;
A <= "00001000";
B <= "00001000";
T <= '1';
wait for 5 ns;
A <= "00011000";
B <= "00011000";
T <= '1';
wait for 5 ns;

-- Put test bench stimulus code here

wait;
end process;

end;
```