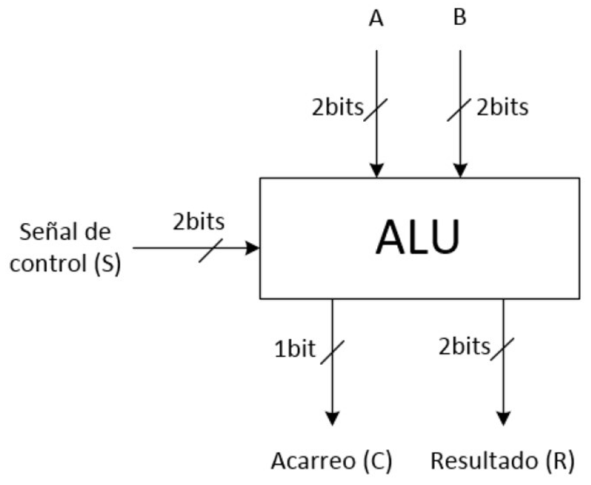
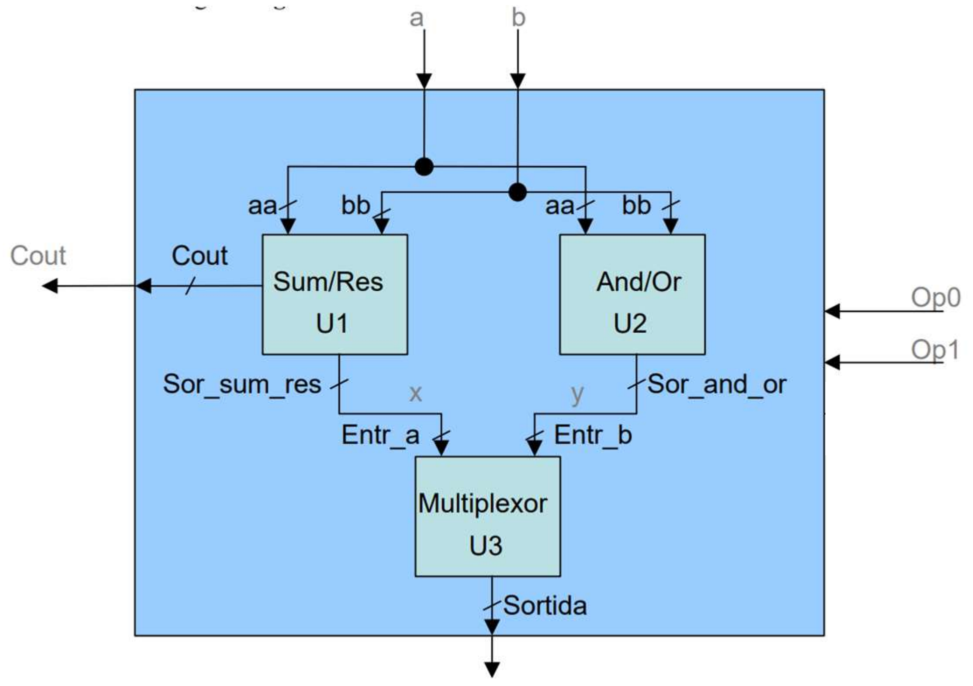
Diseño VHDL
Unidad Aritmético Lógica (ALU)

Práctica 4

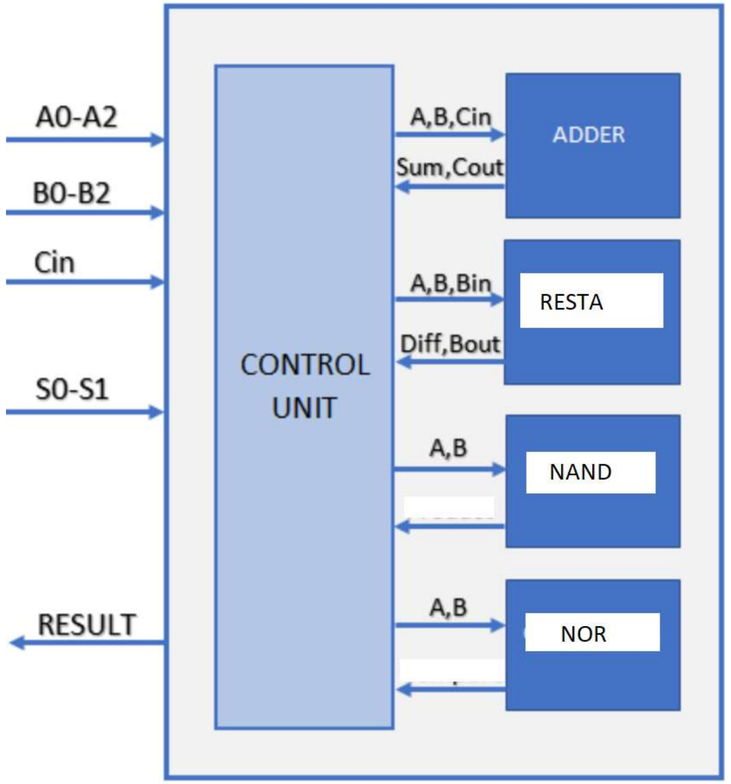
Definición de la ALU

Bits de control de la ALU	Función a realizar
00	Suma
01	Resta
10	NAND
11	NOR





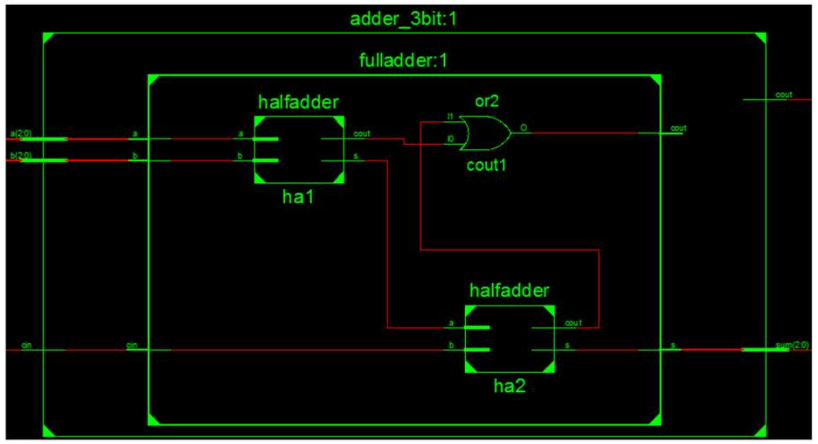
Ejemplo ALU: operación suma

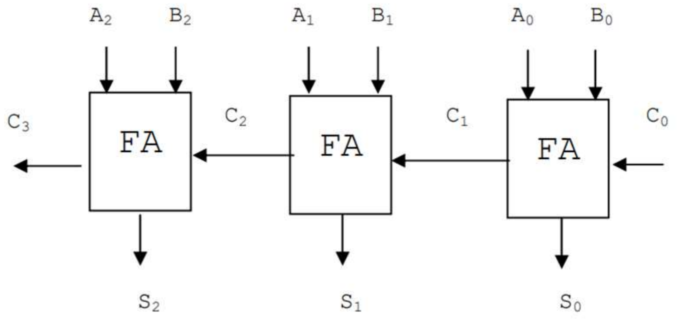


```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ALU is
Port ( a : in STD_LOGIC_VECTOR (2 downto 0);
b : in STD_LOGIC_VECTOR (2 downto 0);
c: in std_logic;
sel : in STD_LOGIC_VECTOR (1 downto 0);
result : out STD_LOGIC_VECTOR (5 downto 0));
end ALU;

architecture Behavioral of ALU is

component adder_3bit is
Port ( a : in STD_LOGIC_VECTOR (2 downto 0);
b : in STD_LOGIC_VECTOR (2 downto 0);
cin : in STD_LOGIC;
sum : out STD_LOGIC_VECTOR (2 downto 0);
cout : out STD_LOGIC);
end component;

signal t_adder:std_logic_vector(5 downto 0):= (others => '0');
begin

alu1: adder_3bit port map(a=>a,b=>b,cin=>c,sum=>t_adder(2 downto 0),cout=>t_adder(3));
process(sel,t_adder)
begin
case sel is
when "00"=>
result<=t_adder;
when others =>result<="000000";
end case;
end process;

end Behavioral;
```

https://allaboutfpga.com/tutorial-3-alu-structural-modelling-fpga-implementation/

Módulo SUMA: adder_3bits

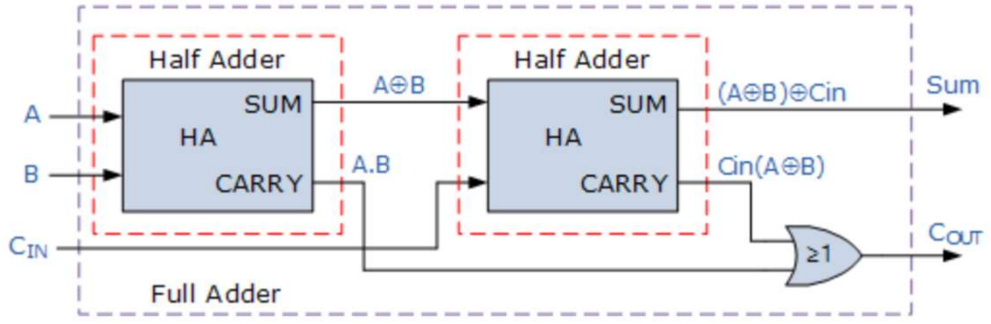




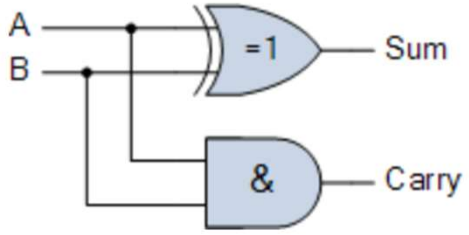
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity adder_3bit is
Port ( a : in STD_LOGIC_VECTOR (2 downto 0);
b : in STD_LOGIC_VECTOR (2 downto 0);
cin : in STD_LOGIC;
sum : out STD_LOGIC_VECTOR (2 downto 0);
cout : out STD_LOGIC);
end adder_3bit;

architecture Behavioral of adder_3bit is

component fulladder
port( a : in STD_LOGIC;
b : in STD_LOGIC;
cin : in STD_LOGIC;
s : out STD_LOGIC;
cout : out STD_LOGIC);
end component;

signal c1,c2:std_logic;

begin

fa1:fulladder port map(a(0),b(0),cin,sum(0),c1);
fa2:fulladder port map(a(1),b(1),c1,sum(1),c2);
fa3:fulladder port map(a(2),b(2),c2,sum(2),cout);

end Behavioral;
```

FullAdder && Halfadder

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity fulladder is
port( a : in STD_LOGIC;
b : in STD_LOGIC;
cin : in STD_LOGIC;
s : out STD_LOGIC;
cout : out STD_LOGIC);
end fulladder;

architecture Behavioral of fulladder is

component halfadder
Port( a,b : in std_logic;
s : out STD_LOGIC;
cout : out STD_LOGIC);
end component;

signal s1,c1,c2:std_logic;

begin

ha1: halfadder port map(a,b,s1,c1);
ha2: halfadder port map(s1,cin,s,c2);
cout <= c1 or c2;

end Behavioral;
```



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity halfadder is
Port( a,b : in std_logic;
s : out STD_LOGIC;
cout : out STD_LOGIC);
end halfadder;

architecture Behavioral of halfadder is

begin

s <= a xor b;
cout <= a and b;

end Behavioral;
```



A
B
=1
&
Sum
Carry