Trabalho Prático 3

24 de março de 2021 09:13

Exercise 1

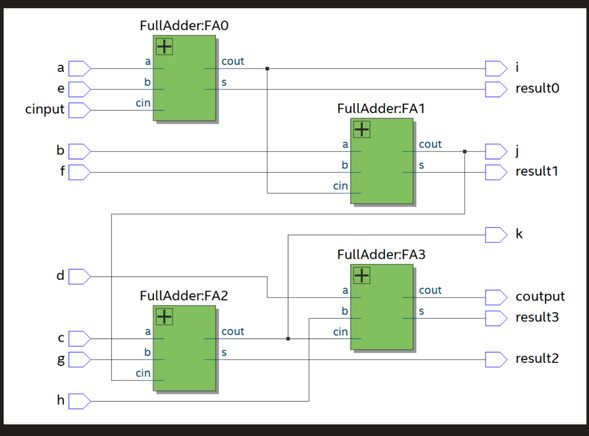
- s <= a xor b xor cin;
- cout <= (a and b) or (cin and(a or b));

Exercise 2

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity FullAdder is
      port(a, b, cin:  in
std_logic;
           s, cout : out
std_logic);
end FullAdder;
architecture Behavioral of
FullAdder is
begin
      s <= a xor b xor cin;
      cout  <= (a and b) or (cin
and(a or b));
end Behavioral;
```

Exercise 3

Exercise 4



Exercise 5

Serão necessários 3 sinais de Carry (buffers i, j e k) para interligar os 4 full Adders de 1 bit, além disso existem ainda uma entrada Carry in (transporte de entrada) e uma saída Carry Out (transporte de saída).

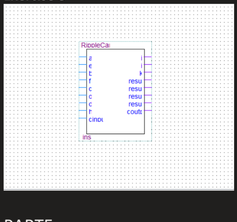
Exercise 6

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity RippleCarry is
   port(a, e, b, f, c, g, d, h, cinput:   in std_logic;
        i, j, k: buffer std_logic;
        result0, result1, result2, result3, coutput : out std_logic);
end RippleCarry;

architecture Structural of RippleCarry is
begin
   FA0: entity work.FullAdder
        port map (a => a,
                  b => e,
                  cin=> cinput,
                  s => result0,
                  cout => i );
   FA1: entity work.FullAdder
        port map (a => b,
                  b => f,
                  cin=> i,
                  s => result1,
                  cout => j );
   FA2: entity work.FullAdder
        port map (a => c,
                  b => g,
                  cin=> j,
                  s => result2,
                  cout => k );
   FA3: entity work.FullAdder
        port map (a => d,
                  b => h,
                  cin=> k,
                  s => result3,
                  cout => coutput );

end Structural;
```

Exercise 8



PARTE II

Exercise 1

- Adição - 1100
- Subtração - 0110
- Multiplicação - 11011
- Divisão - 0011
- AND - 0001
- OR - 1011
- XOR - 1010

Exercise 4