

Projeto Sistemas Digitais: Circuito contador de 1 (Uns)

- A partir das entradas A, B e C indicar nas saídas S1 e S2 a quantidade de uns nas entradas correspondentes. Desenhe tabela verdade, equações simplificadas, circuito em portas lógicas, simulação nos Quartus e gravação na FPGA.

- Tabela verdade

A	B	C	S1	S2
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- Equações

$$S1 = \underline{A'BC} + AB'C + ABC' + \underline{ABC}$$

$$S1 = BC + AB'C + ABC'$$

$$S1 = BC + A(B \text{ xor } C)$$

$$S2 = A'BC' + A'B'C + AB'C' + ABC$$

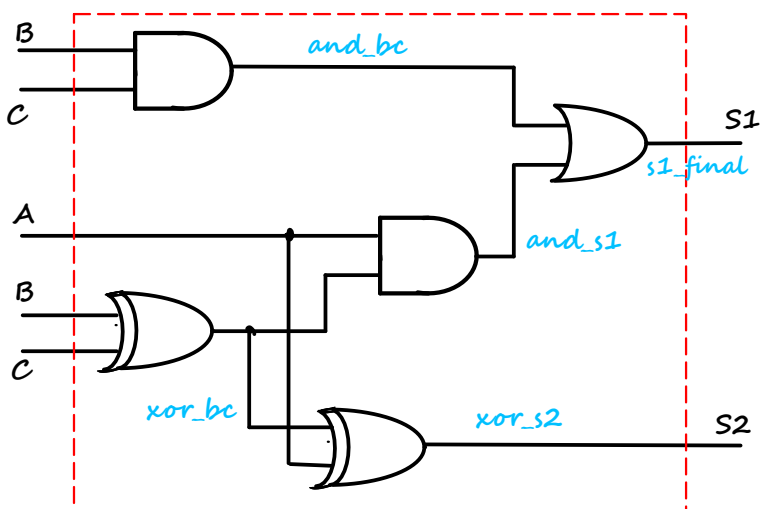
Uma observação dor Xor (ou exclusivo):

O valor do ou exclusivo é 1 quando a quantidade de uns é ímpar. Como é o caso de S2.

Dessa forma:

$$S2 = A \text{ xor } B \text{ xor } C$$

- Circuitos em portas lógicas:



■ Algumas combinações de sinais para mapeamento na placa:

Códigos das portas lógicas em vhdl:

```
ENTITY PortaAnd IS
  PORT(en1, en2 : IN BIT;
        saida_and : OUT BIT);
END PortaAnd;
```

```
ARCHITECTURE behav OF PortaAnd IS
BEGIN
  saida_and <= en1 AND en2;
END ARCHITECTURE behav;
```

```
ENTITY PortaOr IS
  PORT(input1, input2 : IN BIT;
        saida_or : OUT BIT);
END PortaOr;
```

```
ARCHITECTURE behav OF PortaOr IS
BEGIN
  saida_or <= input1 OR input2;
END ARCHITECTURE behav;
```

ENTITY PortaXor IS

PORT(entXor1, entXor2 : IN BIT;
saida_xor : OUT BIT);

END PortaXor;

ARCHITECTURE behav OF PortaXor IS

BEGIN

saida_xor <= entXor1 XOR entXor2;

END ARCHITECTURE behav;

Circuito Contador de Uns

entity bitsContador is

port(

A,B,C: in bit;

S1, S2: out bit

);

end bitsContador;

architecture behav of bitsContador is

signal and_bc: bit;

signal xor_bc: bit;

signal and_s1: bit;

signal xor_s2: bit;

signal s1_final: bit;

component PortaAnd is

port(en1,en2 : in bit;

saida_and : out bit

);

end component;

component PortaOr is

port(input1,input2 : in bit;

saida_or : out bit

);

end component;

component PortaXor is

port(entXor1,entXor2 : in bit;

saida_xor : out bit

);

end component;

begin

u1: PortaAnd port map(en1 => B, en2 => C, saida_and => and_bc);

u2: PortaXor port map(entXor1 => B, entXor2 => C, saida_xor => xor_bc);

u3: PortaAnd port map(en1 => A, en2 => xor_bc, saida_and => and_s1);

u4: PortaOr port map(input1 => and_bc, input2 => and_s1, saida_or => s1_final);

u5: PortaXor port map(entXor1 => A, entXor2 => xor_bc, saida_xor => xor_s2);

S1 <= s1_final;

S2 <= xor_s2;

end architecture behav;