



**Universidade Federal de Santa Catarina**  
**Centro Tecnológico – CTC**  
**Departamento de Engenharia Elétrica**



# **“EEL5105 - Circuitos e Técnicas Digitais”**

**Prof. Héctor Pettenghi Roldán\***

[Hector@eel.ufsc.br](mailto:Hector@eel.ufsc.br)

**Florianópolis, março de 2016.**

**\*Baseados nos slides do Professor Eduardo Bezerra EEL5105 2015.2**

# Sistemas Digitais

***Projeto hierárquico***

# Objetivos do laboratório

---

1. Entender o conceito de “projeto hierárquico”.
2. Implementação de projeto hierárquico em VHDL com o uso de “**component**” e “**port map**”.
3. Uso de ferramentas de desenvolvimento para o projeto hierárquico de sistemas digitais em VHDL.
4. Implementação de estudo de caso em VHDL, incluindo a simulação com diagramas de formas de ondas, e prototipação no FPGA do kit de desenvolvimento.

# Projeto hierárquico

---

Ex. projetar um circuito para implementar a seguinte equação:

$$F = (F1 \text{ and } F2) \text{ or } F3$$

onde:

$$F1 = A \text{ or } B \text{ or } C$$

$$F2 = B \text{ xor } C$$

$$F3 = \text{“a ser definido”}$$

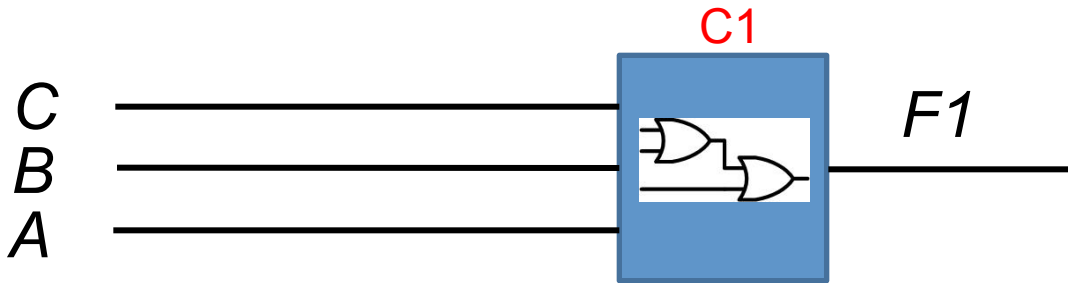
O circuito possui um componente (C4) para o cálculo de F, e três componentes (C1, C2 e C3) para calcular F1, F2 e F3.

O circuito final consiste de um componente que recebe A, B e C como entradas, e fornece F como saída.

# Projeto hierárquico

---

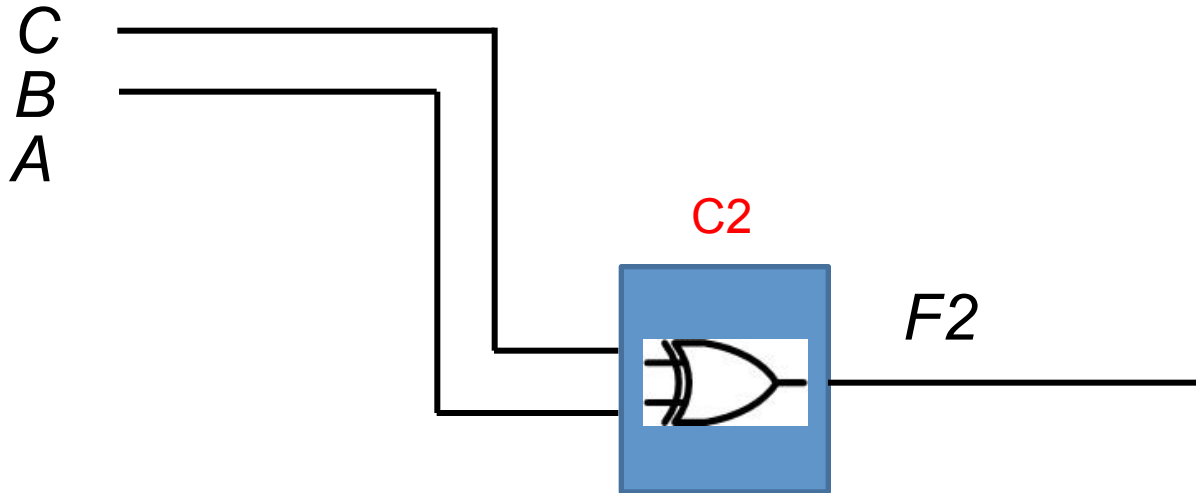
Componente **C1**: implementa  $F1 = A \text{ or } B \text{ or } C$



# Projeto hierárquico

---

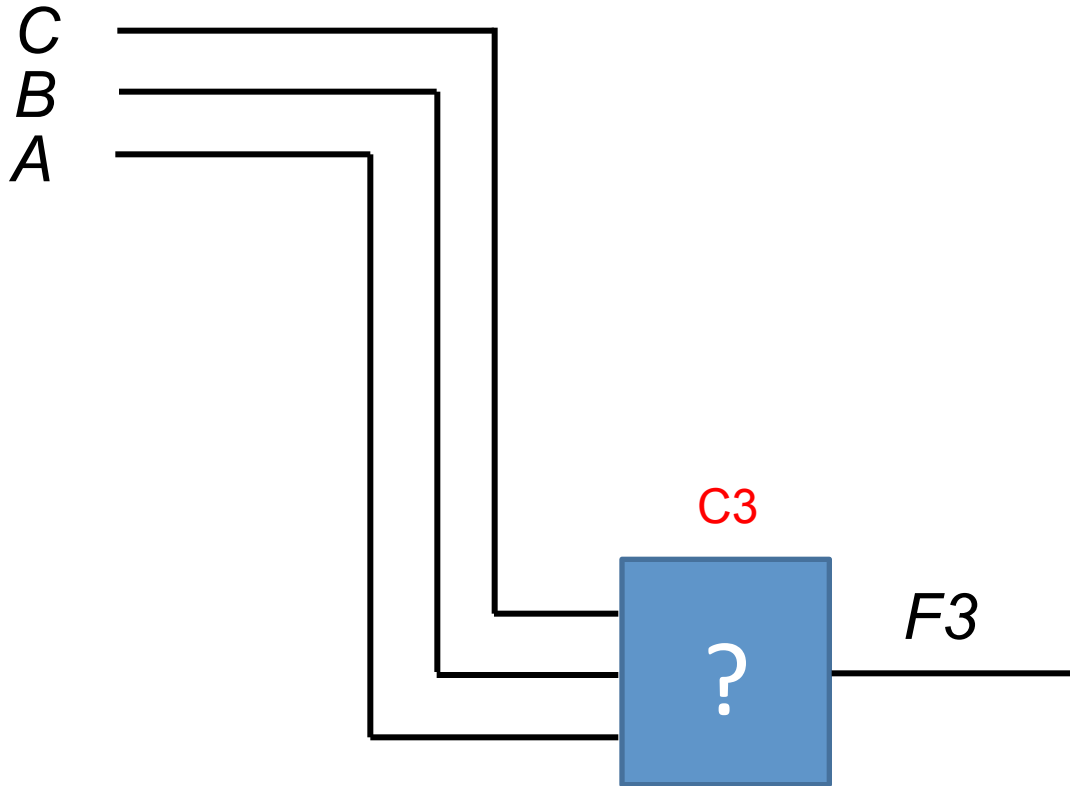
Componente **C2**: implementa  $F2 = B \text{ xor } C$



# Projeto hierárquico

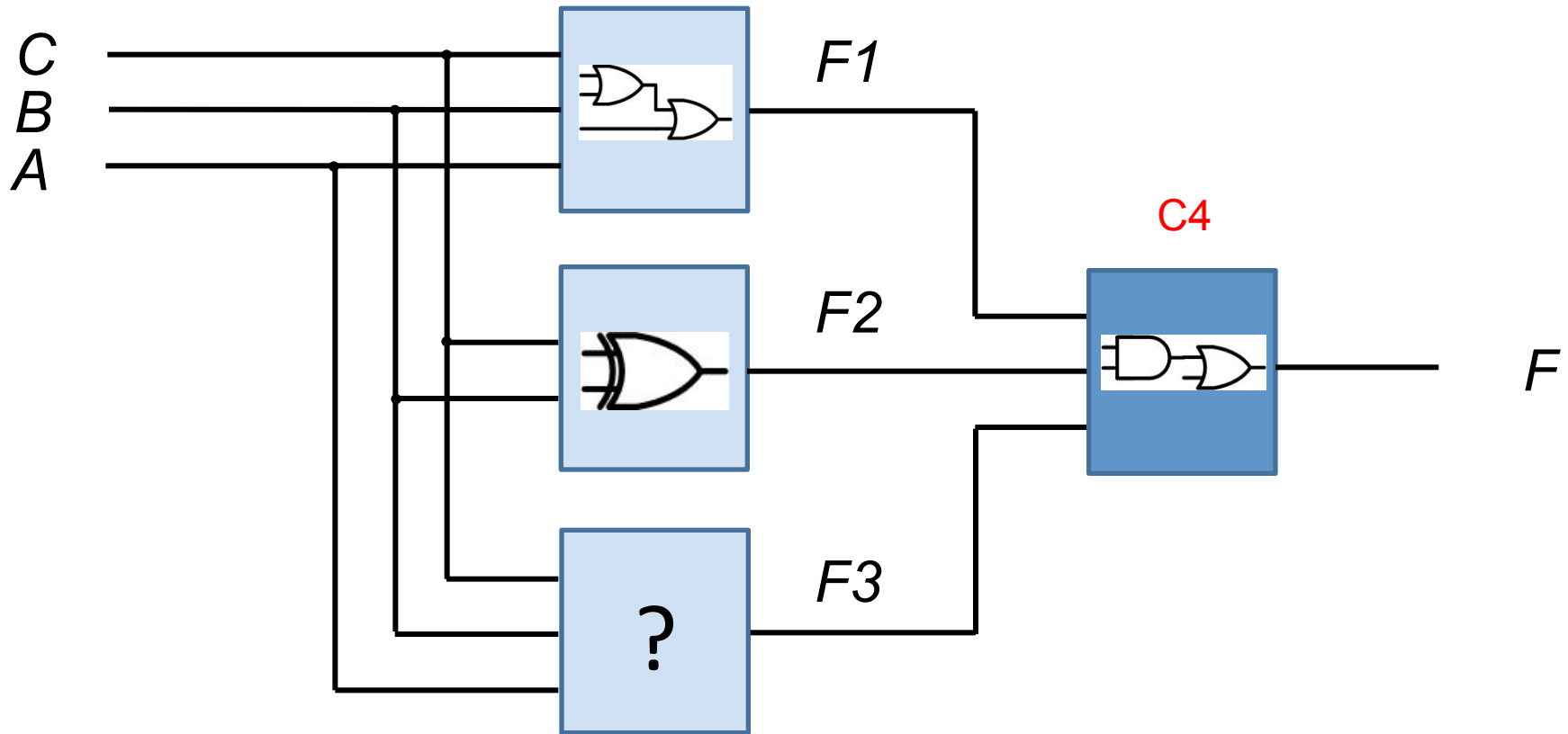
---

Componente **C3**: a ser implementado como tarefa da aula



# Projeto hierárquico

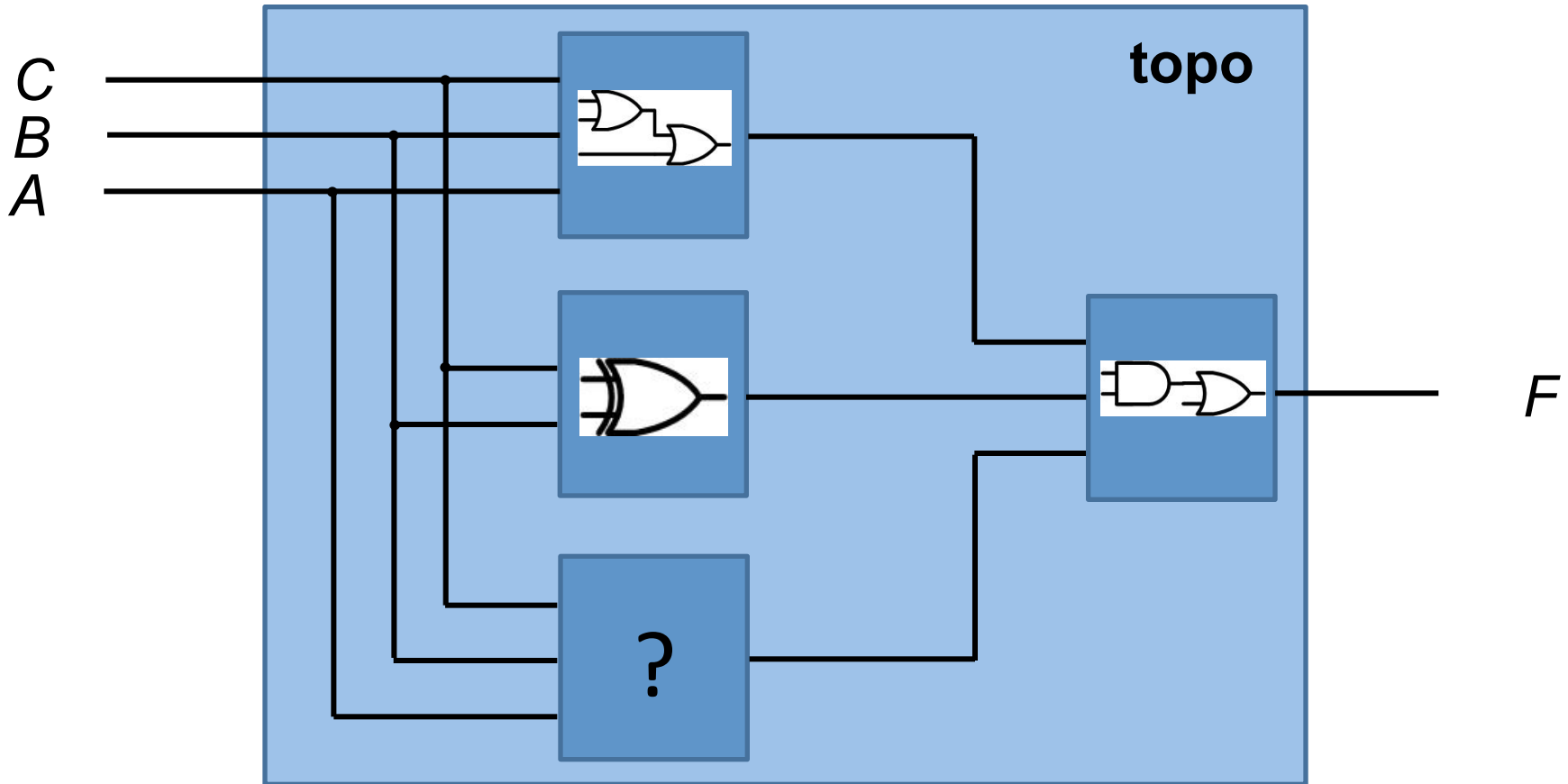
Componente **C4**: implementa  $F = (F1 \text{ and } F2) \text{ or } F3$





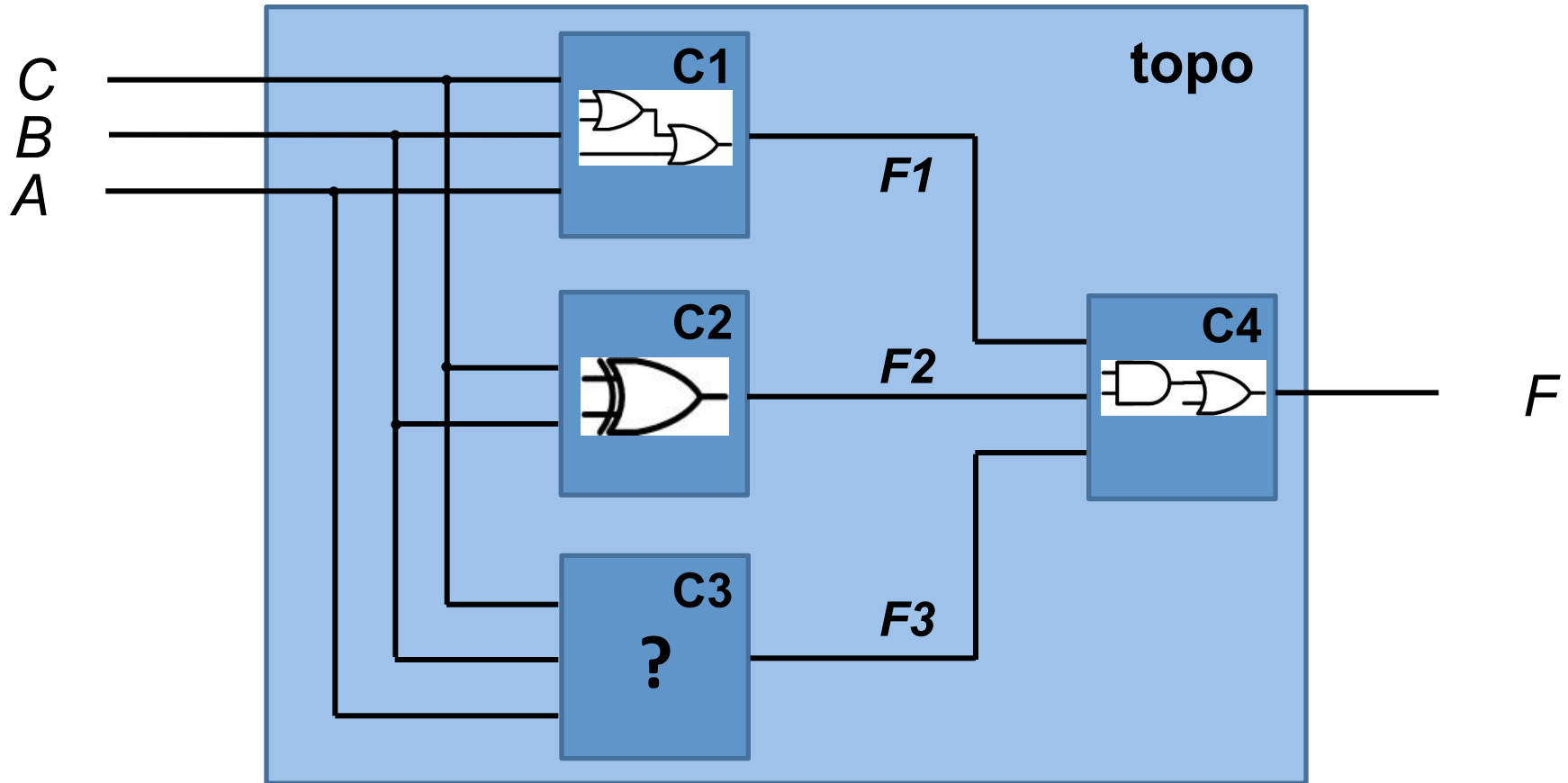
# Projeto hierárquico

O componente TOPO possui 4 componentes internamente.



# Projeto hierárquico

O componente TOPO possui 4 componentes internamente.

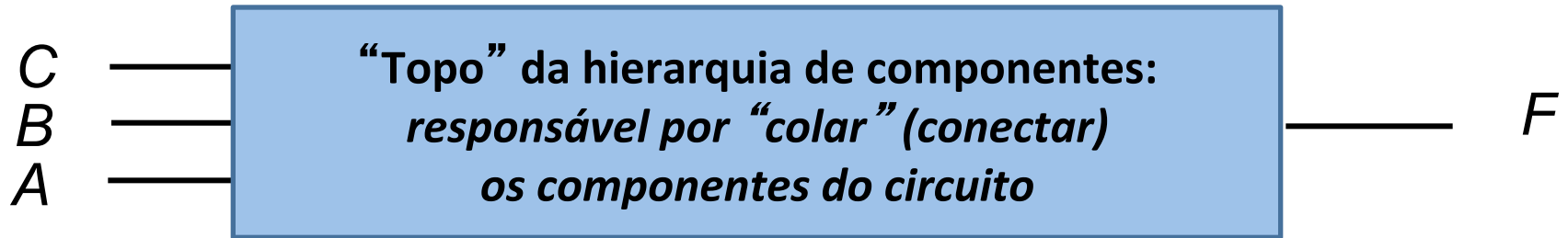


# Projeto hierárquico

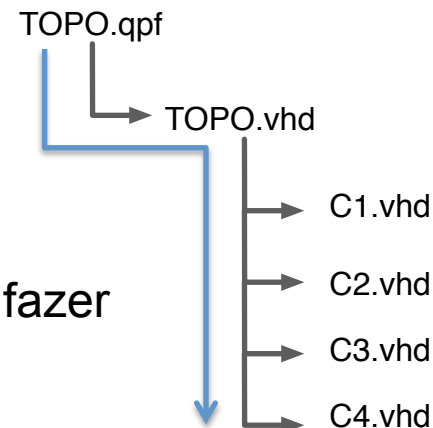
Componente “TOPO” recebe A, B, C e fornece F como saída.

ENTRADAS

SAÍDAS

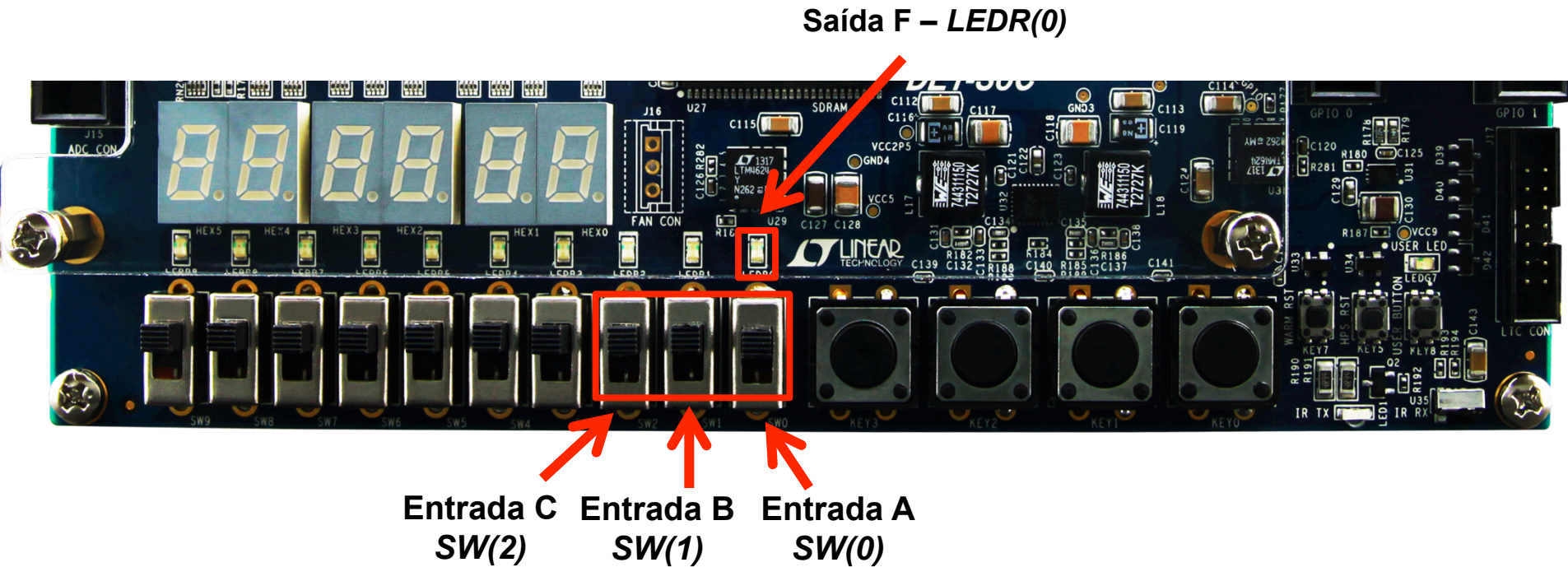


Quando o projeto TOPO.qpf é compilado as componentes são chamadas. Não é preciso fazer a descrição completa na hierarquia superior



***Implementação do circuito no kit de desenvolvimento***

# Implementação do circuito no kit de desenvolvimento

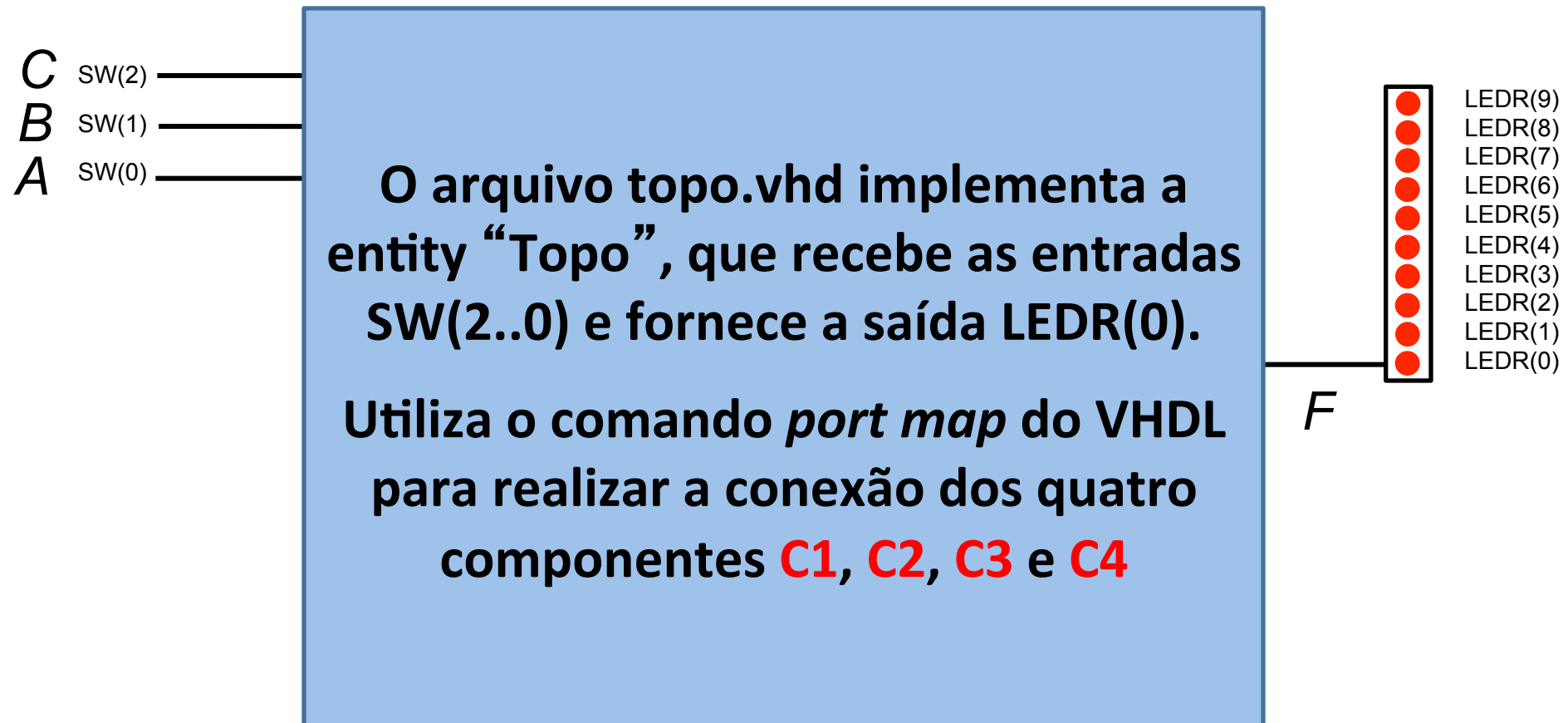


**A, B, C são as chaves SW(0), SW(1) e SW(2), respectivamente.  
F é o LED vermelho LEDR(0).**

---

### ENTRADAS

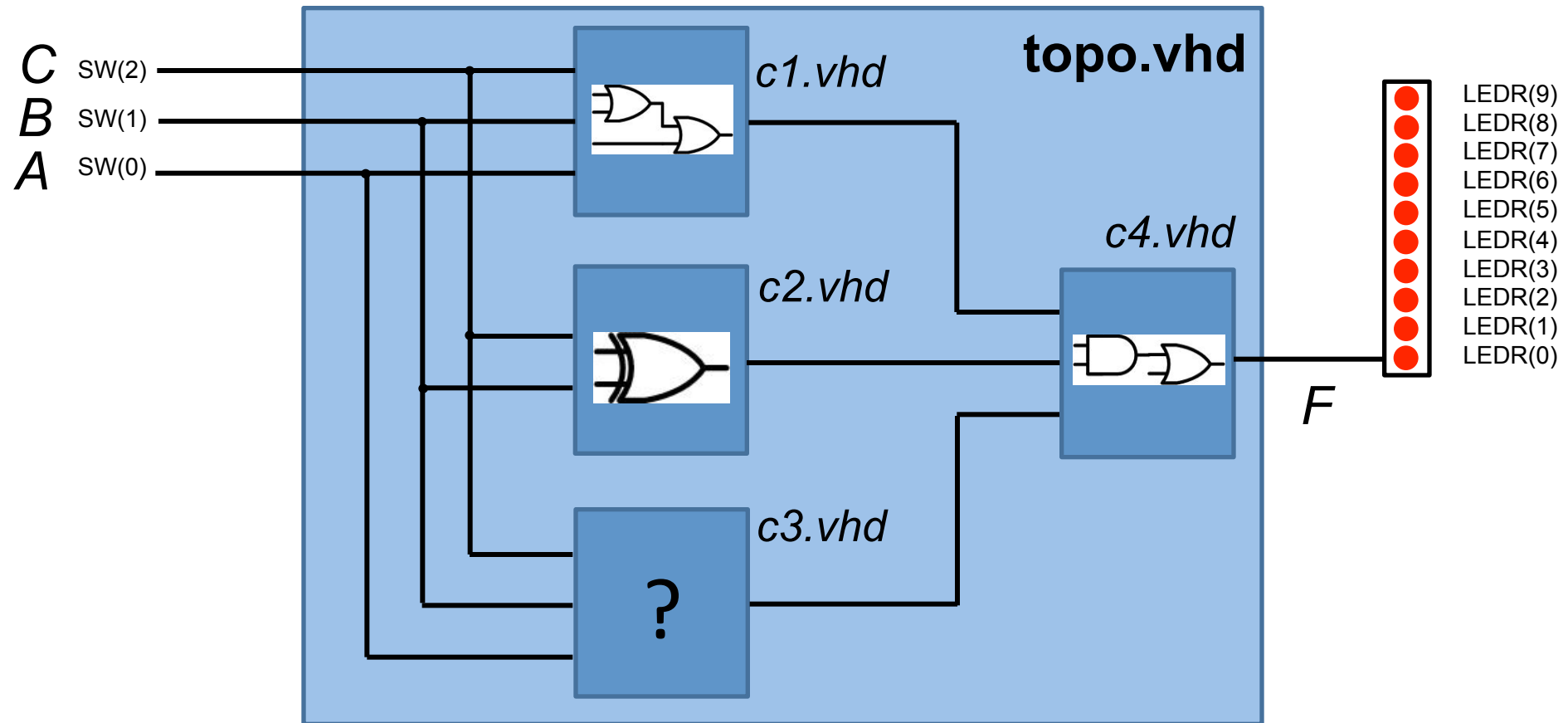
### SAÍDAS



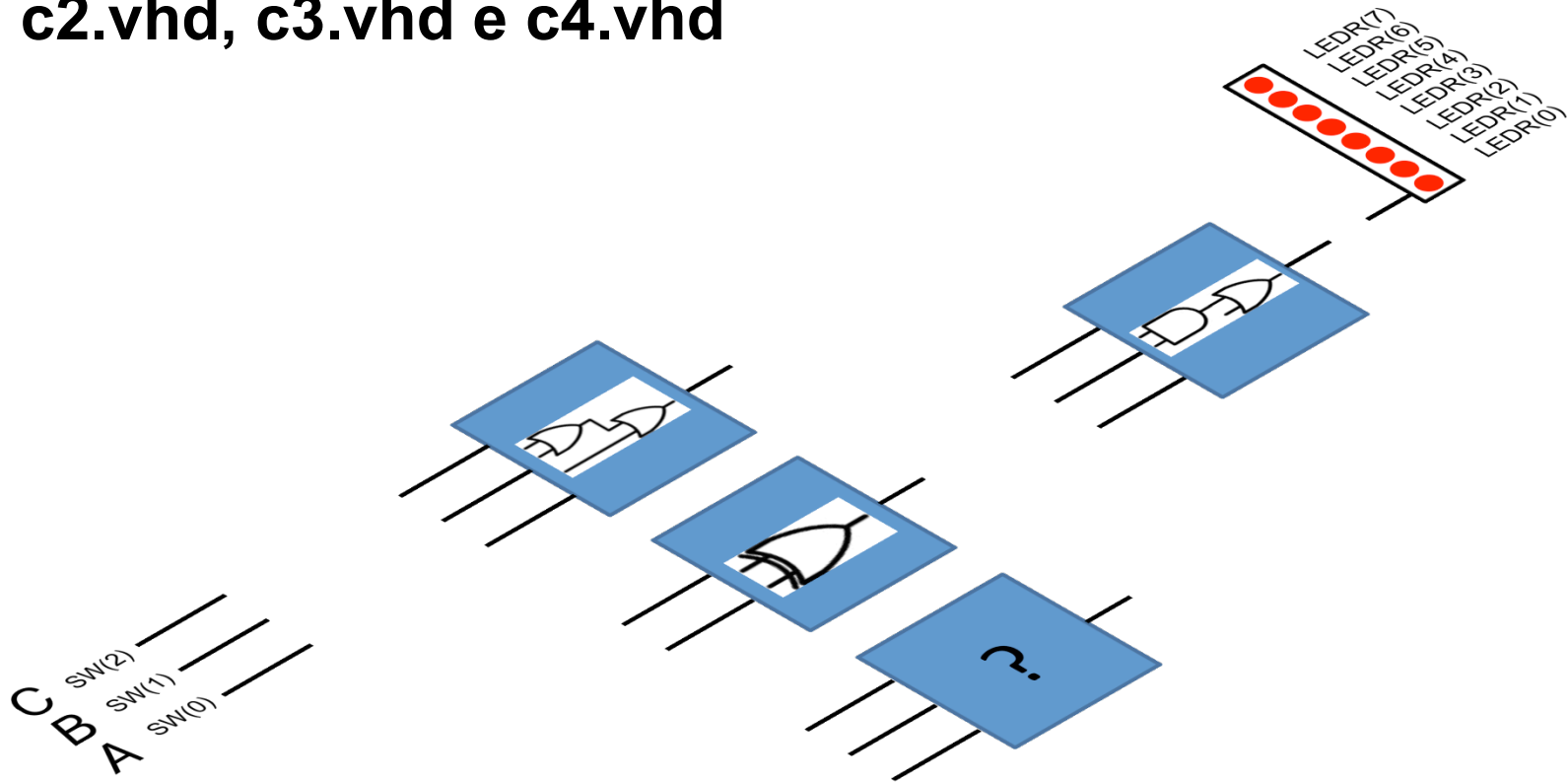
**A, B, C são as chaves SW(0), SW(1) e SW(2), respectivamente.  
F é o LED vermelho LEDR(0).**

## ENTRADAS

## SAÍDAS

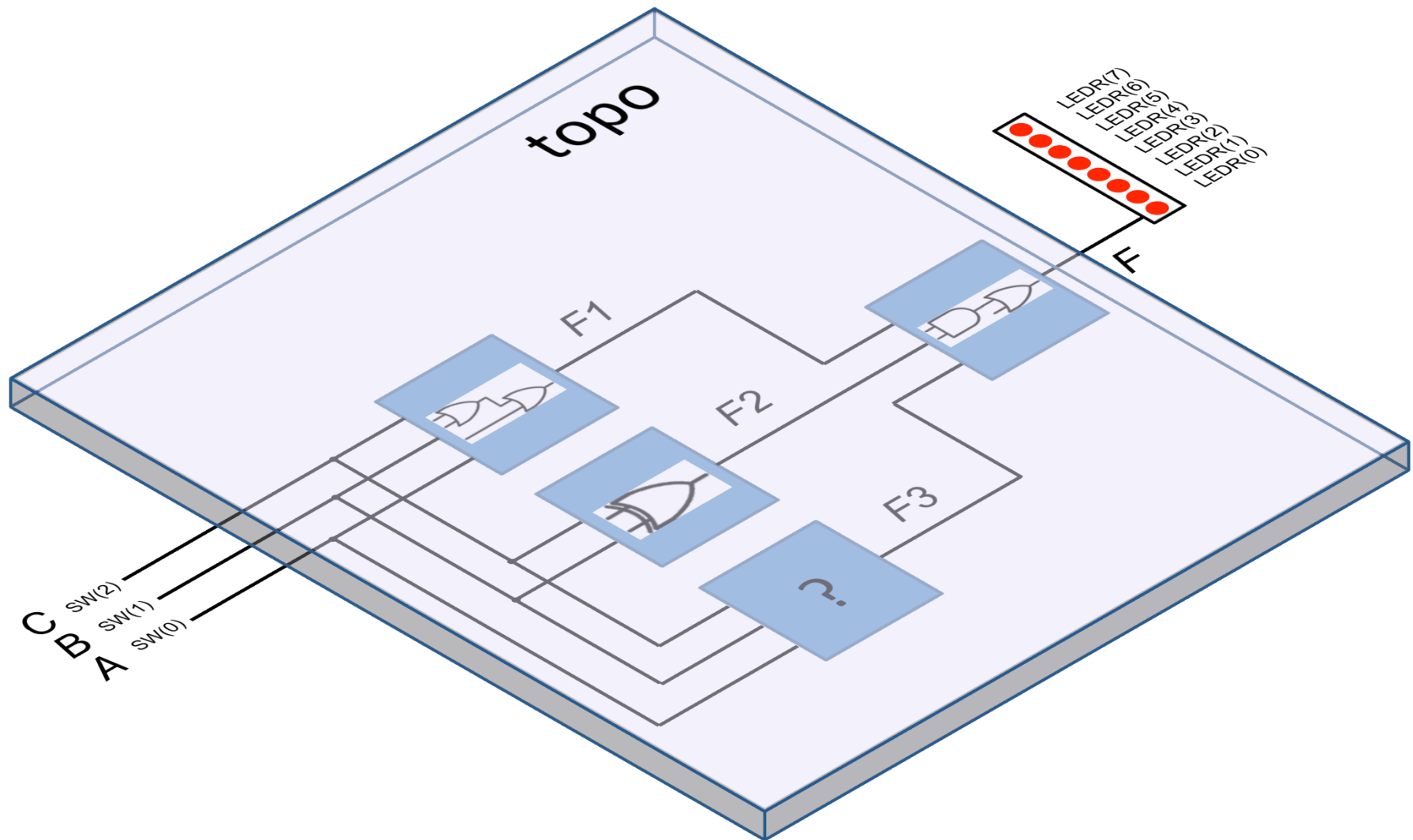


- O projeto possui um total de 5 arquivos VHDL, cada um com uma *entity* e uma *architecture*.
- Os quatro componentes abaixo estão nos arquivos c1.vhd, c2.vhd, c3.vhd e c4.vhd

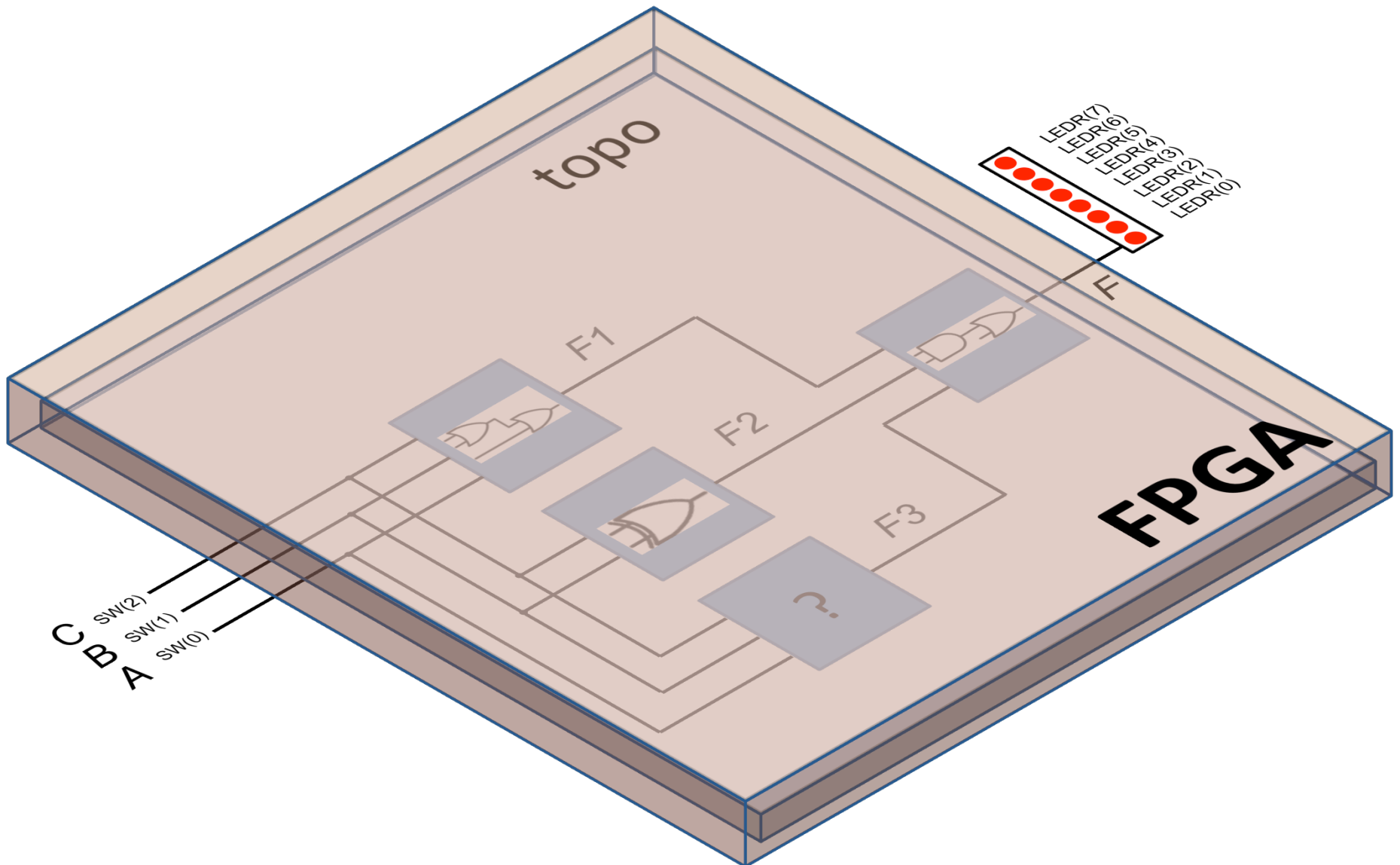




**A *entity/architecture* “topo” realiza as conexões entre os componentes, e está no arquivo topo.vhd**



# Implementação do circuito no FPGA do kit de desenvolvimento

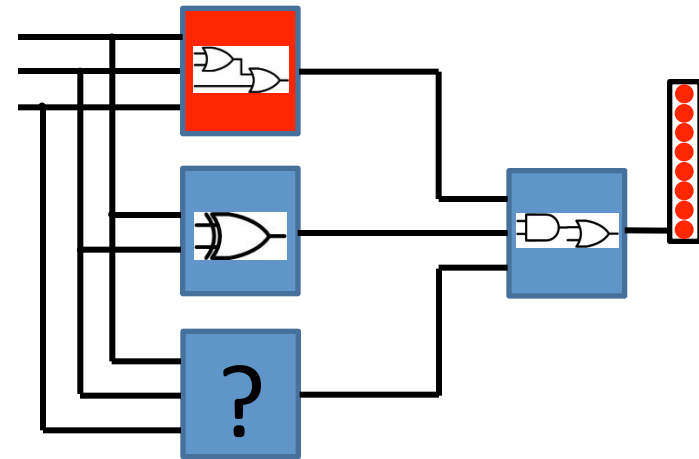


# Componente C1 (arquivo *c1.vhd*)

```
library IEEE;  
use IEEE.Std_Logic_1164.all;
```

```
entity C1 is  
  port (A: in std_logic;  
        B: in std_logic;  
        C: in std_logic;  
        F: out std_logic  
        );  
end C1;
```

```
architecture c1_estr of C1 is  
begin  
  F <= A or B or C;  
end c1_estr;
```

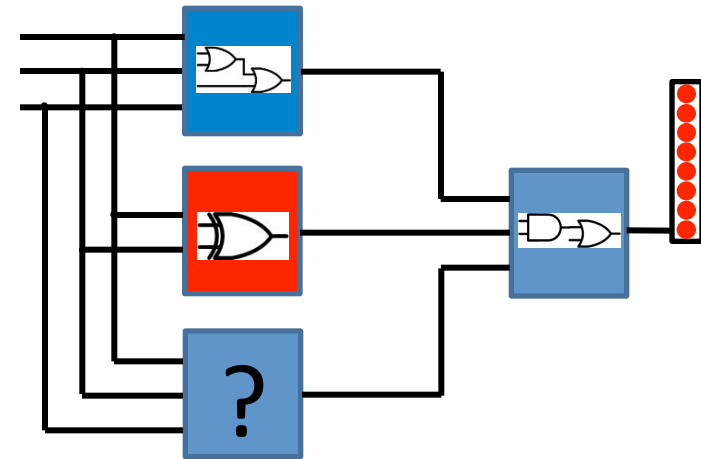


# Componente C2 (arquivo *c2.vhd*)

```
library IEEE;  
use IEEE.Std_Logic_1164.all;
```

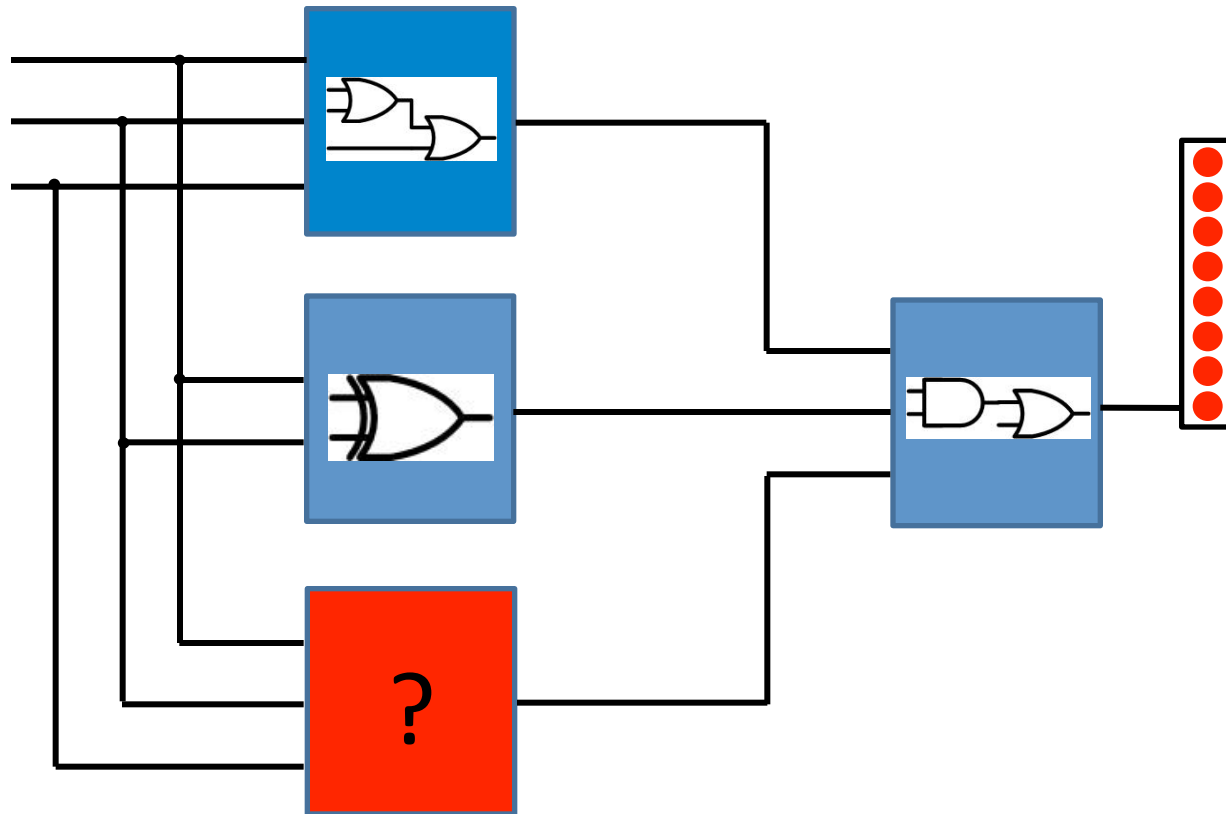
```
entity C2 is  
  port (A: in std_logic;  
        B: in std_logic;  
        F: out std_logic  
        );  
end C2;
```

```
architecture c2_estr of C2 is  
begin  
  F <= A xor B;  
end c2_estr;
```



# Componente C3 (arquivo *c3.vhd*) - tarefa da aula!

---

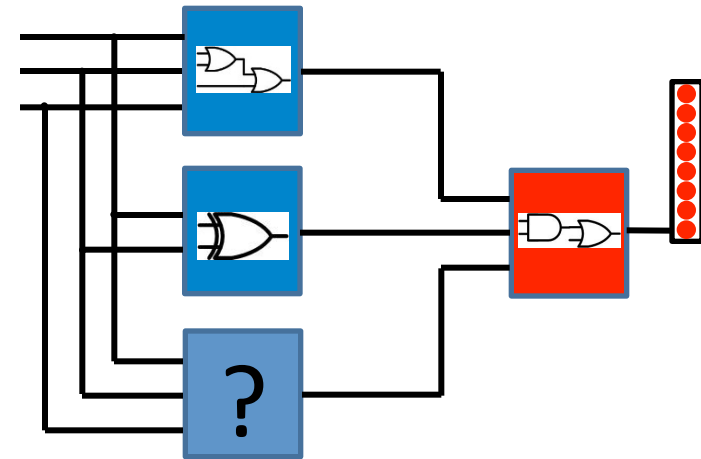


# Componente C4 (arquivo *c4.vhd*)

```
library IEEE;  
use IEEE.Std_Logic_1164.all;
```

```
entity C4 is  
  port (A: in std_logic;  
        B: in std_logic;  
        C: in std_logic;  
        F: out std_logic  
        );  
end C4;
```

```
architecture c4_estr of C4 is  
begin  
  F <= (A and B) or C;  
end c4_estr;
```



# Componente Topo (arquivo *topo.vhd*)

```
library ieee;  
use ieee.std_logic_1164.all;  
entity topo is  
  port ( SW : IN STD_LOGIC_VECTOR(9 downto 0);  
         LEDR : OUT STD_LOGIC_VECTOR(9 downto 0)  
  );
```

```
end topo;  
architecture topo_estru of topo is  
  signal F1, F2, F3: std_logic;
```

*component C1*

```
  port (A : in std_logic;  
        B : in std_logic;  
        C : in std_logic;  
        F : out std_logic);
```

*end component;*

*component C2*

```
  port (A : in std_logic;  
        B : in std_logic;  
        F : out std_logic);
```

*end component;*

**-- INCLUIR AQUI O C3**

*component C4*

```
  port (A : in std_logic;  
        B : in std_logic;  
        C : in std_logic;  
        F : out std_logic
```

```
  );
```

*end component;*

**begin**

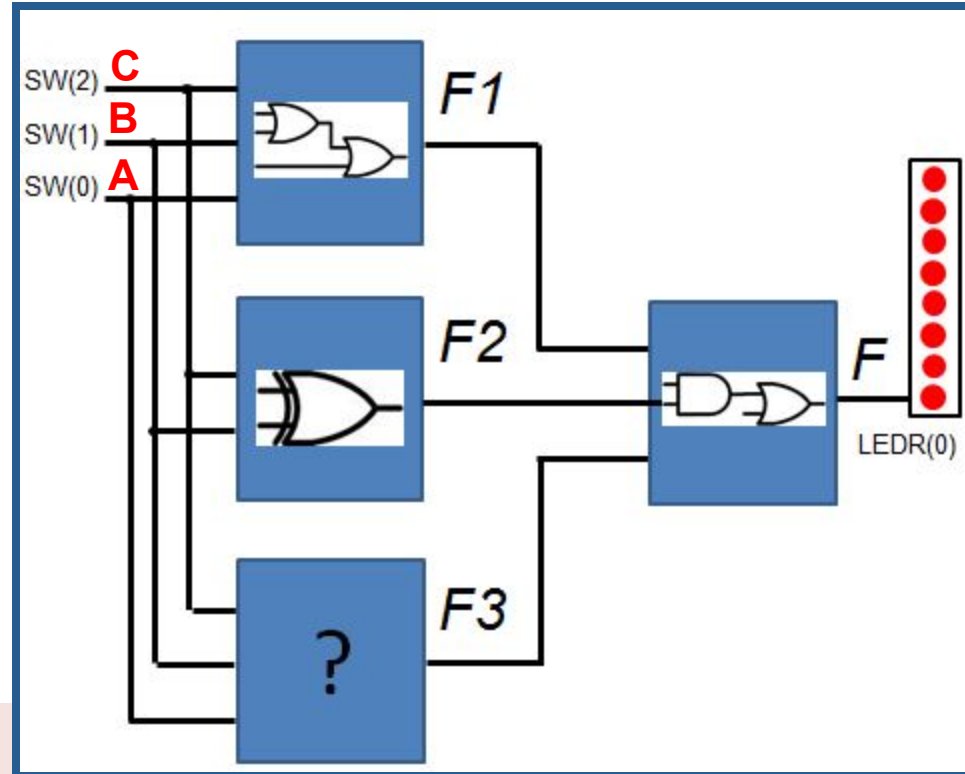
**L0: C1 port map (SW(0), SW(1), SW(2), F1);**

**L1: C2 port map (SW(1), SW(2), F2);**

**-- INCLUIR AQUI AS CONEXÕES DE C3**

**L3: C4 port map (F1, F2, F3, LEDR(0));**

**end topo\_estru; -- esse e' o END da architecture**



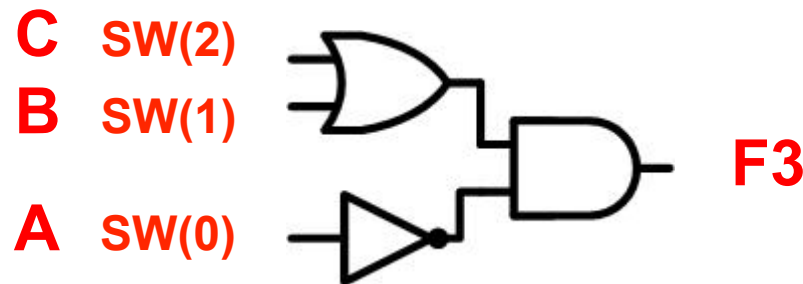
**Tarefa a ser realizada na aula prática**



# Descrição da tarefa

---

- Utilizando os componentes C1, C2 e C4 como exemplo, desenvolver o componente **C3** em VHDL que implementa a funcionalidade do seguinte circuito:



- Editar o arquivo topo.vhd, e realizar a inclusão do novo componente C3 no circuito, utilizando as construções do VHDL *component* e *port map*, conforme descrito nos slides anteriores.

# Análise dos resultados

A partir da análise da função lógica, preencher a tabela verdade a seguir, e comparar com os resultados obtidos na simulação e na execução na placa DE2

$$F = (F1 \text{ and } F2) \text{ or } F3$$

$$F1 = A \text{ or } B \text{ or } C$$

$$F2 = B \text{ xor } C$$

$$F3 = (B \text{ or } C) \text{ and } (\text{not } A)$$

Entrada C B A  
SW(2..0)

Saída F  
LEDR(0)

0 0 0

0 0 1

0 1 0

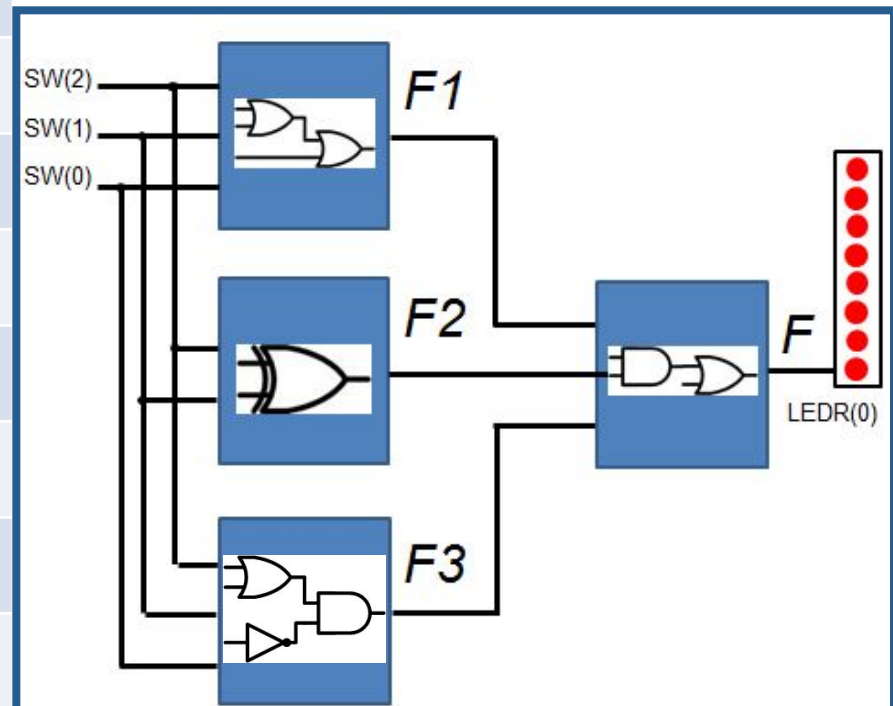
0 1 1

1 0 0

1 0 1

1 1 0

1 1 1

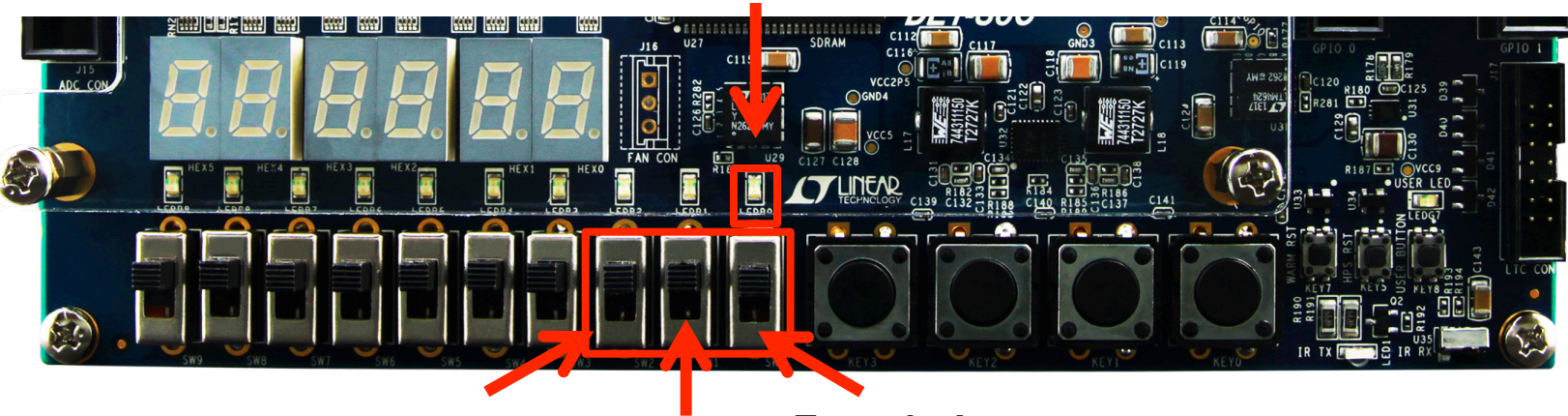


# Análise dos resultados

Entrada  SW(2..0)  C B A	Saída “Equação” $F = (F1 \text{ and } F2) \text{ or } F3$ onde: F1 = A or B or C F2 = B xor C F3 = (B or C) and (not A)	Saída “Simulação”  Obtida no sinal LEDR(0), no <u>simulador</u>	Saída “FPGA”  Obtida no LEDR(0) da <u>placa DE2</u>
0 0 0			
0 0 1			
0 1 0			
0 1 1			
1 0 0			
1 0 1			
1 1 0			
1 1 1			

# Implementação do circuito no kit de desenvolvimento

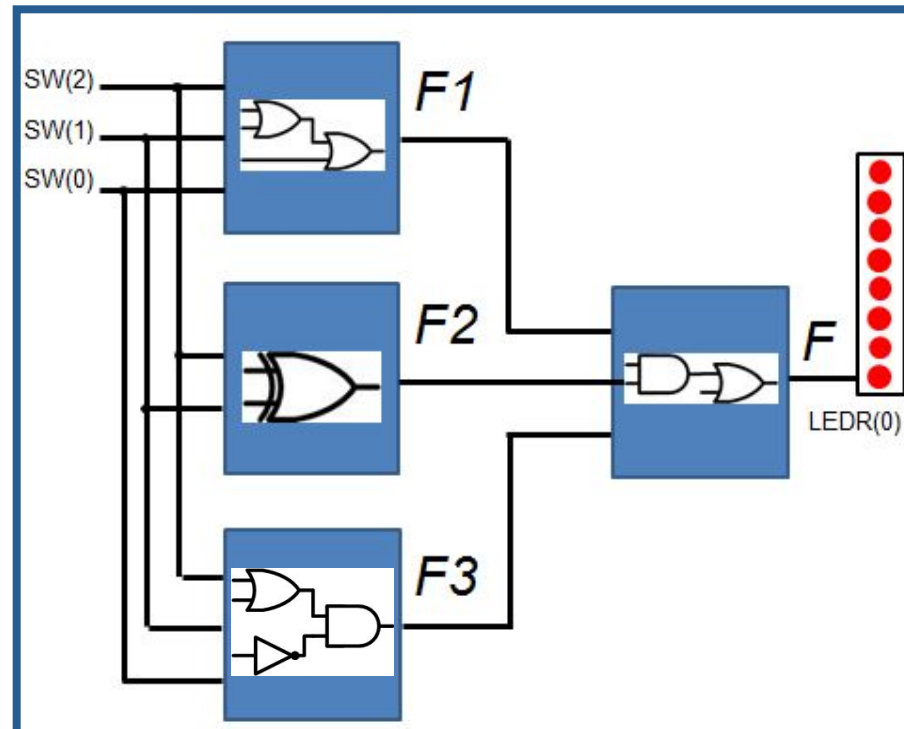
Saída F - LEDR(0)



Entrada C  
SW(2)

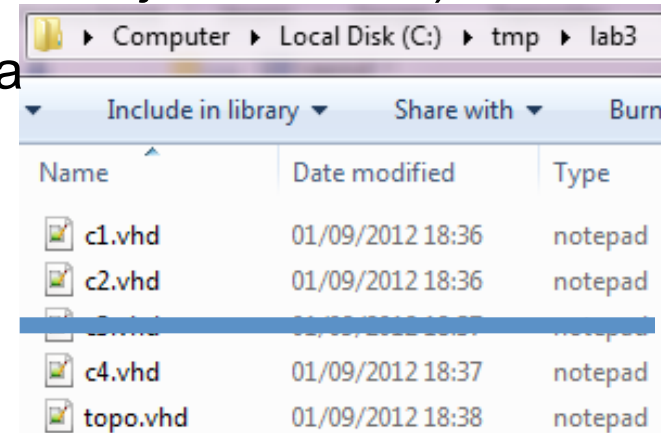
Entrada B  
SW(1)

Entrada A  
SW(0)



# Dicas úteis

1. Entrar no site da disciplina (<http://gse.ufsc.br/bezerra>) e obter os fontes VHDL a serem utilizados na aula (ou copiar e colar dos slides).
2. Criar uma pasta vazia e copiar os arquivos VHDL. **Utilizar um caminho sem espaços ou caracteres acentuados nos nomes das pastas.**
3. Executar o Quartus e criar um novo projeto (“New Project Wizard”).
4. No campo “diretório de trabalho”, apontar para a nova pasta contendo os arquivos VHDL (no exemplo ao lado está em c:\tmp\lab3).
5. No campo “nome do projeto”, colocar a palavra “topo”.
6. A *entity* “top-level” também se chamará “topo”.
7. Os 4 arquivos VHDL poderão ser incluídos no projeto durante ou após o *wizard*. Para adicionar durante o *wizard*, basta selecionar a pasta em que os arquivos se encontram (na etapa *add files*), e adicionar os arquivos.
8. Selecionar o FPGA existente na placa, conforme tutorial dos laboratórios anteriores.



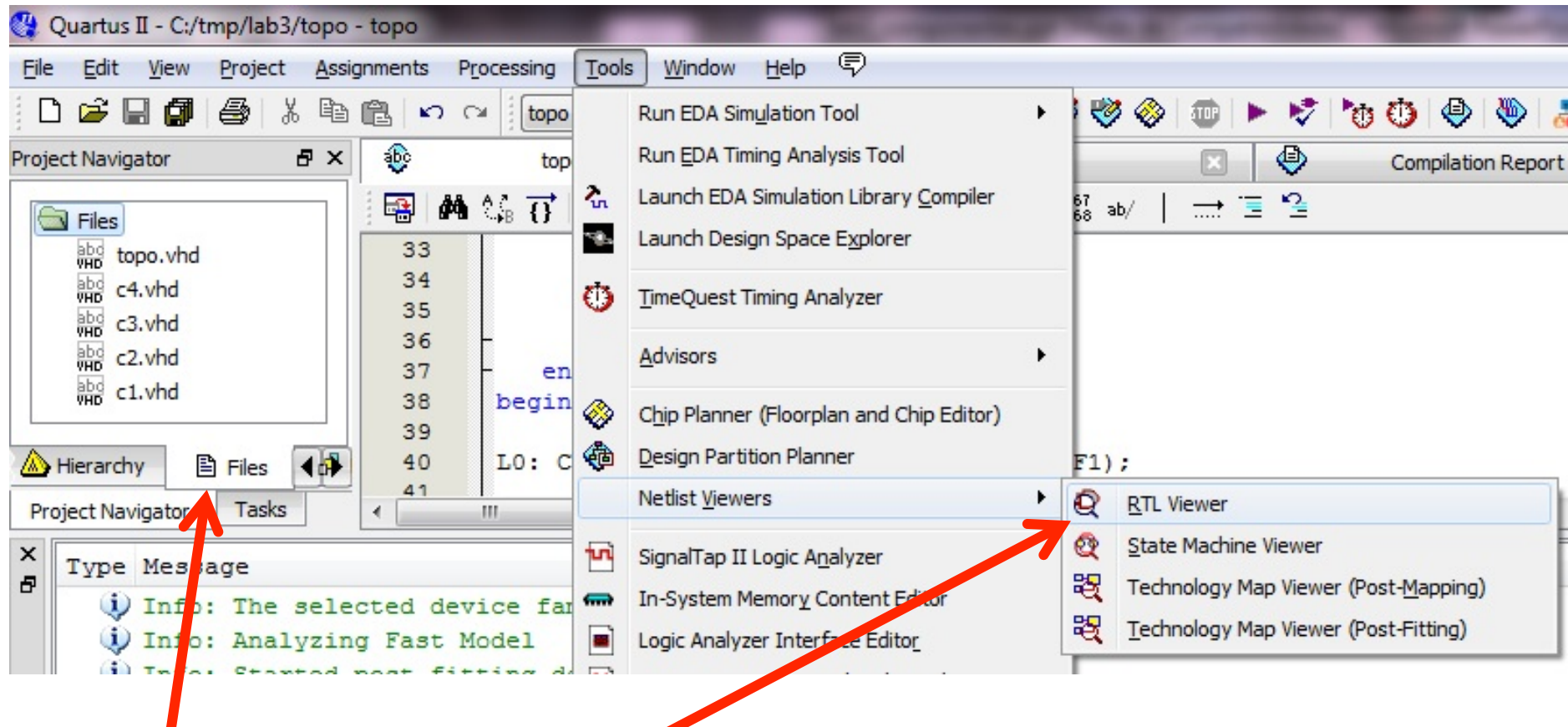
→ FIM DO WIZARD

# Dicas úteis

---

9. Criar um novo arquivo c3.vhd (se ainda não tiver sido criado), e realizar a implementação desse componente conforme circuito especificado na tarefa.
10. Editar o arquivo topo.vhd, e incluir o novo componente C3.
11. Obter o arquivo de pinos na página da disciplina (pinos.qsf), copiar para a pasta do projeto, e importar esse arquivo utilizando a opção “*import assignments*” do Quartus II conforme o tutorial da aula anterior.
12. Realizar a síntese (*compile*).
13. Realizar a simulação (diagrama de formas de ondas).
14. Verificar a funcionalidade do circuito no kit de desenvolvimento.

# Dicas úteis



- Verificar se os 5 arquivos VHDL estão listados no projeto (*Files*).
- Usar o “*RTL Viewer*” para obter o diagrama de blocos e verificar a correta geração do projeto hierárquico.

# Dicas úteis

RTL Viewer – dois clicks para “entrar” no bloco (hierarquia).

