



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



**<http://gse.ufsc.br>**

# **“EEL7020 – Sistemas Digitais”**

**Prof. Eduardo Augusto Bezerra**

**Eduardo.Bezerra@ufsc.br**

**Florianópolis, março de 2015.**

# Sistemas Digitais

*Circuito multiplexador - Mux*

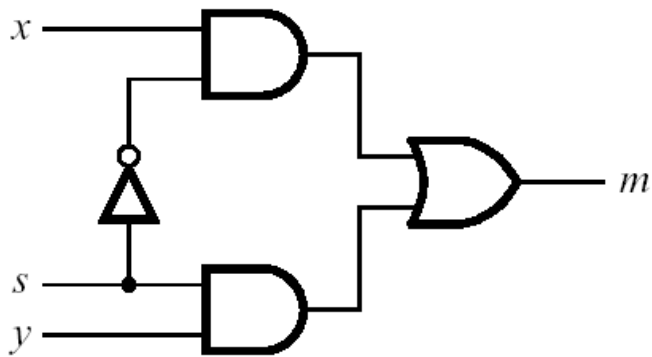
# Objetivos do laboratório

---

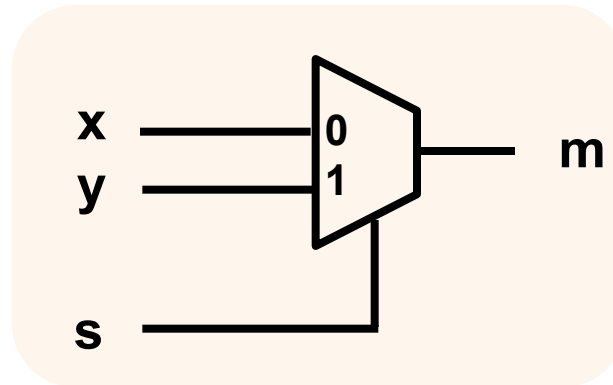
1. Entender o conceito de “multiplexador”.
2. Implementação de multiplexador em VHDL utilizando apenas funções booleanas (VHDL estrutural).
3. Implementação de multiplexador em VHDL utilizando *when / else* (VHDL comportamental).
4. Estudo de caso: uso de mux no projeto hierárquico do lab anterior.

# Projeto de multiplexador - MUX 2x1

- No circuito, se  $s = 0$ , a saída  $m$  será igual a entrada  $x$ .  
Se  $s = 1$ , a saída  $m$  será igual a entrada  $y$ .



(a)



(b)

$s$	$m$
0	$x$
1	$y$

(c)

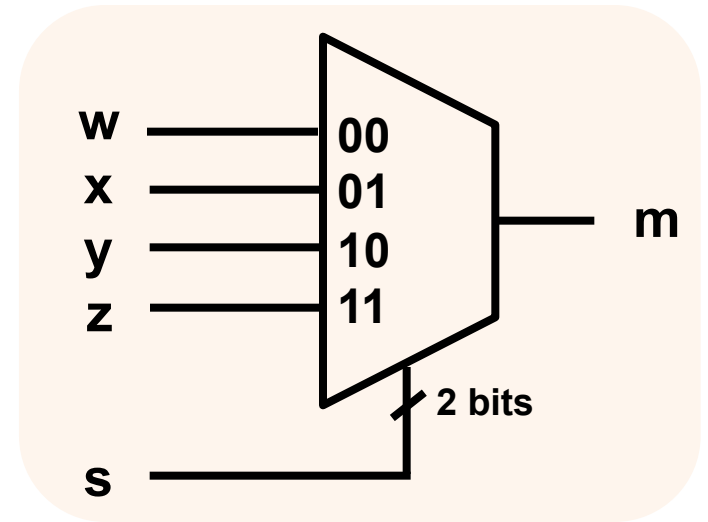
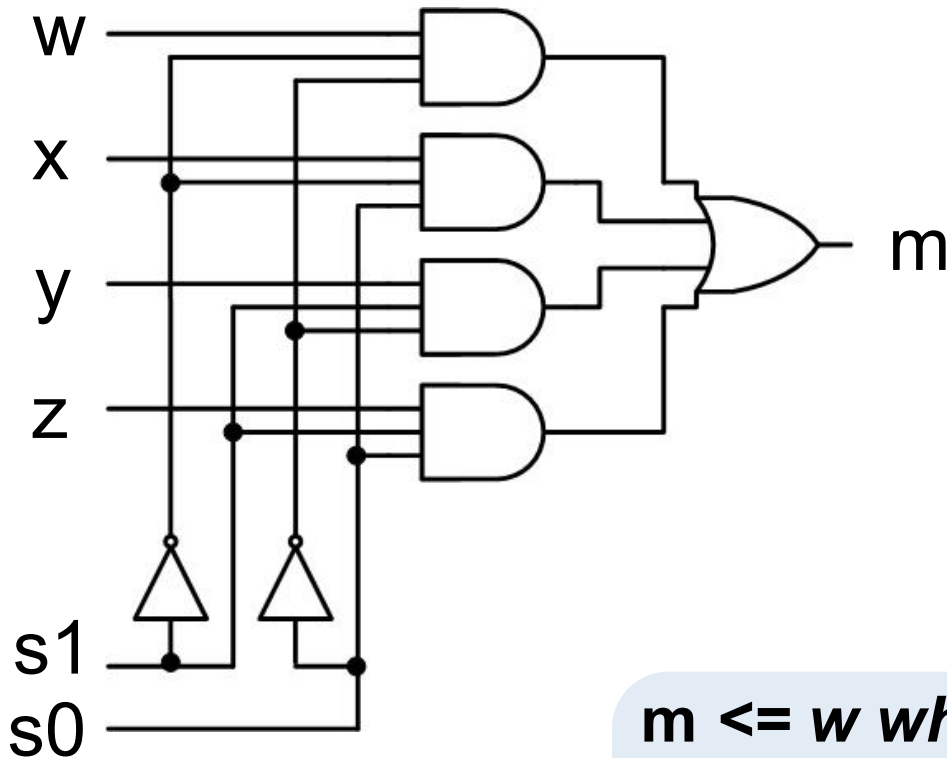
VHDL estrutural:

$m \leftarrow (NOT(s) AND x) OR (s AND y)$

VHDL comportamental:

$m \leftarrow x$  when  $s = '0'$  else  
 $y$ ;

# Projeto de multiplexador - MUX 4x1



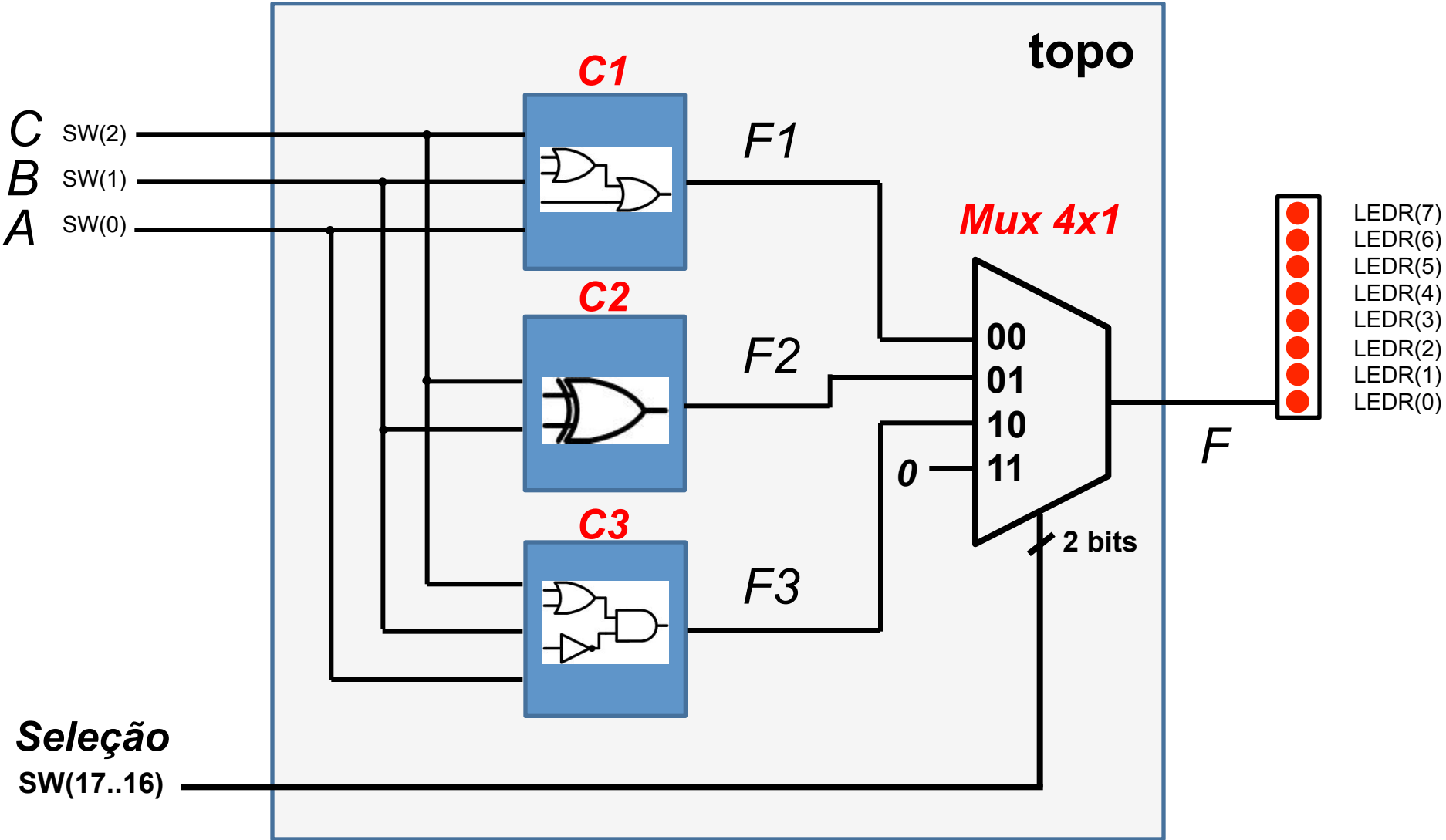
$m \leftarrow w$  when  $s = "00"$  else  
 $x$  when  $s = "01"$  else  
 $y$  when  $s = "10"$  else  
 $z$ ;

## Tarefa a ser realizada na aula prática

**PARTE I – Mux 4x1 em VHDL estrutural**

**PARTE II – Mux 4x1 em VHDL comportamental**

# Reutilizar os arquivos do lab anterior, e realizar as alterações indicadas a seguir (Mux 4x1 no lugar de C4):

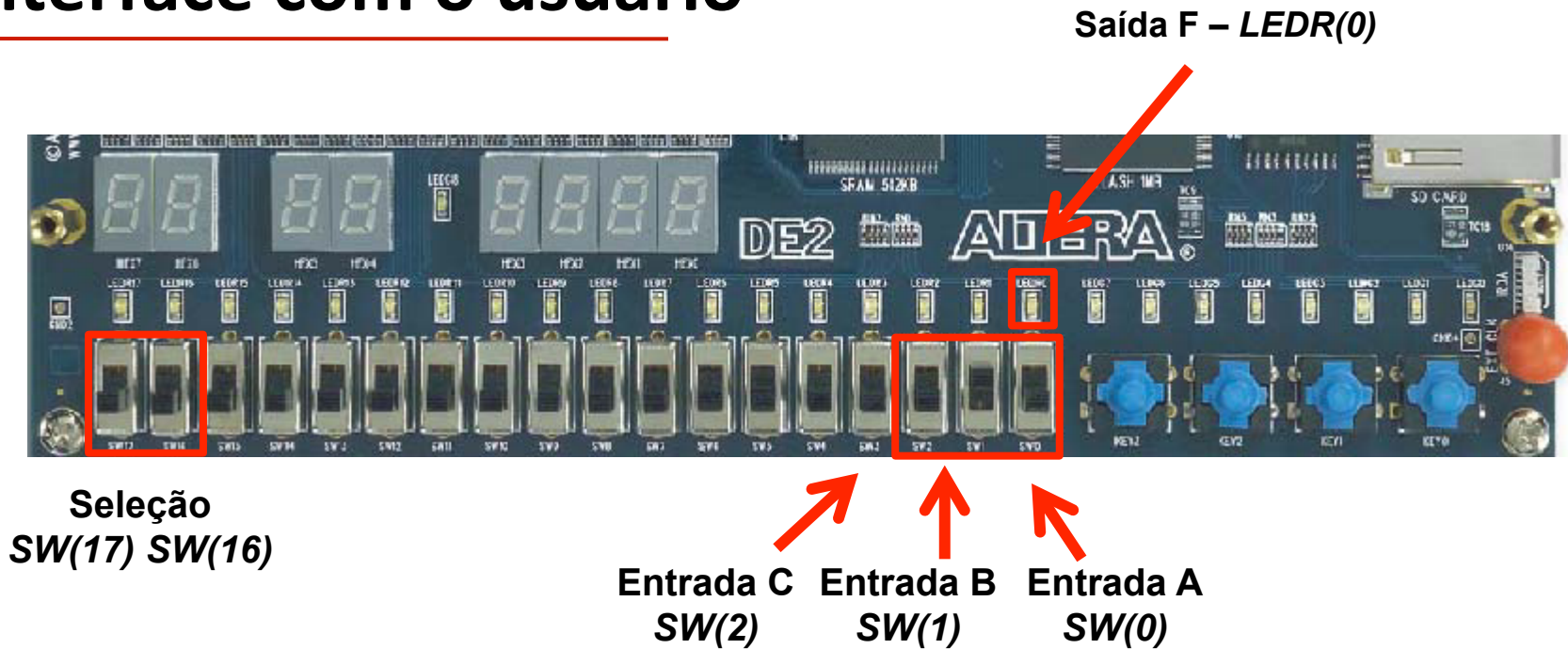


## Descrição da tarefa: **PARTE I – VHDL estrutural**

- Utilizar os componentes C1, C2, C3 e topo do laboratório anterior.
- Remover o componente C4 do projeto:
  - Remover C4 da lista de arquivos
  - Remover o *Component* C4 do topo.vhd
  - Remover o *port map* do C4 do topo.vhd
- Criar um novo arquivo VHDL, e implementar um mux 4x1 utilizando **VHDL estrutural**, conforme exemplo do slide 4.
- Editar o arquivo topo.vhd, e realizar a inclusão do novo componente mux 4x1 no circuito, utilizando as construções do VHDL *component* e *port map*.



# Interface com o usuário



**Seleção**  
**SW(17) SW(16)**

**Entrada C**  
**SW(2)**

**Entrada B**  
**SW(1)**

**Entrada A**  
**SW(0)**

Seleção SW(17..16)	Saída LED_R(0)
00	F1
01	F2
10	F3
11	0 (LED apagado)

## Descrição da tarefa: **PARTE II – VHDL comportamental**

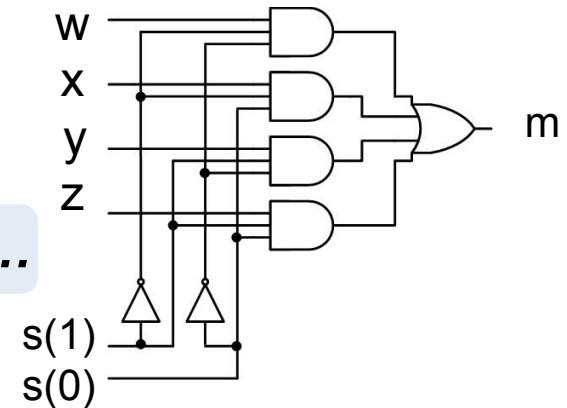
- Criar um novo arquivo VHDL, e implementar um NOVO mux 4x1 utilizando **VHDL comportamental**, conforme exemplos dos slides 4 e 5 (uso de *when / else*).
- Não é necessário alterar o arquivo topo.vhd, pois o novo mux 4x1 deverá possuir exatamente a mesma interface do mux 4x1 implementado em VHDL estrutural da PARTE I (mesma *entity*).
- Na *entity* (Parte I e Parte II) o seletor deverá ser definido como um vetor de dois bits, ou seja `s: in std_logic_vector(1 downto 0)`.
- Importar os pinos, realizar a síntese, simulação, e prototipação na placa DE2, e verificar se o novo circuito funciona de acordo com o esperado.

# Resumo da tarefa – **Atenção!! SÃO DOIS PROJETOS DIFERENTES!**

- **PARTE I** – Projetar e implementar o MUX em **VHDL**

**estrutural:**

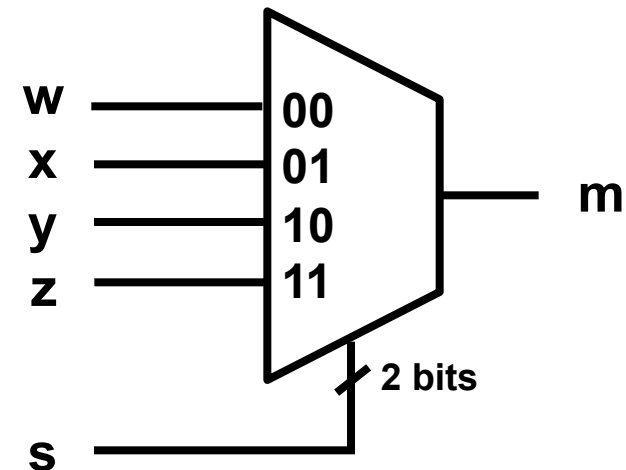
```
m <= (w and ((NOT (s(1)) AND (NOT(s(0)))))) OR ...
```



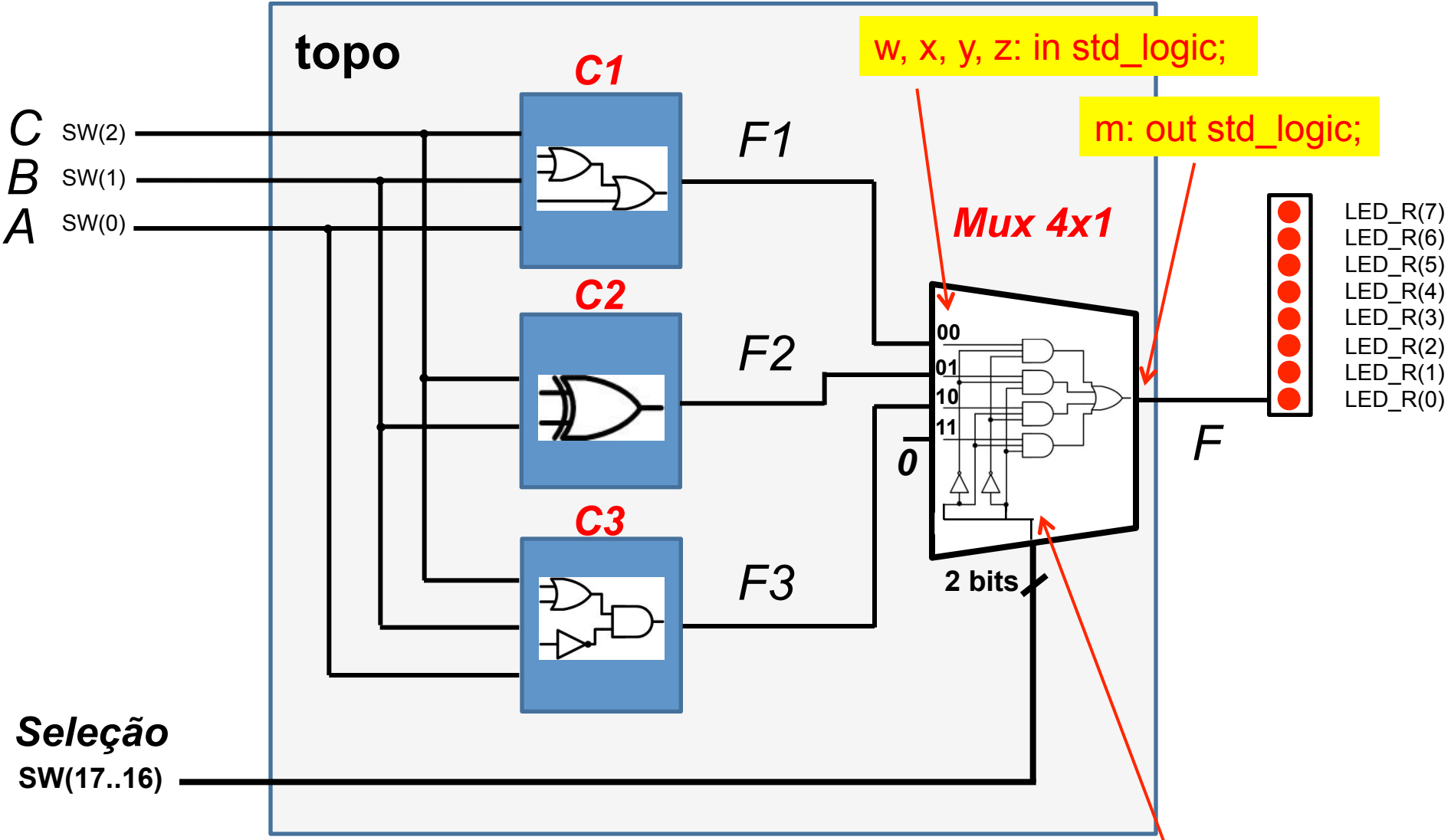
- **PARTE II** – Projetar e implementar o MUX em **VHDL**

**comportamental:**

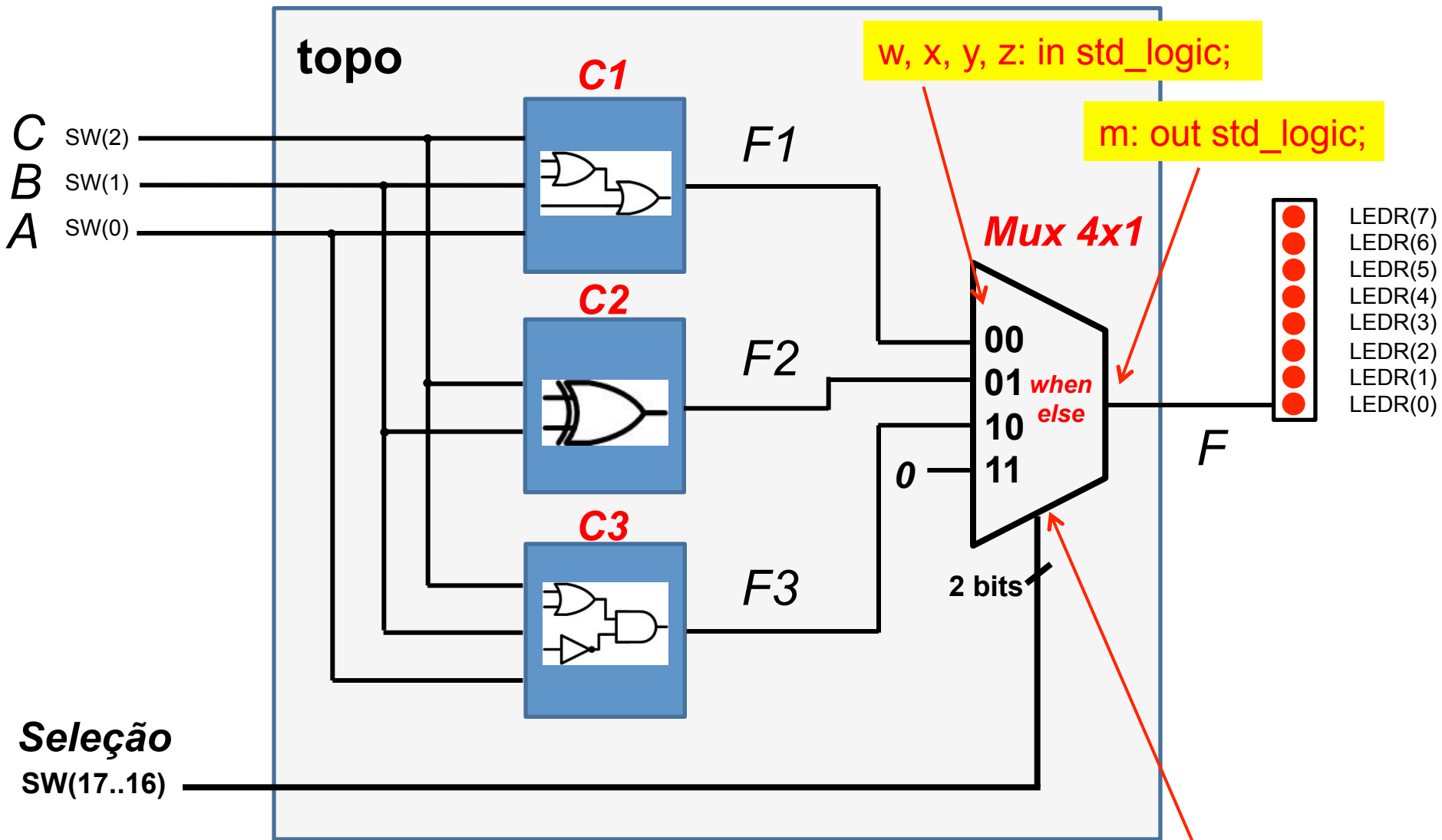
```
m <= w when s = "00" else  
  x when s = "01" else  
  y when s = "10" else  
  z;
```



# PARTE I – VHDL estrutural



# PARTE II – VHDL comportamental



Entradas		Saídas		
SW <sub>2..0</sub> C B A	SW <sub>17..16</sub> Seleção	F1 = A or B or C F2 = B xor C F3 = (B or C) and (not A)	Simulação Livro - <i>step</i> 5 LED_R <sub>0</sub>	FPGA Livro - <i>step</i> 6 LED_R <sub>0</sub>
0 0 0	00	F1 =	0	
0 0 1	00	F1 =	1	
0 1 0	00	F1 =	1	
0 1 1	00	F1 =	1	
1 0 0	00	F1 =	1	
1 0 1	00	F1 =	1	
1 1 0	00	F1 =	1	
1 1 1	00	F1 =	1	
0 0 0	01	F2 =		
0 0 1	01	F2 =		
0 1 0	01	F2 =		
0 1 1	01	F2 =		
1 0 0	01	F2 =		
1 0 1	01	F2 =		
1 1 0	01	F2 =		
1 1 1	01	F2 =		
0 0 0	10	F3 =		
0 0 1	10	F3 =		
0 1 0	10	F3 =		
0 1 1	10	F3 =		
1 0 0	10	F3 =		
1 0 1	10	F3 =		
1 1 0	10	F3 =		
1 1 1	10	F3 =		
X X X	11		0	