



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

Florianópolis, março de 2016.

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

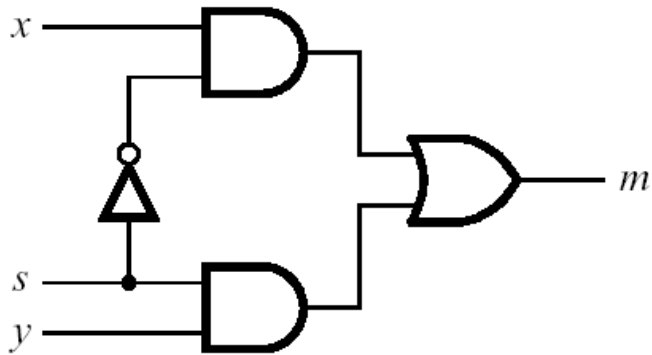
Circuito multiplexador - Mux

Objetivos do laboratório

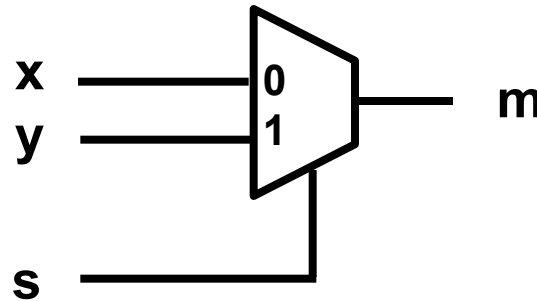
1. Entender o conceito de “multiplexador”.
2. Implementação de multiplexador em VHDL utilizando apenas funções booleanas (**VHDL estrutural**).
3. Implementação de multiplexador em VHDL utilizando *when* / *else* (**VHDL comportamental**).
4. Estudo de caso: uso de multiplexador no projeto hierárquico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

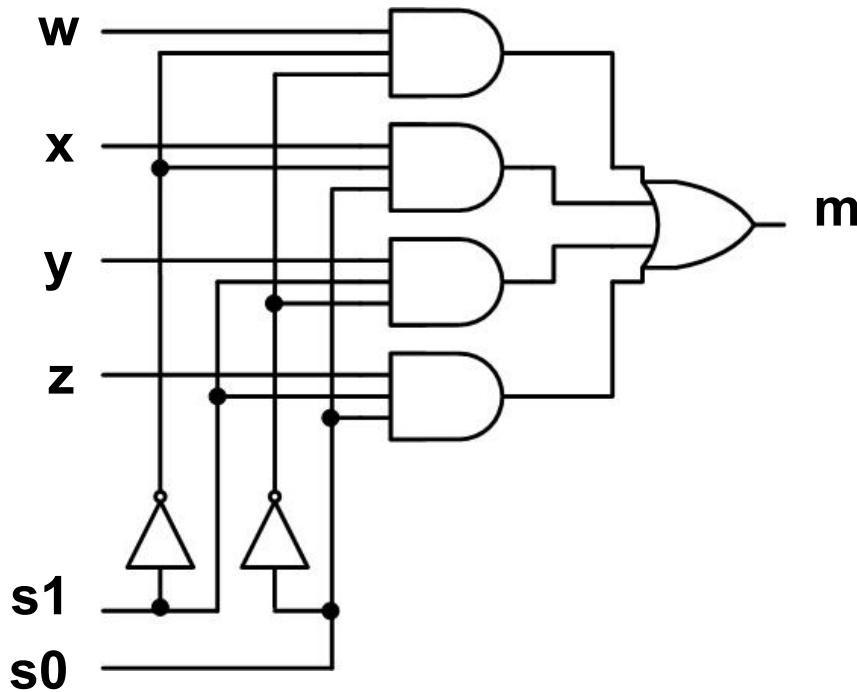
VHDL estrutural:

```
m <= (NOT (s) AND x) OR (s AND y)
```

VHDL comportamental:

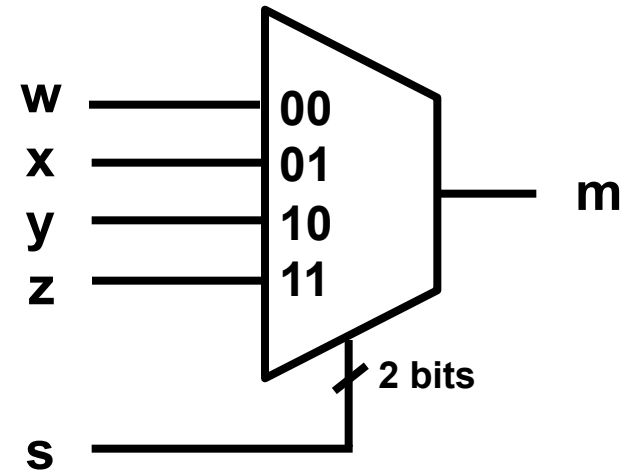
```
m <= x when s = '0' else  
y;
```

Projeto de multiplexador - MUX 4x1



VHDL estrutural:

```
m <= ((((((not (s1)) and not (s0))
and w) and x) and y) and z) or ...
```



VHDL comportamental:

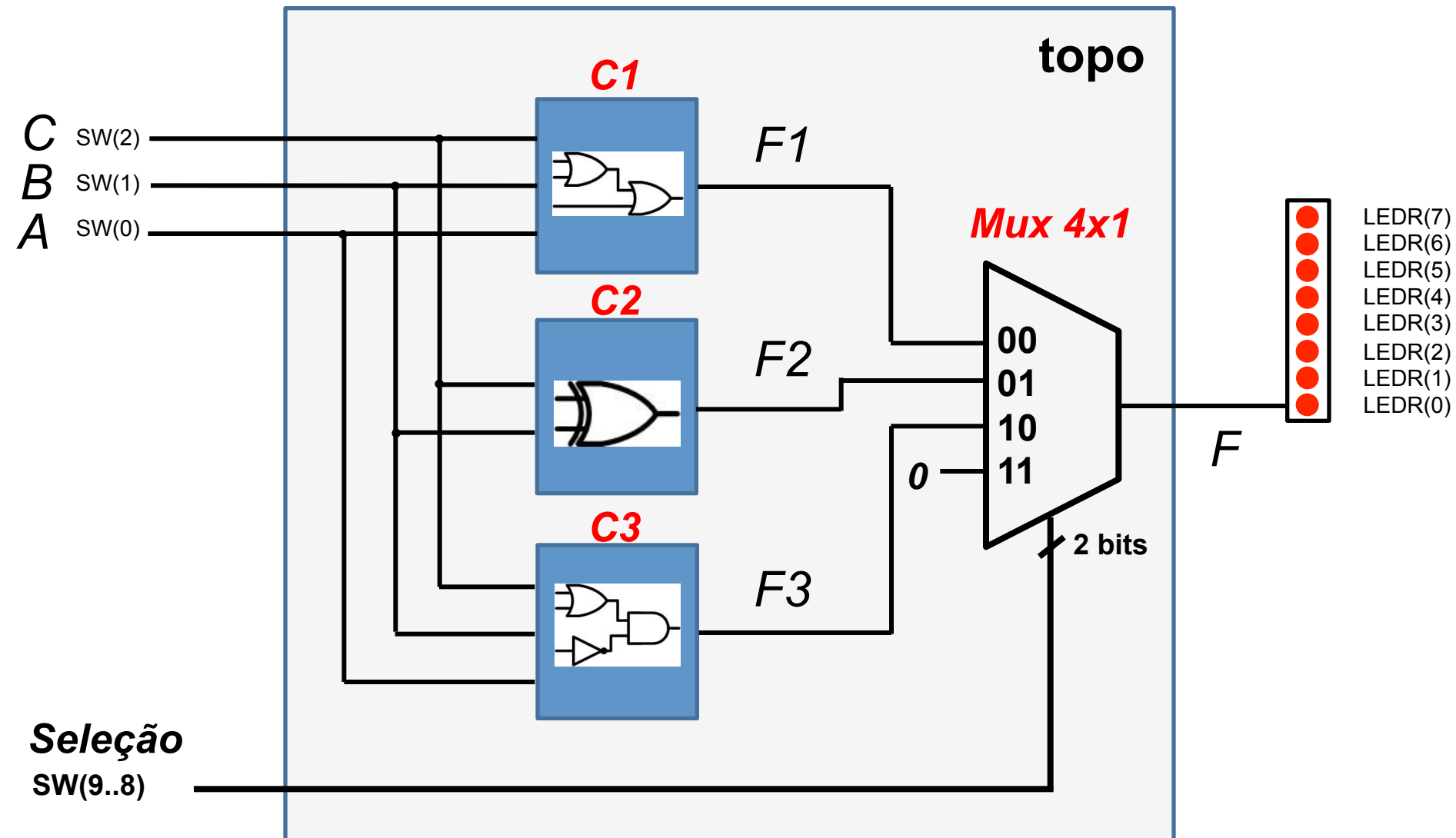
```
m <= w when s = "00" else
x when s = "01" else
y when s = "10" else
z;
```

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):

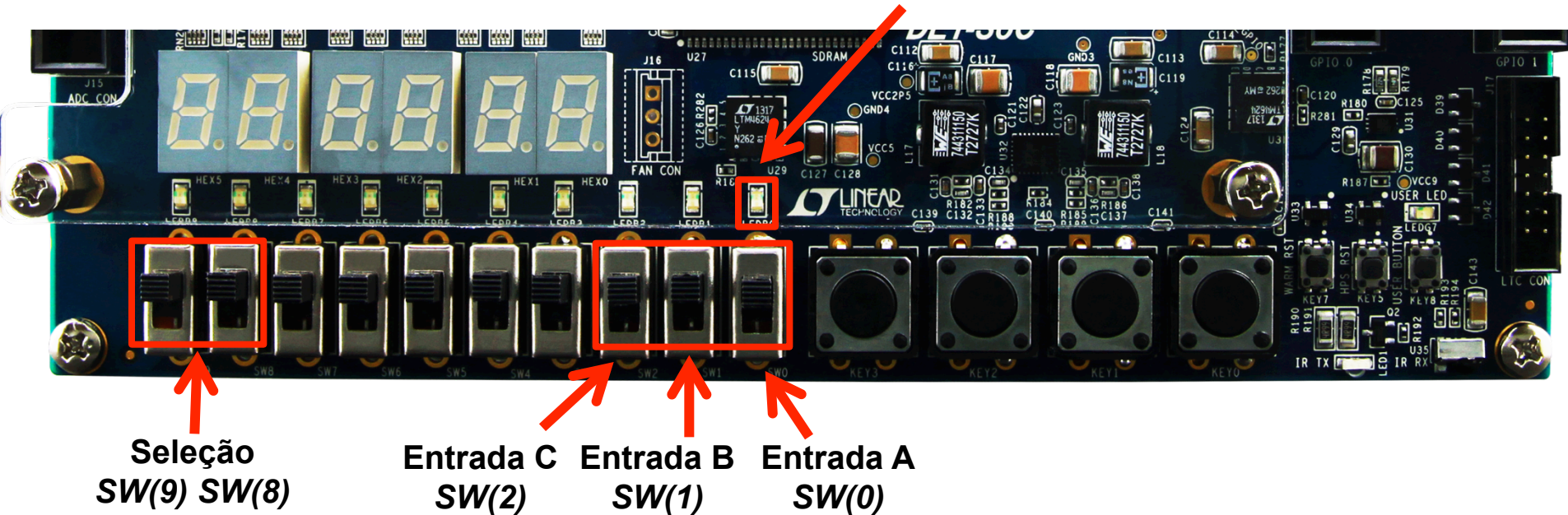


Descrição da tarefa: **PARTE I – VHDL estrutural**

- 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

Saída F – LEDR(0)



Seleção SW(9..8)	Saída LEDR(0)
00	F1
01	F2
10	F3
11	0 (LED apagado)

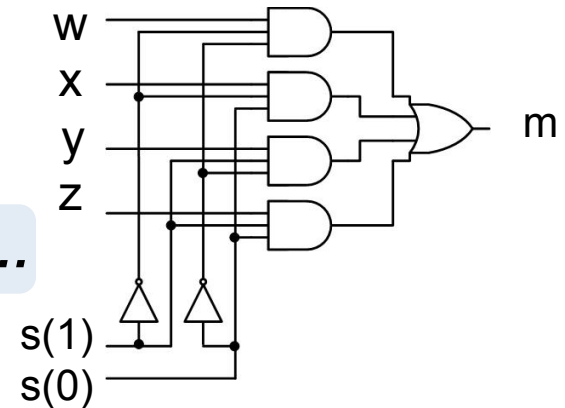
Descrição da tarefa: **PARTE II – VHDL comportamental**

- 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 no kit de desenvolvimento, e verificar se o novo circuito funciona de acordo com o esperado.

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

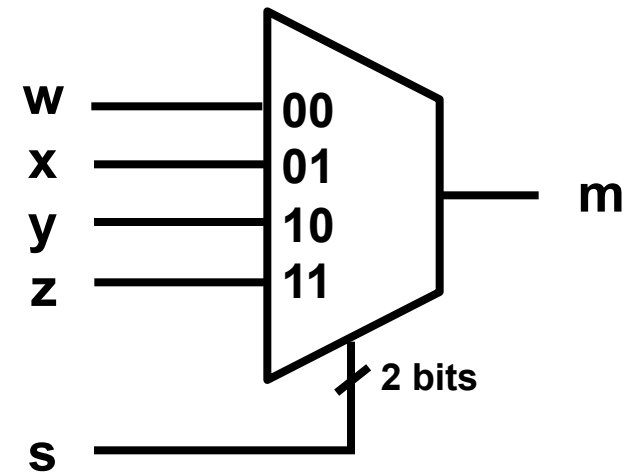
- **PARTE I** – Projetar e implementar o MUX em **VHDL**
estrutural:

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

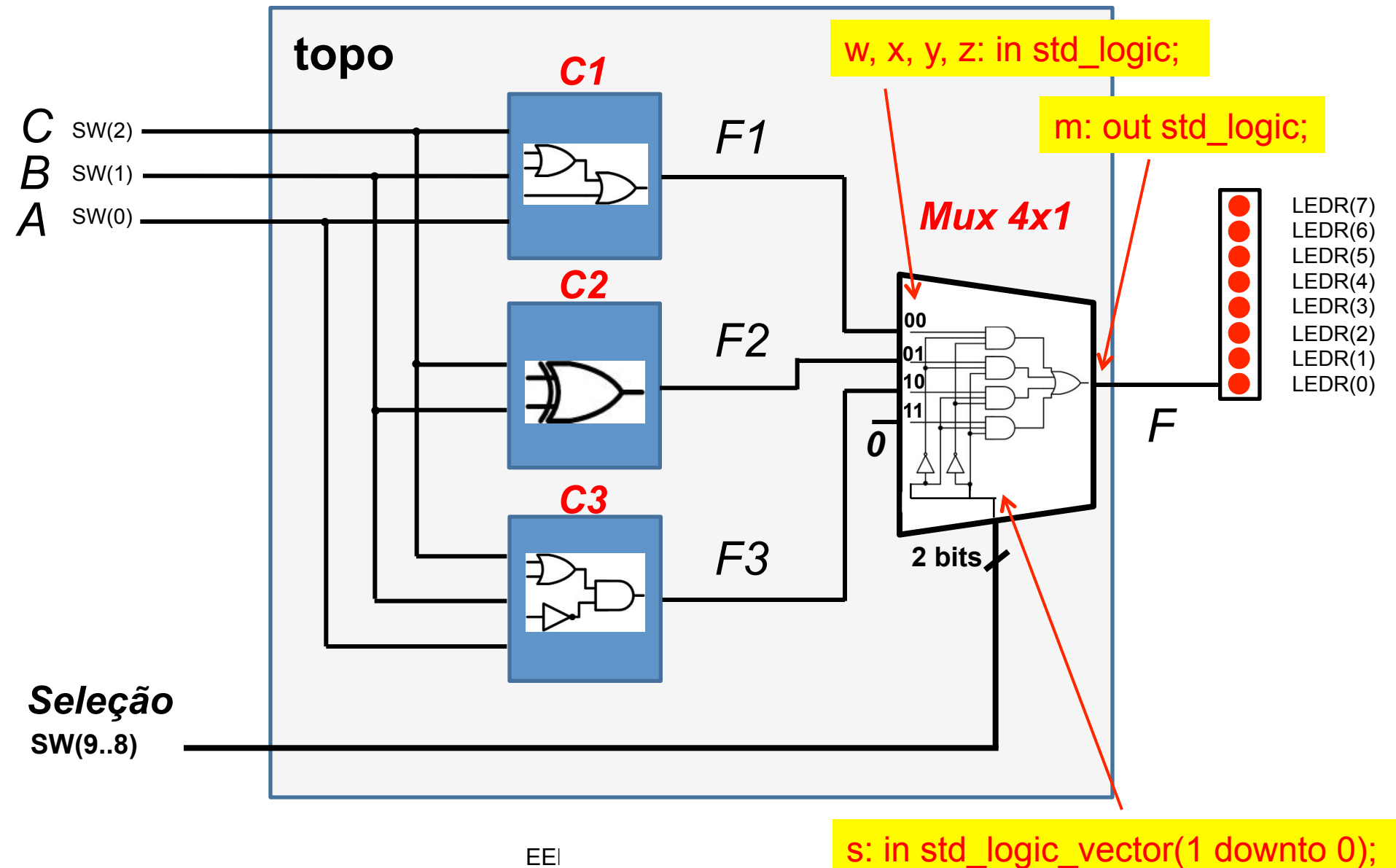


- **PARTE II** – Projetar e implementar o MUX em **VHDL**
comportamental:

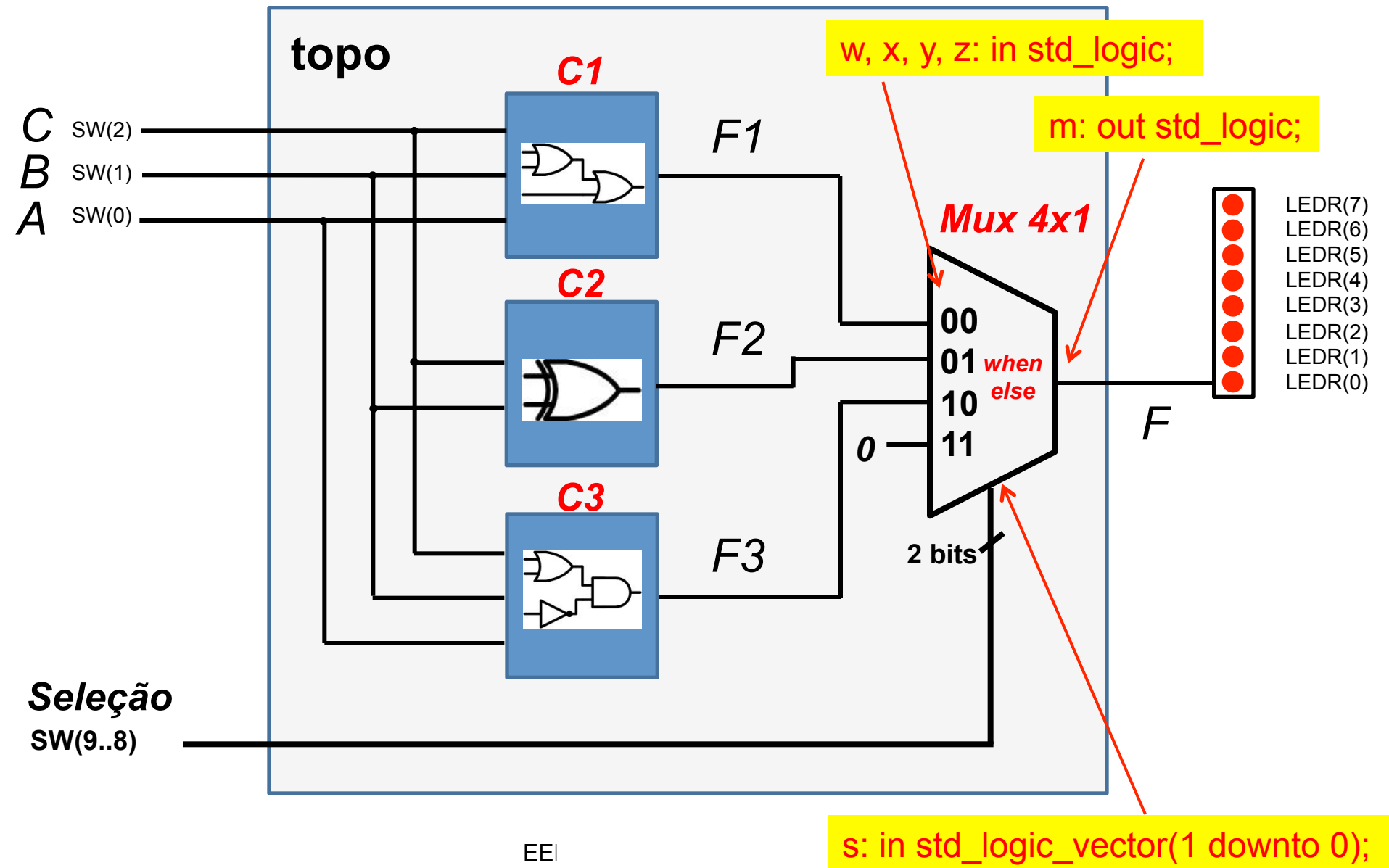
```
m <= w when s = "00" else  
    x when s = "01" else  
    y when s = "10" else  
    z;
```



PARTE I – VHDL estrutural



PARTE II – VHDL comportamental



Entradas		Saídas		
SW _{2..0} C B A	SW _{9..8} Seleção	F1 = A or B or C F2 = B xor C F3 = (B or C) and (not A)	Simulação Livro - <i>step</i> 5 LEDR ₀	FPGA Livro - <i>step</i> 6 LEDR ₀
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	0	