



Instituto Federal de Santa Catarina
Área de Telecomunicações

SST20707 – Síntese de Sistemas de Telecomunicações

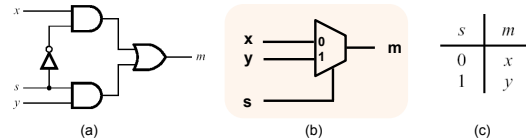
Prof. Roberto de Matos

roberto.matos@ifsc.edu.br

São José, setembro de 2013.

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 .

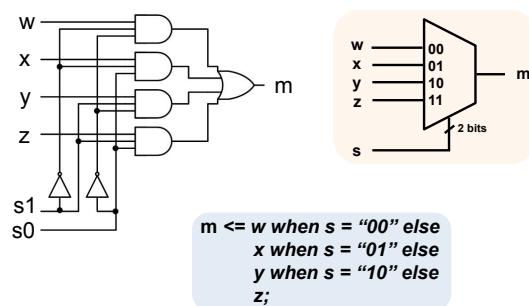


VHDL estrutural: $m \leq (NOT(s) AND x) OR (s AND y)$

VHDL comportamental: $m \leq x \text{ when } s = '0' \text{ else } y;$

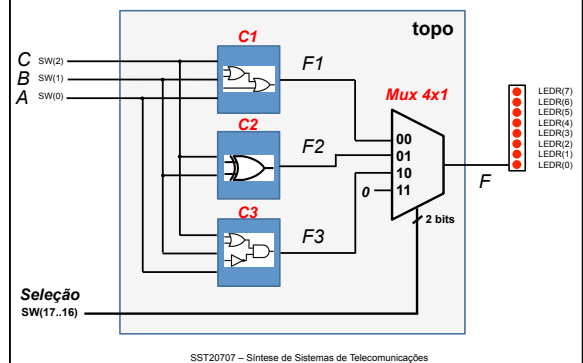
SST20707 – Síntese de Sistemas de Telecomunicações

Projeto de multiplexador - MUX 4x1



SST20707 – Síntese de Sistemas de Telecomunicações

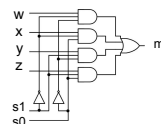
Reutilizar os arquivos do lab anterior, e realizar as alterações indicadas a seguir (Mux 4x1 no lugar de C4):



SST20707 – Síntese de Sistemas de Telecomunicações

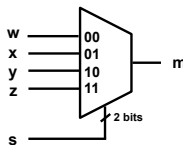
- PARTE I** – Projetar e implementar o MUX em VHDL estrutural:

$m \leq (w \text{ and } ((NOT(s1) AND (NOT(s0))) OR ...$

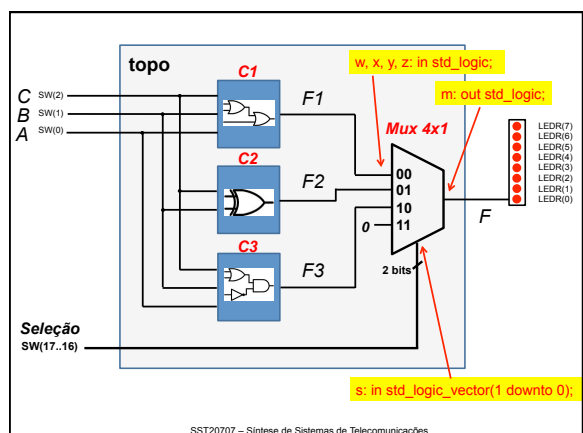


- PARTE II** – Projetar e implementar o MUX em VHDL comportamental:

$m \leq w \text{ when } s = "00" \text{ else } x \text{ when } s = "01" \text{ else } y \text{ when } s = "10" \text{ else } z;$



SST20707 – Síntese de Sistemas de Telecomunicações

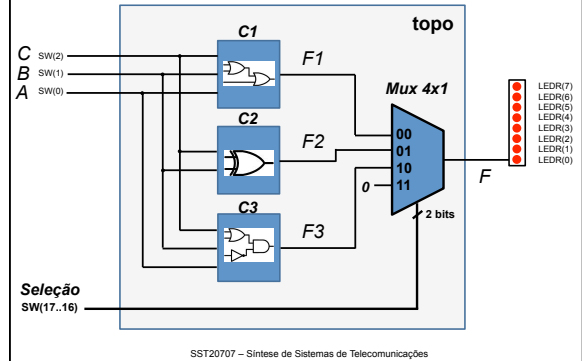


SST20707 – Síntese de Sistemas de Telecomunicações

Barramentos em VHDL - "std_logic_vector"

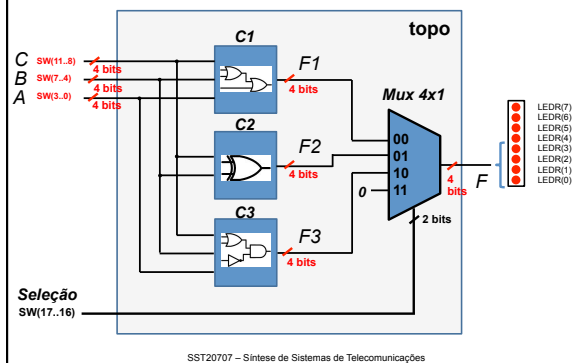
SST20707 – Síntese de Sistemas de Telecomunicações

No circuito do lab anterior, as entradas (A, B, C) e saídas (F) possuem o tamanho de 1 bit:



SST20707 – Síntese de Sistemas de Telecomunicações

Assumir o mesmo circuito, porém com entradas e saídas de 4 bits:



SST20707 – Síntese de Sistemas de Telecomunicações

Utilizando operandos de 4 bits

- Componente C1 realiza a operação $F1 = A \text{ or } B \text{ or } C$
- Componente C2 realiza a operação $F2 = B \text{ xor } C$
- Operandos de 1 bit - ex. $A = 0, B = 1, C = 0$
- Operandos de 4 bits - ex. $A = 0101, B = 1000, C = 0001$

1 bit		4 bits	
0	0	0101	
1	0	1000	1000
0	1	0001	0001
$F1 = 1$	$F2 = 1$	$F1 = 1101$	$F2 = 1001$

SST20707 – Síntese de Sistemas de Telecomunicações

Implementação de barramento em VHDL

std_logic → fio

std_logic_vector → vários fios (barramento)

- std_logic_vector** é utilizado para implementar um "vetor" de sinais (vários fios, barramento).
- Por exemplo, a saída F deverá ser definida na **entity** como:

F: out std_logic_vector(3 downto 0);

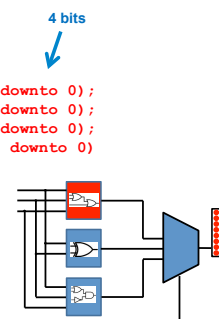
SST20707 – Síntese de Sistemas de Telecomunicações

Componente C1

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

entity C1 is
port (A: in std_logic_vector(3 downto 0);
      B: in std_logic_vector(3 downto 0);
      C: in std_logic_vector(3 downto 0);
      F: out std_logic_vector(3 downto 0);
);
end C1;

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



SST20707 – Síntese de Sistemas de Telecomunicações

Componente C2

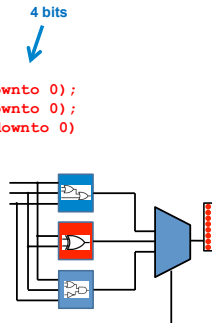
```

library IEEE;
use IEEE.Std_Logic_1164.all;

entity C2 is
port (A: in std_logic_vector(3 downto 0);
      B: in std_logic_vector(3 downto 0);
      F: out std_logic_vector(3 downto 0)
);
end C2;

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

```



SST20707 – Síntese de Sistemas de Telecomunicações

Componente C3

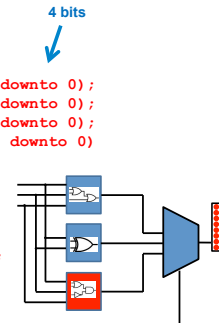
```

library IEEE;
use IEEE.Std_Logic_1164.all;

entity C3 is
port (A: in std_logic_vector(3 downto 0);
      B: in std_logic_vector(3 downto 0);
      C: in std_logic_vector(3 downto 0);
      F: out std_logic_vector(3 downto 0)
);
end C3;

architecture c3_estr of C3 is
begin
    -- ver lab. sobre componentes
end c3_estr;

```



SST20707 – Síntese de Sistemas de Telecomunicações

Componente Mux

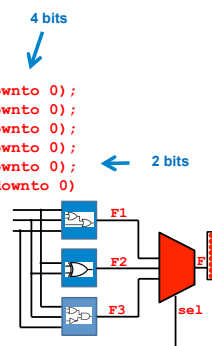
```

library IEEE;
use IEEE.Std_Logic_1164.all;

entity Mux is
port (w: in std_logic_vector(3 downto 0);
      x: in std_logic_vector(3 downto 0);
      y: in std_logic_vector(3 downto 0);
      z: in std_logic_vector(3 downto 0);
      s: in std_logic_vector(1 downto 0);
      m: out std_logic_vector(3 downto 0)
);
end Mux;

architecture mux_bhv of Mux is
begin
    -- ver lab. sobre Mux
end mux_bhv;

```



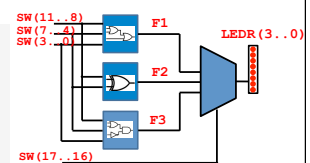
SST20707 – Síntese de Sistemas de Telecomunicações

Componente Topo

```

library ieee;
use ieee.std_logic_1164.all;
entity topo is
port (SW: in STD_LOGIC_VECTOR(17 downto 0);
      LEDR: out STD_LOGIC_VECTOR(17 downto 0)
);
end topo;
architecture topo_estr of topo is
    signal F1, F2, F3: std_logic_vector(3 downto 0);
    component C1
    port (A: in std_logic_vector(3 downto 0);
          B: in std_logic_vector(3 downto 0);
          C: in std_logic_vector(3 downto 0);
          F: out std_logic_vector(3 downto 0));
    end component;
    component C2
    port (A: in std_logic_vector(3 downto 0);
          B: in std_logic_vector(3 downto 0);
          F: out std_logic_vector(3 downto 0));
    end component;
    component C3
    port (A: in std_logic_vector(3 downto 0);
          B: in std_logic_vector(3 downto 0);
          C: in std_logic_vector(3 downto 0);
          F: out std_logic_vector(3 downto 0));
    end component;
begin
    -- INCLUIR AQUI O Mux
end topo_estr;

```



```

begin
    L0: C1 port map (SW(3 downto 0),
                    SW(7 downto 4), SW(11 downto 8), F1);
    L1: C2 port map (SW(3 downto 0),
                    SW(7 downto 4), SW(11 downto 8), F2);
    L2: C3 port map (SW(3 downto 0),
                    SW(7 downto 4), SW(11 downto 8), F3);
    L3: Mux port map (F1, F2, F3,
                    SW(17 downto 16), LEDR(3 downto 0));
end topo_estr; -- END da architecture

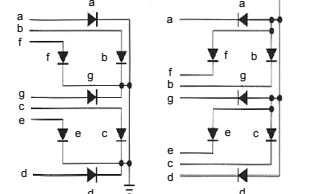
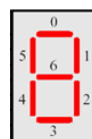
```

Display de 7-segmentos

• Um display de 7-segmentos é composto por sete LEDs que podem ser ligados ou desligados de forma independente.

• **Catodo comum** - terminais catodo dos LEDs estão conectados a GND, e cada LED é ligado ao conectar seu anodo em Vcc.

• **Anodo comum** - terminais anodo dos LEDs estão conectados a Vcc, e cada LED é ligado ao conectar seu catodo em GND.



Catodo comum

Anodo Comum

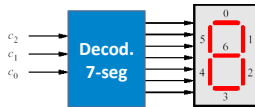
SST20707 – Síntese de Sistemas de Telecomunicações

Decodificadores em VHDL

SST20707 – Síntese de Sistemas de Telecomunicações

Decodificador de binário para 7-segmentos

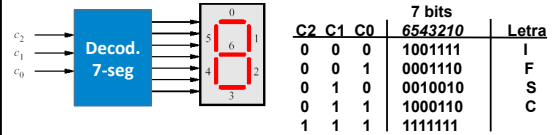
- Placa DE2 possui 7 displays de 7-segmentos, todos do tipo anodo comum (LEDs acendem com GND, ou zero lógico).
- Para escrever um valor binário em um dos displays, é preciso realizar uma conversão do código binário para o código 7-segmentos.
- Os 3 bits de entrada do circuito "Decod. 7-seg" são *decodificados*, e a palavra de 7 bits gerada é enviada para o display de 7 segmentos.



SST20707 – Síntese de Sistemas de Telecomunicações

Projeto de decodificador de binário para 7-segmentos

- Exemplos de valores em binários convertidos (decodificados) para 7-segmentos, visando escrita no display da placa DE2 anodo comum)



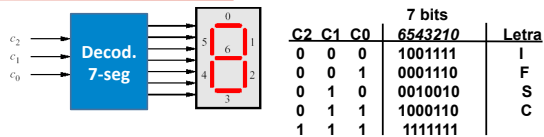
- Nesse exemplo, ao receber "000" na entrada, o decodificador gera o código equivalente ao acendimento da letra "I" no display 7-seg.

- Ao receber "111", todos os segmentos são desligados.

- Notar que por ser do tipo anodo comum, um "0" liga um segmento.

SST20707 – Síntese de Sistemas de Telecomunicações

Projeto de decodificador "binário para 7-segmentos"



- Um circuito para implementar a lógica do decodificador em questão poderia ser projetado utilizando **vários métodos**:

- Soma de produtos:**

$$F(0) = C2' C1' C0' + C2 C1 C0$$

$$F(1) = C2' C1' C0 + C2' C1 C0' + C2' C1 C0 + C2 C1 C0$$

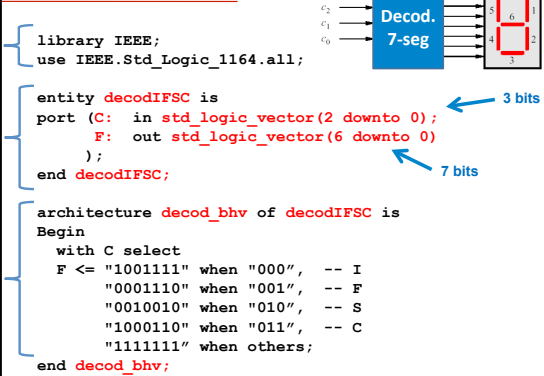
$$F(2) = \dots$$

- Análise comportamental:**

F = "1000001" quando C2C1C0 = "000" senão
 "0001110" quando C2C1C0 = "001" senão
 ... senão
 "1111111"

SST20707 – Síntese de Sistemas de Telecomunicações

Componente Decod_IFSC



SST20707 – Síntese de Sistemas de Telecomunicações

