



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



“EEL7020 – Sistemas Digitais”

Prof. Eduardo Augusto Bezerra

Eduardo.Bezerra@eel.ufsc.br

Florianópolis, agosto de 2011.

Sistemas Digitais

“Prática: Chaves, Luzes e Multiplexadores”

Arquivo: lab1_VHDL.pdf
parte I e parte II

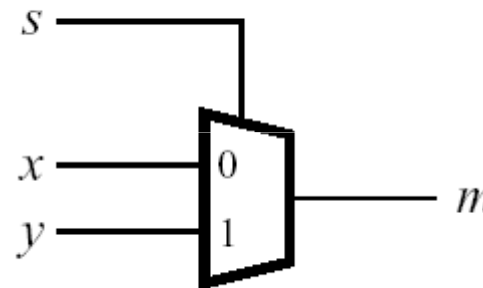
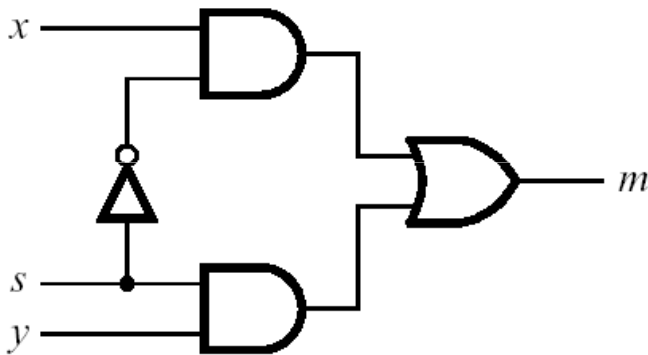
Chaves e Luzes - Revisão

- Código VHDL para “leitura” das chaves e “escrita” nos LEDs

```
library ieee;
use ieee.std_logic_1164.all;
entity part1 is
    port ( SW      : in std_logic_vector(17 downto 0);
          LEDR     : out std_logic_vector(17 downto 0)
        );
end part1;
architecture behavior of part1 is
begin
    LEDR <= SW;
end behavior;
```

Projeto de multiplexador

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



s	m
0	x
1	y

VHDL estrutural:

```
 $m \leq (NOT(s) AND x) OR (s AND y);$ 
```

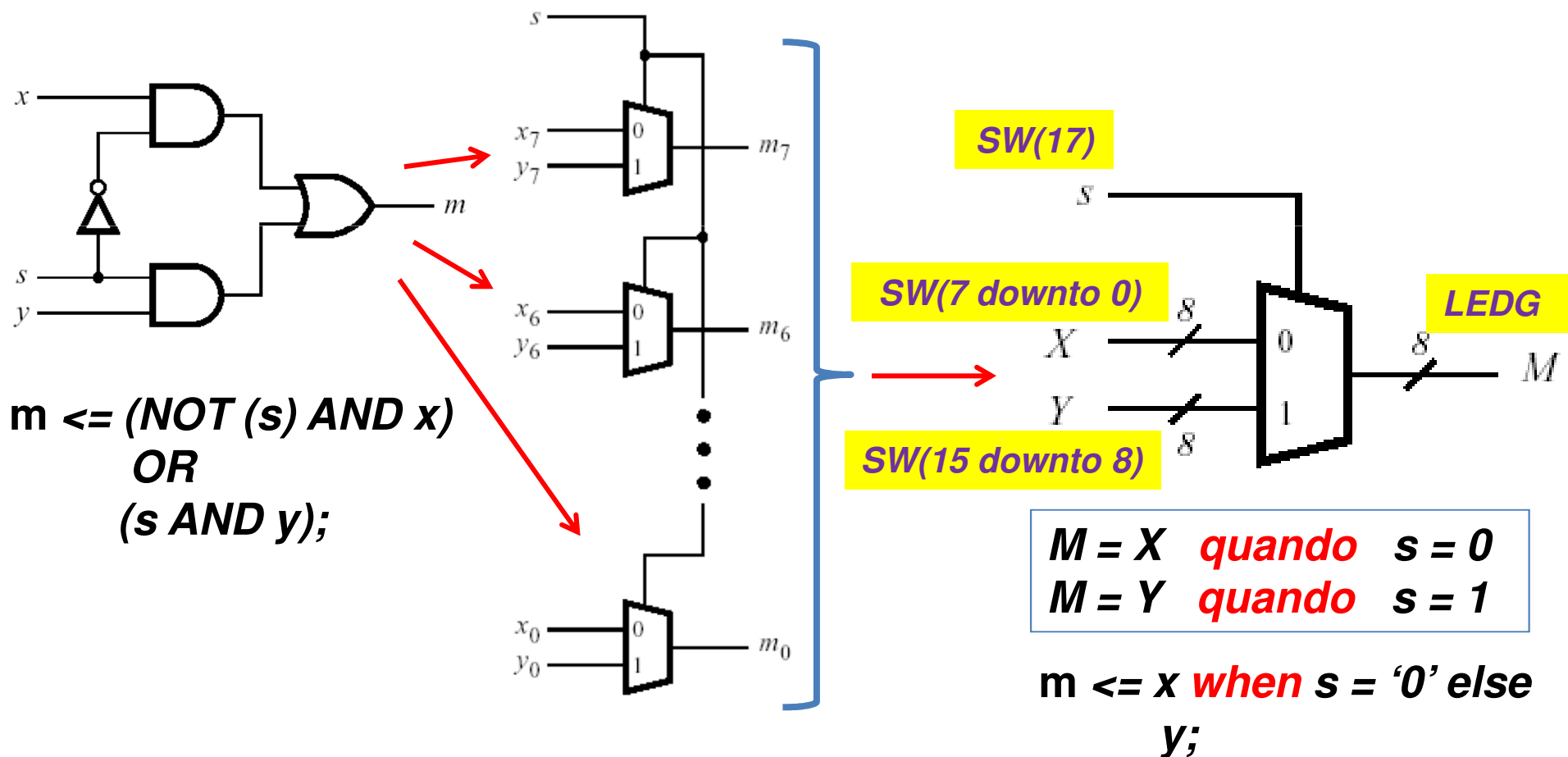
VHDL comportamental:

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

Tarefa a ser realizada na aula prática

Projeto de multiplexador

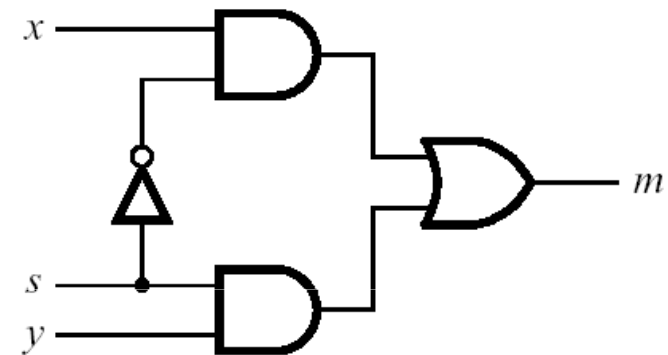
Projetar um circuito em VHDL com duas entradas de 8 bits, X e Y, e produzir uma saída M de 8 bits. Se $s = 0$, $M = X$. Se $S = 1$, $M = Y$.



Projeto de multiplexador

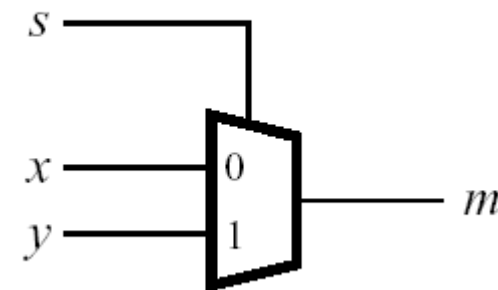
- **PARTE I** – Projetar e implementar o circuito em VHDL estrutural:

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



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

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



Projeto de multiplexador

Algumas dicas

- **std_logic** → fio e **std_logic_vector** → vários fios (barramento)
- As entradas e a saída do multiplexador devem possuir 8 bits. Assim, será necessário usar um “vetor” de sinais (vários fios, barramento).
- Por exemplo, a saída M deverá ser definida na **entity** como:

M: out **std_logic_vector**(7 downto 0);

- A entidade (**entity**) deverá conter as declarações de todos os sinais de entrada e saída do multiplexador.

Projeto de multiplexador: etapas

1. Criar um novo projeto no Quartus II
2. Criar o VHDL para o mux 2x1 com entradas/saída de 8 bits
3. Usar a chave SW₁₇ como entrada S (selector)
4. Usar as chaves SW₇₋₀ para a entrada X e as chaves SW₁₅₋₈ para a entrada Y
5. Conectar as chaves SW (todas) aos LEDs vermelhos (LEDR) e a saída M do mux aos LEDs verdes LEDG₇₋₀
3. Incluir as atribuições de pinos no projeto (arquivo *DE2_pin_assignments.csv*)
4. Compilar o projeto (síntese)
5. Realizar o download para o FPGA, e testar o multiplexador usando as chaves, e observando os LEDs.

Tarefa adicional

Tarefa adicional – mais multiplexadores

Projetar um circuito em VHDL, com multiplexadores, para controlar o acendimento dos LEDs **vermelhos** e **verdes** de acordo com os seguintes comandos (chaves SW):

SW ₁	SW ₀	LEDG _{3..0}	LEDG _{7..4}	LEDR pares	LEDR ímpares
OFF	OFF	ON	OFF	OFF	OFF
OFF	ON	OFF	ON	OFF	OFF
ON	OFF	OFF	OFF	ON	OFF
ON	ON	OFF	OFF	OFF	ON

Simulação com *ModelSim*

Simulação do projeto com ModelSim

1. Criar uma nova pasta dentro da pasta do projeto.
2. Copiar os **scripts de simulação** disponíveis na página da disciplina para dentro da nova pasta.
3. Entrar na nova pasta, editar o arquivo "*compila.do*", e alterar ***mux.vhd*** para o nome do seu arquivo VHDL a ser simulado.
4. Copiar APENAS o seu arquivo VHDL (multiplexador) a ser simulado para a nova pasta, que já deve possuir os *scripts* de simulação copiados da página da disciplina.
5. Executar o *ModelSim-Altera*, que se encontra no menu *Iniciar* do Windows, pasta "*Altera*".
6. No menu "*File*" do *ModelSim*, definir a pasta do projeto (opção "*Change Directory*"), selecionando a nova pasta.

Simulação do projeto com ModelSim (cont.)

7. Execução da simulação (arquivo *compila.do*)
 - a) No menu "Tools" do ModelSim, selecionar "*Tcl*" -> "*Execute Macro*".
 - b) Selecionar o arquivo "*compila.do*", e "*Open*".
8. O *ModelSim* irá compilar os arquivos VHDL e iniciar a simulação.
9. A janela com as formas de onda irá abrir, apresentando o resultado da simulação.

Obs. Se desejar, editar o arquivo *tb.vhd* para alterar a simulação a ser realizada, e repetir o passo 7.

Simulação do projeto com ModelSim (cont.)

Obs. Se o resultado da simulação não estiver de acordo com o esperado, alterar o seu VHDL, salvar, e executar novamente a simulação (arquivo *compila.do*).

Obs. A simulação só irá funcionar se o seu projeto possuir EXATAMENTE a seguinte *entity*:

```
entity mux IS
```

```
    PORT (SW   : IN   STD_LOGIC_VECTOR(17 DOWNT0 0);
```

```
          LEDG: OUT STD_LOGIC_VECTOR(7 DOWNT0 0)
```

```
          LEDR: OUT STD_LOGIC_VECTOR(17 DOWNT0 0)
```

```
    );
```

```
end mux;
```

Simulação do projeto com ModelSim (cont.)

Obs. O arquivo "*compila.do*" contém os comandos do *ModelSim* necessários para realizar a simulação, incluindo:

- a) Criação da biblioteca de trabalho - comando *vlib*.
- b) Compilação dos arquivos VHDL para a biblioteca de trabalho - comando *vcom*.
- c) Inicialização do simulador com o arquivo *testbench* - comando *vsim*.
- d) Execução da janela de formas de onda (*waveform*) - comando *wave*.
- e) Adição dos sinais na janela de formas de onda - comando *wave*.
- f) Execução da simulação - comando *run*.