



Aula 05 – Comandos Concorrentes básicos



Tópicos da aula

- **Concorrência em VHDL**
- **Comandos concorrentes básicos**



Concorrência em VHDL

- ❏ VHDL é uma linguagem de **síntese** de circuitos digitais
- ❏ Ela não pode ser entendida como as linguagens de programação para processadores (C, Java, etc.)
- ❏ Cada linha de programa em VHDL representa um **circuito digital independente**
- ❏ Pode-se pensar um programa VHDL como um **protoboard**, onde componentes digitais são conectados entre si.



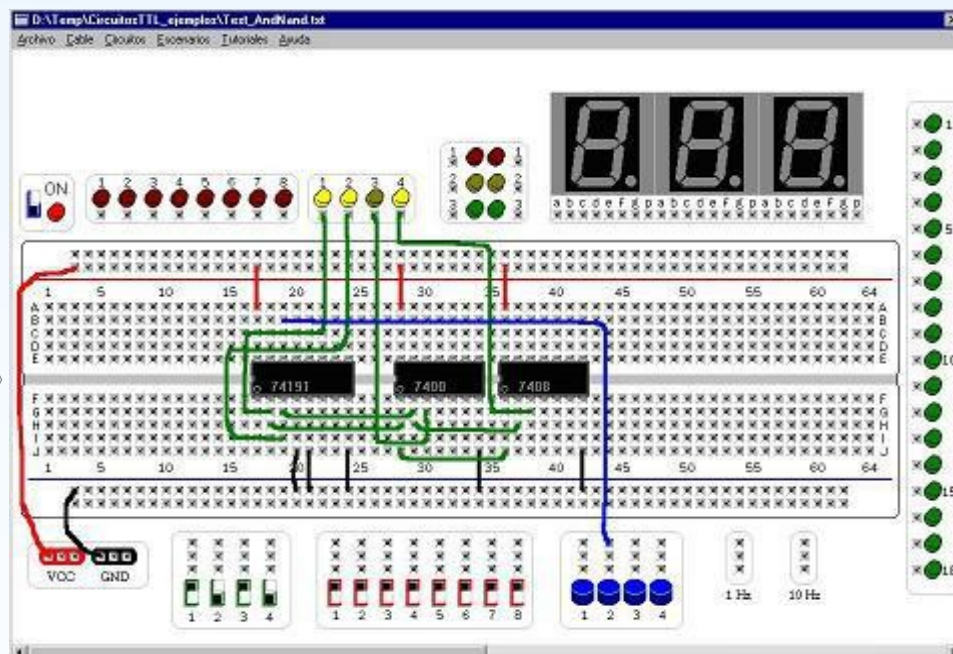
Universidade Federal
de Santa Catarina

Linguagem de descrição de hardware

Aula V: 4 de 12

Concorrência em VHDL

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5  use IEEE.STD_LOGIC_ARITH.ALL;
6
7  ENTITY PORTA_AND IS
8
9  PORT (
10      A      : IN STD_LOGIC;
11      B      : IN STD_LOGIC;
12      C      : OUT STD_LOGIC;
13      D      : OUT STD_LOGIC;
14      E      : OUT STD_LOGIC
15  );
16  END PORTA_AND;
17
18  ARCHITECTURE behavioral OF PORTA_AND IS
19
20  BEGIN
21
22      C <= A and B;
23      D <= (not A) and B;
24      E <= A or B;
25
26  END behavioral;
```





Comando **WHEN ELSE**

- Transferência condicional de um sinal
- **Contém:** uma lista de condições e expressões
- **Primeira condição verdadeira:** define expressão transferida
- **Formato da construção:**

```
sinal_destino <= expressao_a WHEN condicao_1 ELSE  
                    expressao_b WHEN condicao_2 ELSE  
                    expressao_c;
```



Comando **WHEN ELSE**

```
b <= "1000" when a = "00" else  
      "0100" when a = "01" else  
      "0010" when a = "10" else  
      "0001" when a = "11";
```



Exercícios

- ❏ **Implemente um MULTIPLEXADOR 4x1 utilizando os comandos WHEN ELSE**



Construção **WITH SELECT**

- Transferência condicional de um sinal
- Contém: uma lista de opções
- Todas as condições da expressão de escolha devem ser consideradas
 - não existe uma prioridade como na construção **WHEN ELSE**
- As opções podem ser agrupadas: o caractere **|** equivale a “ou”
 - **TO** e **DOWNTO** delimitam faixas de opções
- Opções restantes: palavra reservada **OTHERS**
- Formato da construção:

```
WITH expressao_escolha SELECT      -- expressao_escolha =
  sinal_dest <= expr_a WHEN condicao_1,      -- condicao_1
  expr_b WHEN condicao_2,                  -- condicao_2
  expr_c WHEN condicao_3 | condicao_4,        -- condicao_3 ou condicao_4
  expr_d WHEN condicao_5 TO condicao_7,      -- condicao_5 ate condicao_7
  expr_e WHEN OTHERS;                    -- condicoes restantes
```




Construção **WITH SELECT**

```
with a select b <=
    "1000" when "00",
    "0100" when "01",
    "0010" when "10",
    "0001" when "11";
```



Tipos de modelagem

❑ Estrutural

❑ “O que é”

```
ENTITY mux_2x1 IS
PORT (a, b : IN BIT;
      sel : IN BIT;
      s : OUT BIT);
END mux;

ARCHITECTURE structural OF mux_2x1 IS
BEGIN
    s <= (a AND NOT sel) OR
          (b AND sel);
END structural;
```

❑ Comportamental

❑ “O que faz”

```
ENTITY mux_2x1 IS
PORT (a, b : IN BIT;
      sel : IN BIT;
      s : OUT BIT);
END mux;

ARCHITECTURE behavior OF mux_2x1 IS
BEGIN
    PROCESS(a,b,sel)
    BEGIN
        IF (sel='0') THEN
            s <= a;
        ELSE
            s <= b;
        END IF;
    END PROCESS;
END behavior;
```



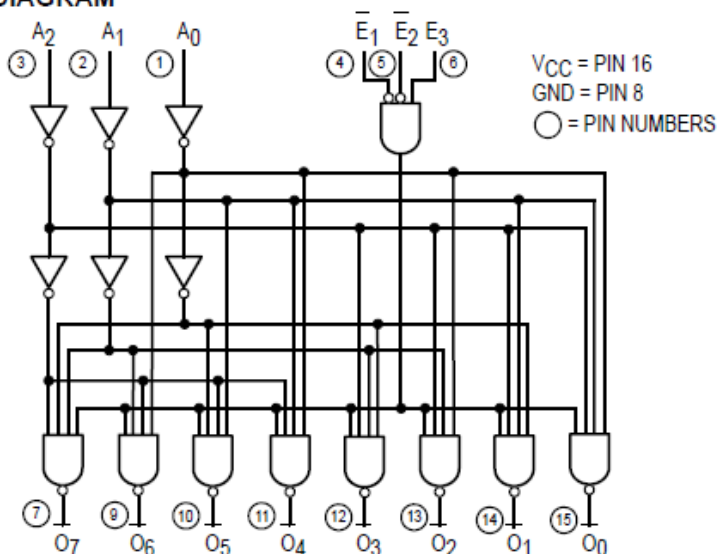
Exercícios

- ❏ **Implemente um MULTIPLEXADOR 8x1 utilizando os comandos WITH SELECT**

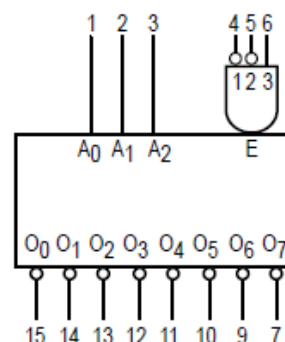


Exercícios

LOGIC DIAGRAM



LOGIC SYMBOL



TRUTH TABLE

INPUTS						OUTPUTS							
E ₁	E ₂	E ₃	A ₀	A ₁	A ₂	O ₀	O ₁	O ₂	O ₃	O ₄	O ₅	O ₆	O ₇
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	L	H	H	H	H
L	L	H	H	L	H	H	H	H	H	L	H	H	H
L	L	H	L	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care



Comando **PROCESS**

- **Objetivo:** delimitar regiões de código sequencial
- **Início:** palavra reservada **PROCESS**
- **Lista de sensibilidade:** identifica que sinais ativam a execução do processo
- **Comandos sequenciais:** próximo capítulo

```
abc: PROCESS (lista de sensibilidade)
  BEGIN
    comando_1;
    comando_2;
    ..
    comando_n;
END PROCESS abc;

def: PROCESS (lista de sensibilidade)
  BEGIN
    comando_1;
    comando_2;
    ..
    comando_n;
END PROCESS def;
```



Universidade Federal
de Santa Catarina

Linguagem de descrição de hardware

Aula V: 14 de 12

Exemplo

```
22  
23 ☐ PROCESS (A, B)  
24 BEGIN  
25     C <= A and B;  
26     D <= not F;  
27 END PROCESS;  
28
```



FIM AULA V