SERVIÇO PÚBLICO FEDERAL
UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ - UNIFESSPA
INSTITUTO DE GEOCIÊNCIAS E ENGENHARIAS - IGE
FACULDADE DE COMPUTAÇÃO E ENG. ELÉTRICA – FACEEL
CURSO ENGENHARIA DE COMPUTAÇÃO

Sistemas Embarcados

T-2018

Prof. José Carlos Da Silva
jcdsilv@hotmail.com
jose-carlos.silva@unifesspa.edu.br
whatsApp: 19-993960156

Junho/2021

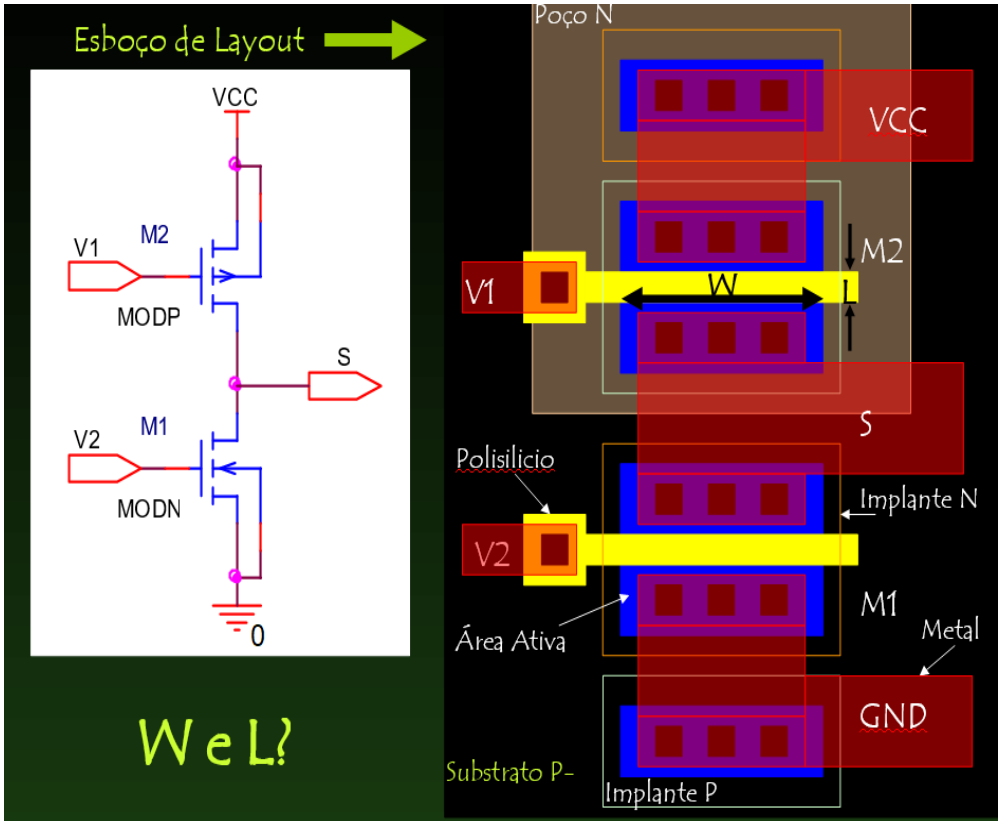
Conteúdo

- Introdução a VHDL;
- Estruturas do código VHDL;
- Bibliotecas e pacotes fundamentais;
- Tipos de dados predefinidos;
- Objetos (Constant, Signal, Variable, File);
- Tipos de dados definidos pelo usuário
- Operadores;
- Atributos;
- Código concorrente versus sequencial;
- Código concorrente (WHEN, SELECT, GENERATE);
- Código sequencial (PROCESS, IF, CASE, LOOP, WAIT);
- Instruções auxiliares (ASSERT, ALIAS);
- Pacotes (PACKAGE);
- Componentes (COMPONENT);
- Funções (FUNCTION);
- Procedimentos (Procedure);
- VHDL para maquinas de estados;
- VHDL 2008;
- Introdução a ferramenta de síntese e simulação quartus II;
- Exercícios (Atividades e trabalhos).

INTRODUÇÃO

O **código** descreve o **comportamento** ou **estrutura** desejada,
a partir do qual um **circuito físico** correspondente é **inferido (deduzido)** pelo **compilador**

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY Mux_1b IS
PORT (
    D0, D1, Sinal : IN  std_logic;
    Saida         : OUT std_logic
  );
END Mux_1b;

ARCHITECTURE behavior_we OF Mux_1b IS
BEGIN
    Saida <=   D0 WHEN Sinal = '0' ELSE
               D1 WHEN Sinal = '1';
END behavior;
```



INTRODUÇÃO

O que significa VHDL?

Very High Speed Integrated Circuit
Hardware
Description
Language

Linguagem de Descrição de Hardware com ênfase em Circuitos Integrados de altíssima velocidade.

INTRODUÇÃO

O que significa HDL?

- Uma **Linguagem de Descrição de Hardware** descreve o que um sistema faz e como;

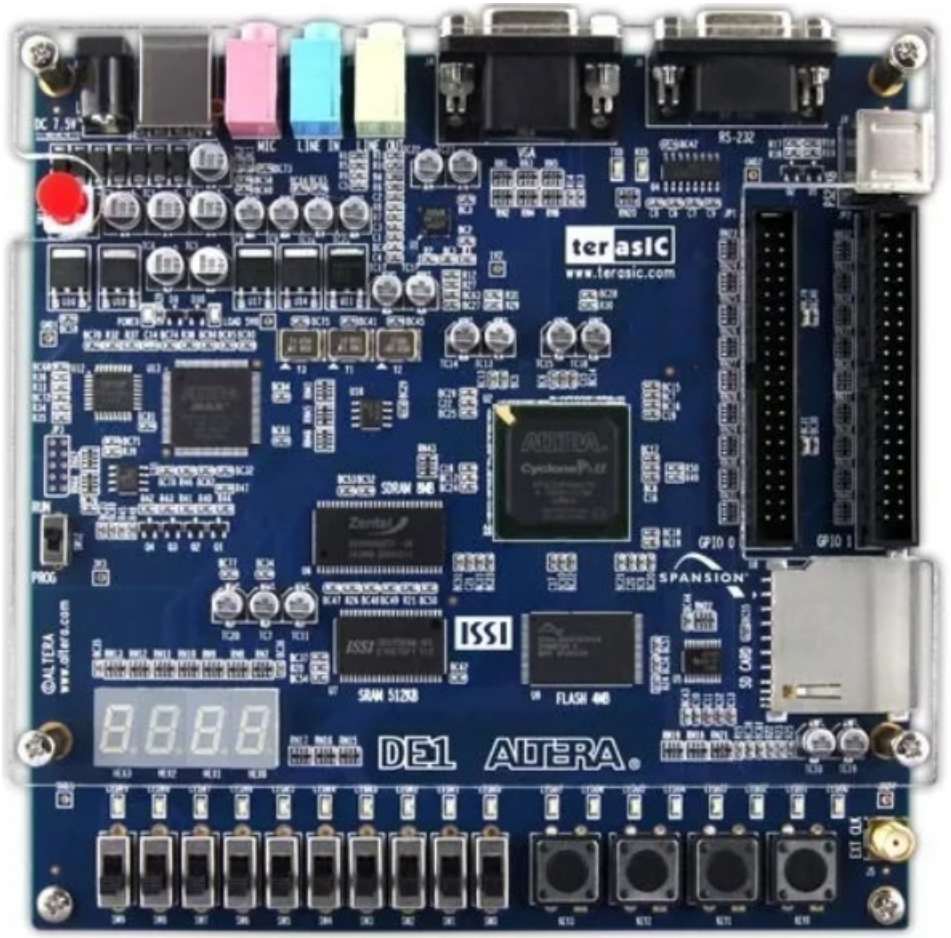
- Um **sistema** descrito em linguagem de hardware pode ser **implementado** em um dispositivo programável **FPGA** (Field Programmable Gate Array) ou um dispositivo ASIC (Aplication Specific Integrated Circuit), permitindo o uso em campo do sistema;

Existem dezenas de HDLs:

- AHDL, VERILOG, Handel-C, SDL, ISP, ABEL …

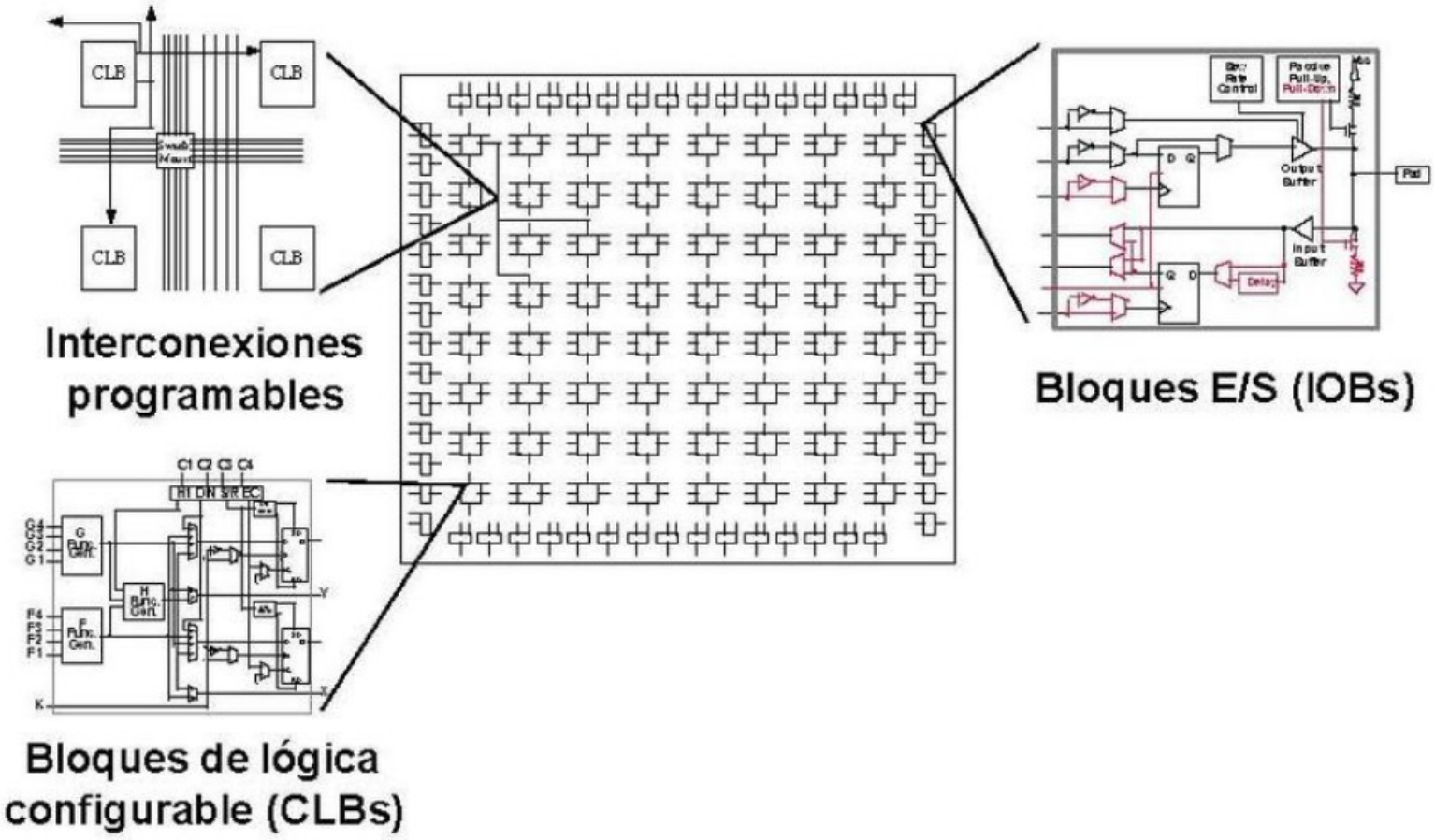
INTRODUÇÃO

FPGA ((Field Programmable Gate Array)



INTRODUÇÃO

FPGA ((Field Programmable Gate Array)



INTRODUÇÃO

Características do VHDL:

- Linguagem concorrente: Todos os comandos ocorrem simultaneamente (com exceção de processos);

- Permite, através de simulação, verificar o comportamento do sistema digital;

- Permite descrever hardware em diversos níveis de abstração, por exemplo:
    - Algorítmico ou comportamental;
    - Transferência entre registradores (RTL).

Hoje utilizada para SIMULAÇÃO e SÍNTESE

INTRODUÇÃO

	Linguagem de Programação	**VHDL**
PROPÓSITO	SOFTWARE	HARDWARE
ENTRADAS	TEXTO ou FERRAMENTAS VISUAIS	
DESENVOLVIMENTO	COMPILAÇÃO	COMPILAÇÃO PARA SIMULAÇÃO E SÍNTESE EM HARDWARE
DEPURAÇÃO	EXECUÇÃO E VISUALIZAÇÃO DOS RESULTADOS	SIMULAÇÃO E VISUALIZAÇÃO DE FORMAS DE ONDA
INSTRUÇÃO	SÓ SEQUENCIAIS	CONCORRENTES E SEQUENCIAIS

INTRODUÇÃO

Vantagens:

- **Time-to-market:** Se há dez anos atrás um produto demorava 6 meses para ser desenvolvido, mas permanecia no mercado por 2 anos, hoje um produto não permanece mais de 18 meses logo o seu desenvolvimento deve levar bem menos tempo;

- **Menor ciclo e custo de desenvolvimento:** devido à eliminação de geração, manutenção de esquemáticos e pela diminuição de erros de desenvolvimento pelo uso de simulação nos ciclos iniciais do projeto;

- **Aumento de qualidade no desenvolvimento:** VHDL facilita o rápido experimento com diferentes arquiteturas e técnicas de implementação, e pela capacidade das ferramentas de síntese otimizarem um projeto tanto para área mínima quanto para velocidade máxima;

INTRODUÇÃO

Vantagens:

- **Evolução da tecnologia:** Novos dispositivos surgem com mais capacidade e mais recursos internos;

- **Gerenciamento do projeto –** Projetos em VHDL facilitam a estruturação de componentes (top-down), facilitam a documentação e são necessárias menos pessoas para desenvolver e verificar sendo também mais simples modificar o projeto;

- **Independente de tecnologia e fabricante:** Porém sabe-se que na prática não é independente de ferramenta de síntese e de simulação.

INTRODUÇÃO

Conceitos Necessários:

Algoritmos;

- Conceitos de linguagem de programação (para descrição comportamental);
- Circuitos Digitais;
- Arquitetura de computadores.
- Para descrições mais complexas:
  - Linguagem Assembly;
  - Microprocessadores;
  - Sistemas embarcados.

INTRODUÇÃO

Ciclos de projetos:

- **Especificação:** Determinar requisitos e funcionalidade do projeto;
- **Codificação:** Descrever em VHDL todo o projeto, segundo padrões de sintaxe;
- **Simulação do Código-Fonte:** Simular o código em ferramenta confiável a fim de verificar preliminarmente cumprimento da especificação.

INTRODUÇÃO

Ciclos de projetos:

- **Síntese:** Compilação de um código VHDL para uma descrição abstrata;
- **Otimização:** Seleção da melhor solução de implementação para uma dada tecnologia;
- **Fitting:** Lógica sintetizada e otimizada mapeada nos recursos oferecidos pela tecnologia.

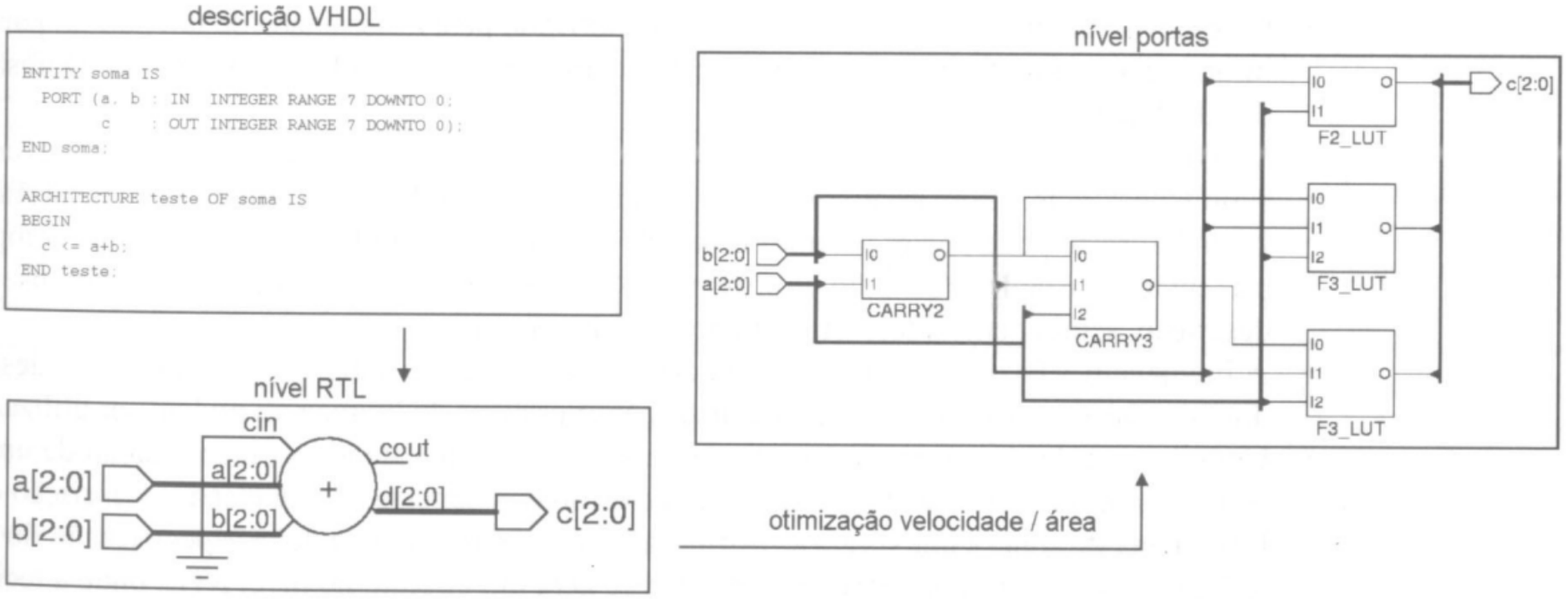
INTRODUÇÃO

Ciclos de projetos:

- **Simulação do modelo:** Resultados mais apurados de comportamento e timing;
- **Geração:** Configuração das lógicas programáveis ou de fabricação de ASICs.

INTRODUÇÃO

Etapas de projetos usando VHDL:

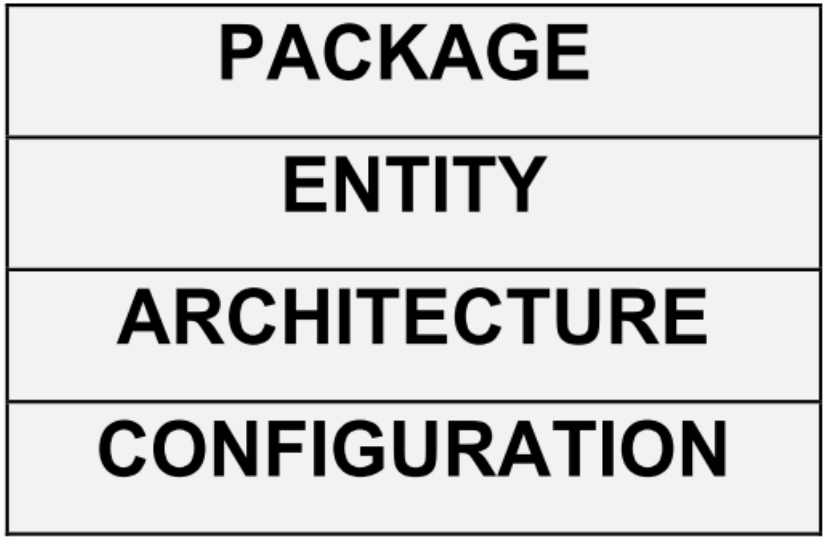


A descrição da operação de um circuito síncrono digital recebe o nome de RTL (do inglês Register Transfer Level)

INTRODUÇÃO
Tipos de circuitos:

- **Circuitos Combinacionais:** circuitos que dependem apenas da combinação das variáveis de entrada;
- **Circuitos Sequenciais:** circuitos que dependem da variável tempo (sincronização, realimentação, etc);
- **Circuitos Hardwired:** circuito projetado para realizar uma tarefa específica, sem a necessidade de programação;
- **Processadores:** sistemas de uso geral, compostos por unidades operativas e de controle, com conjunto de instruções específico;
- **Sistemas embarcados:** sistemas para uso específico com restrições de potência e aplicações de tempo real.

Componentes de um projeto em VHDL



- **Package (Pacote):** constantes, bibliotecas;
- **Entity (Entidade):** pinos de entrada e saída;
- **Architecture (Arquitetura):** implementações do projeto;
- **Configuration (Configuração):** define as arquiteturas que serão utilizadas.

Componentes de um projeto em VHDL

LIBRARY IEEE; **USE IEEE.STD_LOGIC_1164.all;** **USE IEEE.STD_LOGIC_UNSIGNED.all**;	PACKAGE (BIBLIOTECAS)
ENTITY exemplo **IS** **PORT** (<descrição dos pinos de I/O>); **END** exemplo;	ENTITY (PINOS DE I/O)
ARCHITECTURE teste **OF** exemplo **IS** **BEGIN** ... **END** teste;	ARCHITECTURE (ARQUITETURA)

Componentes de um projeto em VHDL

Entity (Entidade)

- Abstração que descreve um sistema, uma placa, um chip, uma função ou uma porta lógica;
- Etapa “caixa preta”, onde é necessário apenas descrever quem são as entradas e saídas do circuito (interface com meio externo).

Componentes de um projeto em VHDL

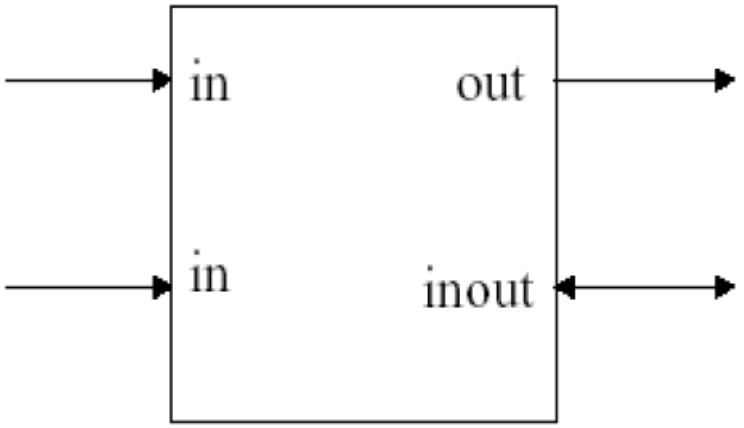
Entity (Entidade)

```
Entity <nome_da_entidade> is
port  (
        entrada_1: in <tipo>;
        entrada_2: in <tipo>;
        saída_1    : out <tipo>;
        ...
        );
end <nome_da_entidade>;
```

Componentes de um projeto em VHDL

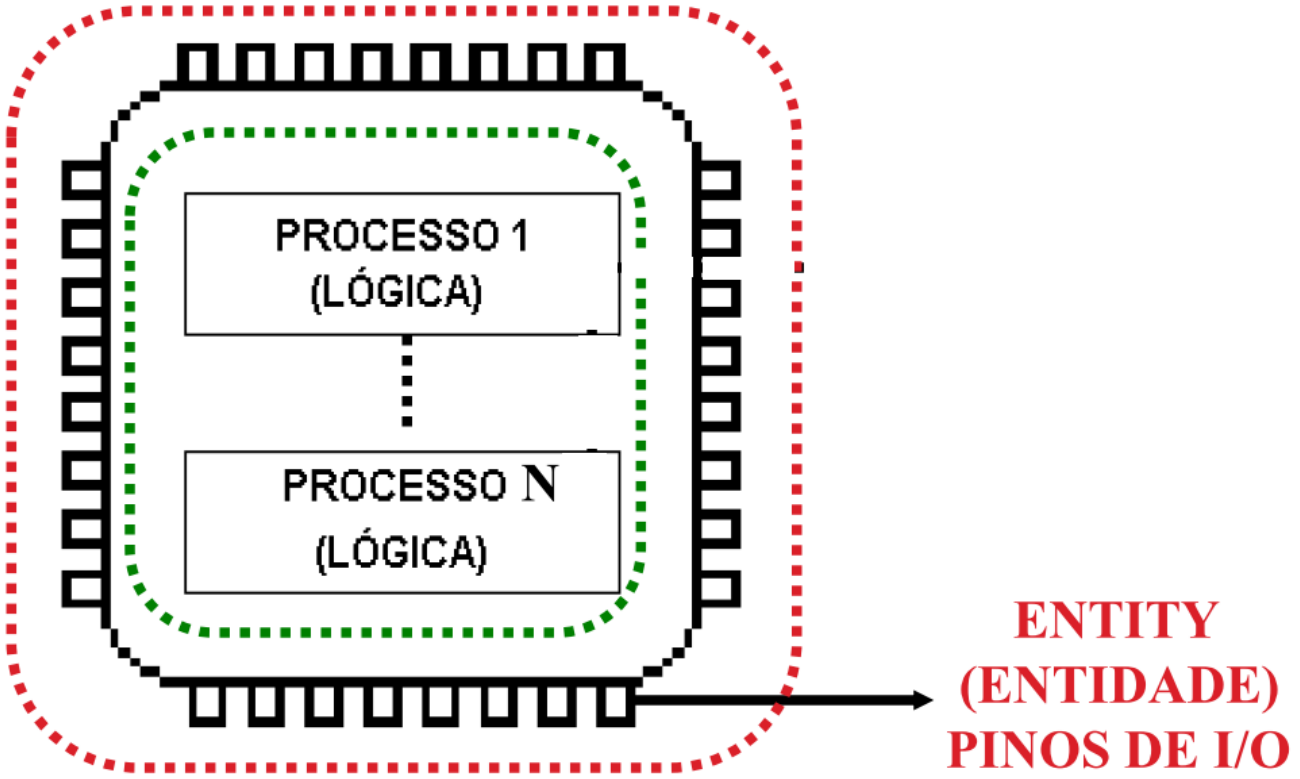
Entity (Entidade)

- Parâmetros:
  - GENERIC: passagem de informações estáticas
  - PORT: correspondem ao pinos de entrada e saída.
- Modos de operação:
  - IN: porta de entrada;
  - OUT: porta de saída (não podem ser usados como entradas, nem seus valores utilizados na lógica interna);
  - INOUT: porta de entrada e saída;
  - BUFFER: saída com possibilidade de realimentação.



Componentes de um projeto em VHDL

Entity (Entidade)



Componentes de um projeto em VHDL

Entity (Entidade)

- Tipo de dados mais utilizados:

bit	Assume valores ‘0’ ou ‘1’. **x: in bit;**
bit_vector	Vetor de bits. **x: in bit_vector(7 downto 0);** **x: in bit_vector(0 to 7);**
std_logic*	**x: in std_logic;**
std_logic_vector	**x: in std_logic_vector(7 downto 0);** **x: in std_logic_vector(0 to 7);**
boolean	Assume valores TRUE ou FALSE

Componentes de um projeto em VHDL

Entity (Entidade)

STD_LOGIC:

- Definida pela biblioteca IEEE:
  - use ieee.std_logic_1164.all;
- Pode assumir nove valores:

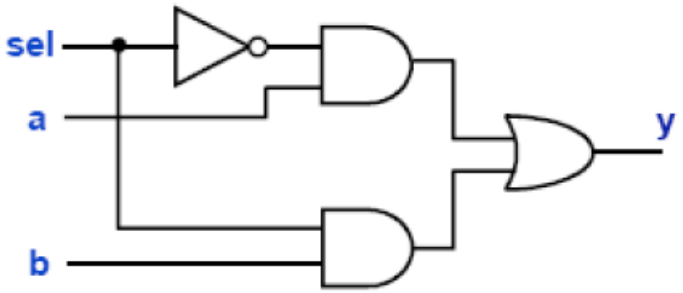
'U': não inicializada	**'Z': alta impedância**
'X': desconhecida	**'W': desconhecida**
'0': valor '0'	**'L': '0' (Low)**
'1': valor '1'	**'H": '1'(High)**
'-': *Don't care.*	

Componentes de um projeto em VHDL

Entity (Entidade)

- Exemplo de circuito:

```
ENTITY exemplo1 IS
   PORT ( sel : IN BIT;
          a : IN BIT;
          b : IN BIT;
          y : OUT BIT);
END exemplo1;
```



- A extensão de um arquivo em VHDL é ”.vhd”. O nome do arquivo DEVE ser o mesmo nome da entidade. No caso acima, o arquivo deve ser salvo com o nome exemplo1.vhd.

Componentes de um projeto em VHDL

Architecture (Arquitetura)

- Especificação do funcionamento do circuito:
- Formada por:
  - Declarações: sinais, constantes, componentes, subprogramas;
  - Comandos: Atribuições a sinais, chamadas a subprogramas, instanciação de componentes, processos.
- Uma entidade pode ter várias arquiteturas: VHDL provê meios de especificar qual arquitetura se deseja utilizar.

Componentes de um projeto em VHDL

Architecture (Arquitetura)

- Descrição:

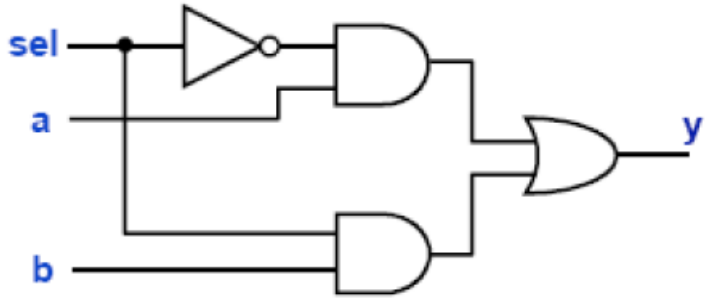
```
ARCHITECTURE nome_identificador OF entidade_abc IS
  --
  -- regiao de declaracoes:
  --   declaracoes de sinais e constantes
  --   declaracoes de componentes referenciados
  --   declaracao e corpo de sub-programas
  --   definicao de novos tipos de dados locais
  --
BEGIN
  --
  -- comandos concorrentes
  --
END;
```

Componentes de um projeto em VHDL

Architecture (Arquitetura)

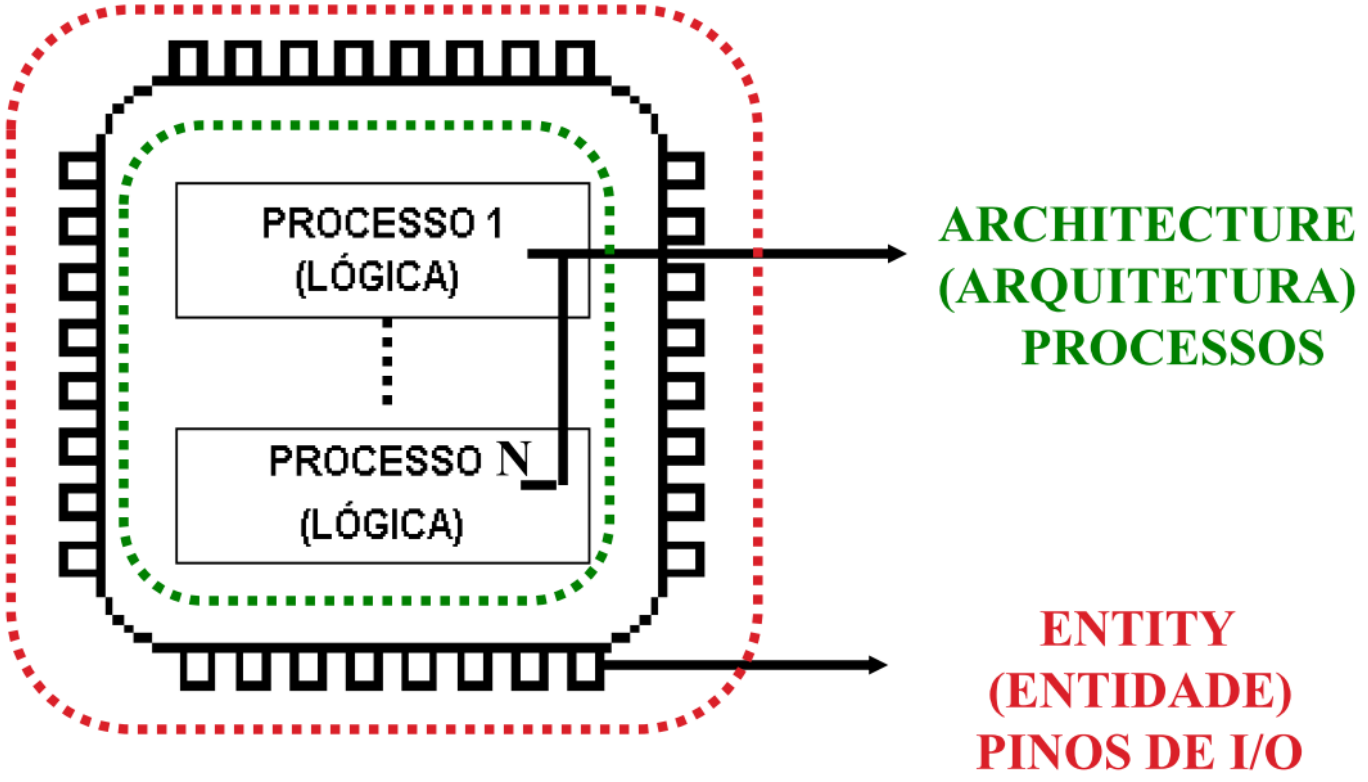
- Exemplo de arquitetura do circuito:

```
ARCHITECTURE comportamento OF exemplo1 IS
BEGIN
    y <= (a AND (NOT(sel))) OR (b AND sel);
END comportamento;
```



Componentes de um projeto em VHDL

Architecture (Arquitetura)



Componentes de um projeto em VHDL

Package (Pacotes)

- Os pacotes (bibliotecas) contém uma coleção de elementos incluindo descrição do tipos de dados;
- Analogia com C/C++: #include <library.h>;
- Para incluir uma biblioteca no código VHDL (início do código):

LIBRARY <nome_da_biblioteca> e/ou

USE <nome_da_biblioteca>.all

Componentes de um projeto em VHDL

Package (Pacotes)

- É necessário o uso de packages quando se deseja utilizar algo não definido pela biblioteca VHDL padrão. A área de packages deve vir antes da área de entidade.

```
library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
```

Biblioteca do usuário (default): work.

Componentes de um projeto em VHDL

Package (Pacotes)

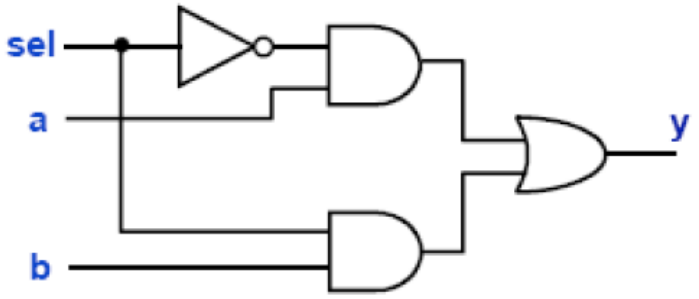
- Exemplo de Packages:

```
package <biblioteca> is
    function soma(a,b: bit) return bit;
    subtype dado is bit_vector(32 downto 0);
    constant mascara : bit_vector(3 donwto 0) := “1100”;
    alias terceiro_bit: bit is dado(3) );
end <biblioteca>.
```

Componentes de um projeto em VHDL

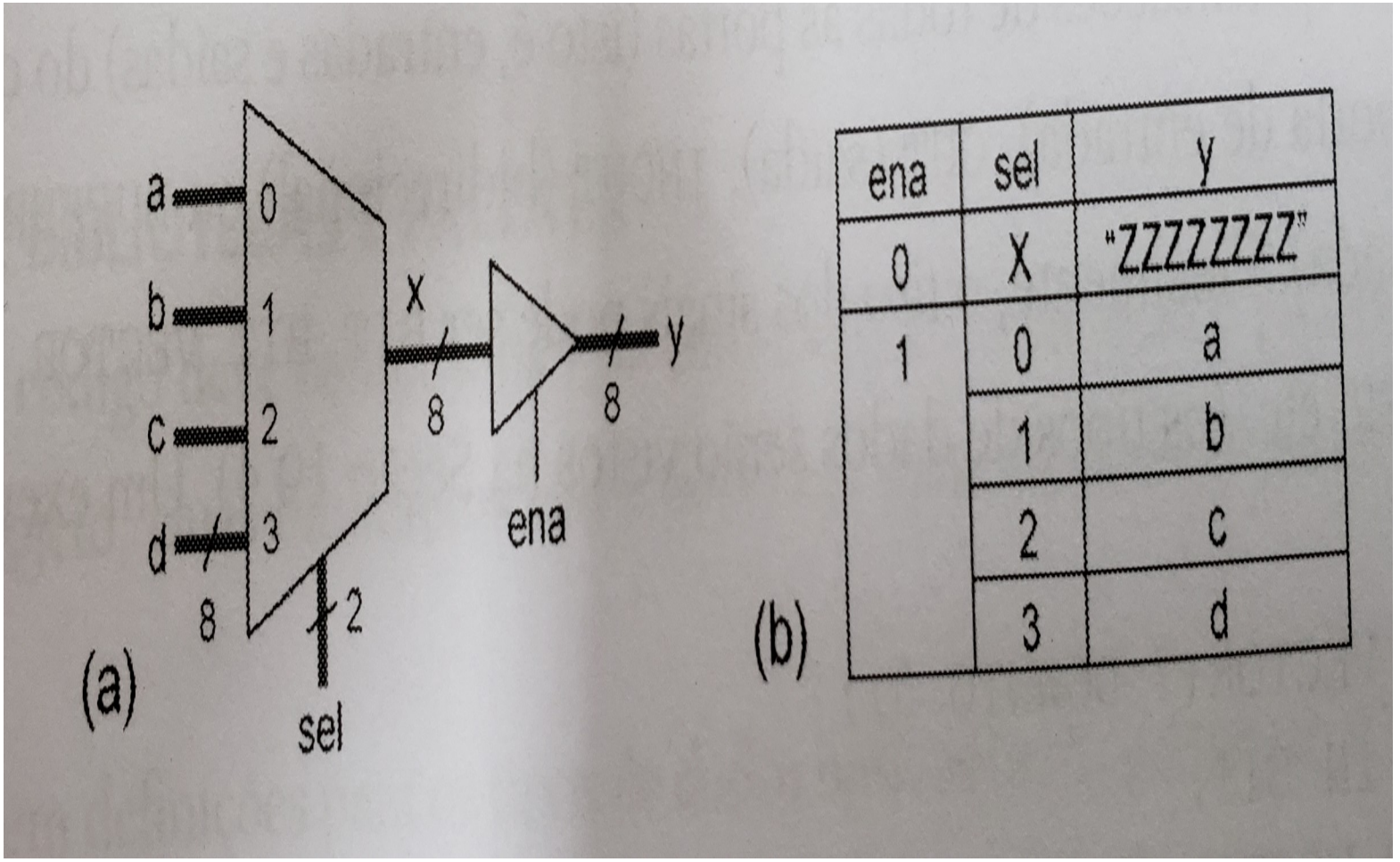
Circuito exemplo completo em VHDL

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY exemplo1 IS
   PORT ( sel, a, b : IN BIT;
          y : OUT BIT);
END exemplo1;

ARCHITECTURE comportamento OF exemplo1 IS
BEGIN
     y <= (a AND (NOT(sel))) OR (b AND sel);
END comportamento;
```



Componentes de um projeto em VHDL

Atividades



ena	sel	y
0	X	"ZZZZZZZZ"
1	0	a
	1	b
	2	c
	3	d

Componentes de um projeto em VHDL

Atividades

```
1  ----------------------------------------------------------
2     LIBRARY ieee;
3     USE ieee.std_logic_1164.all;
4  ----------------------------------------------------------
5     ENTITY Mux IS
6        PORT (a, b, c, d: IN STD_LOGIC_VECTOR (7 DOWNTO 0);
7              sel: IN NATURAL RANGE 0 TO 3;
8              ena: IN STD_LOGIC;
9              y  : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
10    END Mux;
11 ----------------------------------------------------------
12    ARCHITECTURE myarch OF Mux IS
13      SIGNAL X: STD_LOGIC_VECTOR (7 DOWNTO 0);
14    BEGIN
15      x <= a WHEN sel=0 ELSE     -- Mux
16           b WHEN sel=1 ELSE
17           c WHEN sel=2 ELSE
18           d;
19      y <= x WHEN ena='1' ELSE -- Tristate buffer
20          (OTHERS => 'Z');
21    END myarch;
22 ----------------------------------------------------------
```

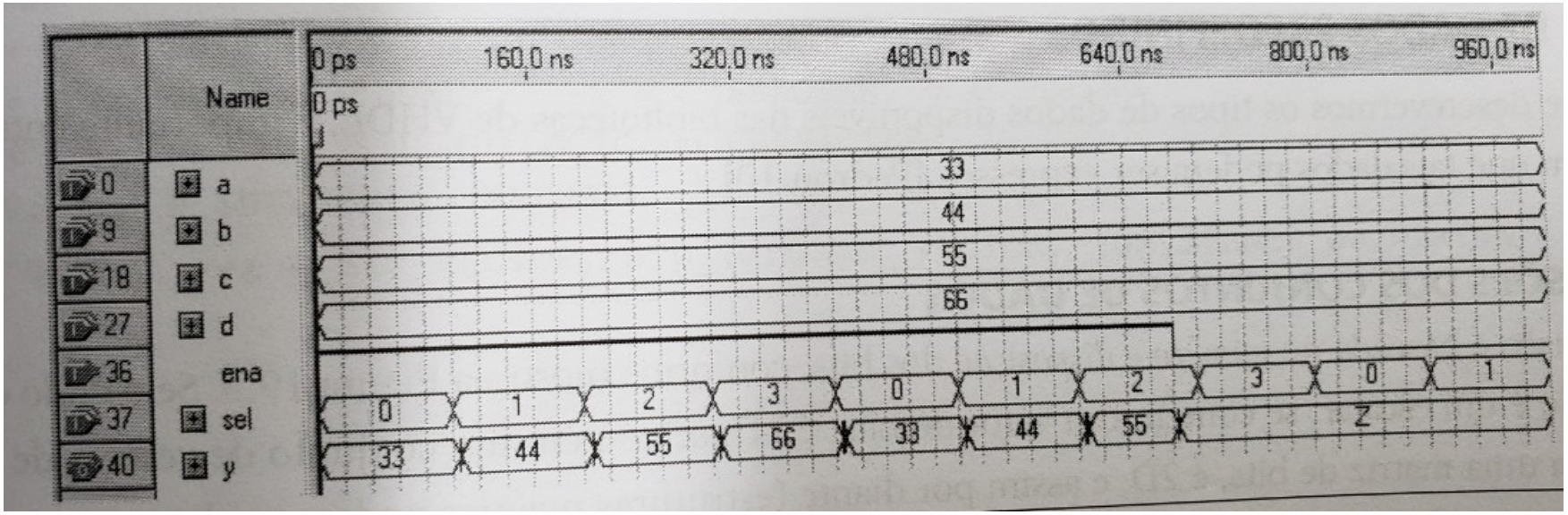
Componentes de um projeto em VHDL

Atividades

```
1  ----------------------------------------------------------
2     LIBRARY ieee;
3     USE ieee.std_logic_1164.all;
4  ----------------------------------------------------------
5     ENTITY test IS
6        PORT (a, b, c, d: IN STD_LOGIC_VECTOR (7 DOWNTO 0);
7              sel: IN NATURAL RANGE 0 TO 3;
8              ena: IN STD_LOGIC;
9              y  : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
10    END test;
11 ----------------------------------------------------------
12    ARCHITECTURE myarch OF test IS
13      SIGNAL X: STD_LOGIC_VECTOR (7 DOWNTO 0);
14    BEGIN
15      x <= a WHEN sel=0 ELSE     -- Mux
16           b WHEN sel=1 ELSE
17           c WHEN sel=2 ELSE
18           d;
19      y <= x WHEN ena='1' ELSE -- Tristate buffer
20          (OTHERS => 'Z');
21    END myarch;
22 ----------------------------------------------------------
```

Componentes de um projeto em VHDL

Atividades



Referencias

- ALTERA. DE2 Development and education board user manual. 2008. Version 1.42.
- ALTERA. Quartus II Introduction Using VHDL Design. 2008.
- MENEZES, M.P.; SATO, L.M.; MIDORIKAWA, E.T. Projeto de Circuitos com Quartus II 9.1.
- Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais,
- Escola Politécnica da USP. Edição de 2011.
- TOCCI, R. J.; WIDMER, N.S.; MOSS, G.L. Sistemas Digitais: Princípios e Aplicações.
- Prentice-Hall, 11 ed., 2011.
- WAKERLY, John F. Digital Design Principles & Practices. 4th edition, Prentice Hall, 2006.