

## 0.1 VHDL

A linguagem VHDL foi desenvolvida pela necessidade de um padrão para o intercâmbio de informações entre fornecedores de equipamentos para o Departamento de Defesa dos Estados Unidos. Esta linguagem é usada para descrever o comportamento de circuitos ou sistemas eletrônicos a partir de um sistema físico. É importante lembrar que esta é uma linguagem concorrente, ou seja, os comandos envolvidos em um mesmo evento acontecem simultaneamente, diferentemente de linguagens de programação de software. Além disso, é uma linguagem portátil, ou seja, independe da tecnologia ou do fornecedor. A Figura 1 mostra as etapas de um projeto utilizando VHDL.

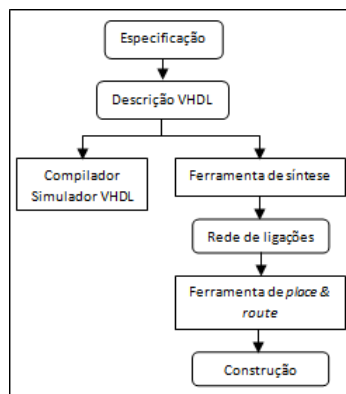


Figura 1: Etapas de Projeto Usando VHDL

Na lógica programável, há dois dispositivos principais: CPLD (*Complex Programmable Logic Devices*) e FPGA (*Field Programmable Gate Arrays*) no campo de ASIC (*Application Specific Integrated Circuits*). A partir do código VHDL, pode-se fabricar um chip de alta complexidade ou executá-lo em um dispositivo programável.

O código VHDL é composto de três partes principais: Biblioteca, Entidade e Arquitetura.

1. Biblioteca (*Library*) : É composta de todas as bibliotecas usadas no projeto.
2. Entidade (*Entity*): Determina as entradas e saídas do circuito.
3. Arquitetura (*Architecture*): Contém o código VHDL que descreve a forma de como o circuito deve ser comportar (*function*).

### 0.1.1 Biblioteca

Uma biblioteca têm várias implementações de código que são úteis a outros projetos. A Figura 2 ilustra a estrutura típica de uma biblioteca. O código, normalmente, é escrito na forma de funções (*Functions*), procedimentos (*Procedures*) ou componentes (*Components*), que ficam dentro de pacotes (*Packages*) e depois é compilado na biblioteca.

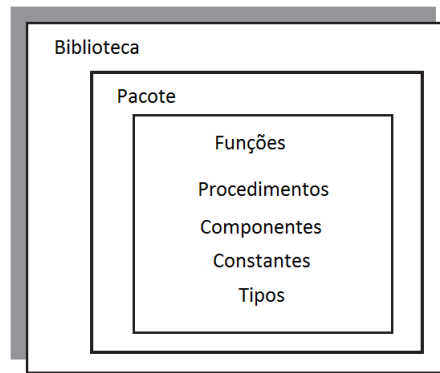


Figura 2: Estrutura de uma Library [?]

### 0.1.2 Entidade

Uma entidade de projeto pode representar uma simples porta lógica como um sistema completo. A declaração da entidade define a interface com o ambiente exterior, como, por exemplo, as entradas e saídas. Os quatro modos de porta são:

1. IN : Apenas entrada.
2. OUT: Apenas saída.
3. BUFFER: Saída que controla sinal interno.
4. INOUT: Porta bidirecional

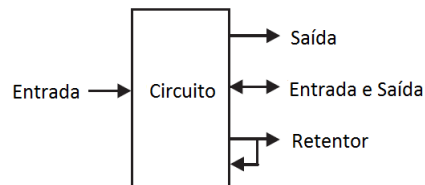


Figura 3: Tipos de Entrada e Saída [?]

### 0.1.3 Arquitetura

A arquitetura contém a parte lógica da entidade utilizando suas entradas e saídas. Ainda é possível declarar sinais internos dentro da arquitetura, estes sinais são chamados classes. São elas:

1. CONSTANT - Define um objeto com valor estático.
2. VARIABLE - São objetos que podem ter o seu valor alterado, e são usadas em regiões de código seqüencial.
3. SIGNAL - São objetos que podem ter o seu valor alterado, e são usadas em regiões de código concorrente ou seqüencial. É bom lembrar que a porta de uma entidade realiza a declaração de um sinal.

```

ARCHITECTURE architecture_name OF entity_name IS
    [declarations]
BEGIN
    (code)
END architecture_name;

```

Figura 4: Sintaxe de uma Architecture

A arquitetura é composta de duas partes, uma para declarações, onde sinais e constantes são declarados e outra onde fica o código. Como no caso da entidade, o nome da arquitetura pode ser qualquer nome (exceto palavras reservadas), até mesmo o nome da entidade.

## 0.2 FPGA

Um dispositivo FPGA é um semicondutor utilizado amplamente no processamento de informações. Criado pela Xilinx Inc., teve seu lançamento no ano

de 1985 como um dispositivo que poderia ser programado de acordo com as aplicações do usuário (programador).

O FPGA é composto basicamente por três tipos de componentes: blocos de entrada e saída (IOB), blocos lógicos configuráveis (CLB) e chaves de interconexão (Switch Matrix). Os blocos de entrada e saída (I/O) formam uma borda ao redor do dispositivo. Cada um desses blocos pode servir como entrada, saída ou acesso bi-direcional a outros pinos de I/O. Os blocos lógicos são dispostos de forma bidimensional, as chaves de interconexão são dispostas em formas de trilhas verticais e horizontais entre as linhas e as colunas dos blocos lógicos como é mostrado na figura.

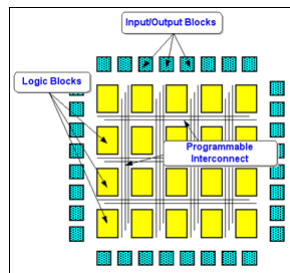


Figura 5: Estrutura de uma FPGA [?]

1. CLB (Configuration Logical Blocks): São circuitos construídos pela reunião de Flip-Flops (entre 2 e 4) juntamente com lógica combinacional. A partir de CLBs é possível construir elementos funcionais lógicos.
2. IOB (Input/Output Block): Circuitos que realizam o interfaceamento das saídas provenientes das saídas das combinações de CLBs. São basicamente buffers, que funcionarão como um pino bidirecional entrada e saída do FPGA.
3. Switch Matrix (chaves de interconexões): Trilhas que conectam os CLBs e IOBs. O terceiro grupo é composto pelas interconexões. Os recursos de interconexões possuem trilhas para conectar as entradas e saídas dos CLBs e IOBs para as redes apropriadas. Geralmente, a configuração é estabelecida por programação interna das células de memória estática, que determinam funções lógicas e conexões internas implementadas no FPGA entre os CLBs e os IOBs. O processo de escolha das interconexões é chamado de roteamento.

### 0.2.1 Programação de FPGAs

De modo geral, a etapa inicial do projeto é seguida de uma simulação funcional. Neste ponto, é utilizado um simulador para conferir as saídas com as diversas entradas de teste. Uma vez de posse de uma representação funcional do hardware, a compilação é iniciada. Esta compilação é dividida em 2 etapas.

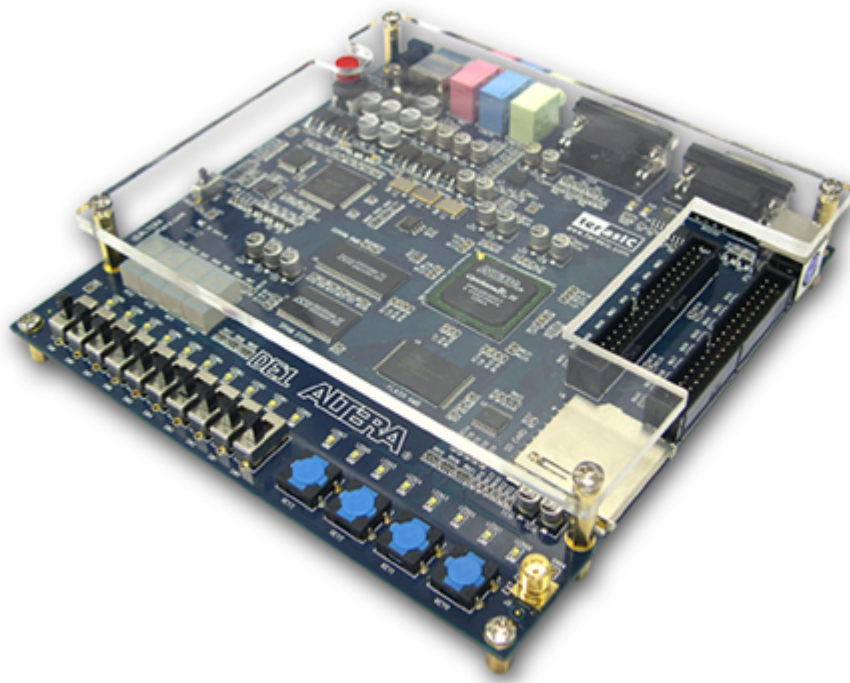
A primeira, chamada síntese (synthesis), é onde o compilador tem uma idéia de como implementar o projeto e falta apenas posicionar e rotear as estruturas lógicas em macrocélulas, interconexões e pinos de entrada e saída, que é a segunda etapa.

Feita a compilação, um bitstream é criado, ou seja, já sabemos quais são os dados binários que deverão ser carregados no FPGA ou CLPD para fazer com que o chip execute um projeto em particular. Geralmente, as empresas que desenvolvem dispositivos programáveis, fornecem também softwares capazes de carregar o bitstream no hardware utilizando linguagens de descrição de hardware (Hardware Description Language ? HDL).

## 0.3 Arquitetura FPGA da família Cyclone II

Neste projeto será utilizado o kit de desenvolvimento DE1, Figura 0.3. Tal kit possui uma FPGA da família Cyclone® II que trabalha com o Quartus II cujas as características são apresentadas logo a seguir.

beginfigure[h]



Kit

de desenvolvimento DE1

A família de FPGAs Cyclone II® da Altera é manufaturada em wafers de silício de 300mm utilizando processadores com low-k dielétricos de 90nm da TSMC para garantir a rápida disponibilidade e o baixo custo. Minimizando a área de silício e oferecendo suporte a sistemas digitais complexos em um único chip, estes dispositivos podem facilmente competir no mercado com os ASICs.

Tendo uma densidade lógica de até 68.416 elementos lógicos (LEs) e um máximo de portas utilizáveis de 622 como é mostrado na Tabela 1, são soluções ideais para uma ampla gama de automóveis, processamento de vídeo, comunicações, teste e medições entre outras áreas.

beginfigure[h]

<i>Table 1–1. Cyclone II FPGA Family Features</i>							
Feature	EP2C5	EP2C8	EP2C15	EP2C20	EP2C35	EP2C50	EP2C70
LEs	4,608	8,256	14,448	18,752	33,216	50,528	68,416
M4K RAM blocks (4 Kbits plus 512 parity bits)	26	36	52	52	105	129	250
Total RAM bits	119,808	165,888	239,616	239,616	483,840	594,432	1,152,000
Embedded multipliers (3)	13	18	26	26	35	86	150
PLLs	2	2	4	4	4	4	4
Maximum user I/O pins	158	182	315	315	475	450	622

Características Dispositivos Família Ciclone II [?]

### 0.3.1 Arquitetura