

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Tipos de HDL e Compiladores de alto nível para FPGA

João Paulo de Oliveira

11611BCC046

Graduando em ciência da computação

Uberlândia

2017

# 1 Tipos de HDL

## 1.1 VHDL

O VHDL foi desenvolvido pelo Departamento de defesa dos Estados Unidos na década 1980 para documentar o comportamento das ASICs do equipamento vendido às Forças Armadas americanas. É baseada em Pascal e ADA. Com o sucesso do uso do VHDL, a sua definição se tornou de domínio público, assim ela foi padronizada pela IEEE (Institute of Electrical and Electronic Engineers). Vejamos um pouco de sua sintaxe:

- **Comentários:** Os comentários são iniciados por `--` e terminam no final da linha.
- **Entity:** Onde é descrito a interface para descrever as entradas e saídas. Exemplo:

```
entity ANDGATE is
.....      --código da entidade
end ANDGATE;
```

- **Architecture:** É o corpo do sistema, onde são feitas as atribuições, operações, etc. Podem existir várias architecture dentro da mesma entity. Exemplo:

```
architecture RTL of ANDGATE is
begin
...
end RTL;
```

- **Principais operadores:**
  - `+` soma ou identidade
  - `-` subtração ou negação
  - **mod** módulo
  - **rem** operador mod
  - `<=` atribuição
  - `/=` diferente
- **After:** After tem a finalidade de ativar o estado indicado depois de determinado tempo. Exemplo:

```
A <= '1' after 1s, '5' after 10s;
```

- **Wait:** Faz o processador gastar ciclos sem executar nenhuma ação na forma de segundos. Exemplo:

```
X <= '8';  
wait for 30s;
```

Exemplo de código de uma porta lógica AND:

```
-- importa std_logic da IEEE library  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
-- Declara a entidade da porta  
entity PORTAAND is  
    port (  
        A : in std_logic;  
        B : in std_logic;  
        S: out std_logic);  
end PORTAAND;  
architecture E of PORTAAND is  
begin  
  
    S <= A and B;  
  
end E;
```

## 1.2 Verilog

O Verilog é uma HDL padronizada pela IEEE que suporta a projeção, verificação e implementação de projetos analógicos, digitais e híbridos em vários níveis de abstração. Criada por Prabhu Goel e Phil Moorby, durante o inverno de 1983/1984, para a Automated Integrated Design Systems (mais tarde Gateway Design Automation, em 1985). A empresa Gateway Design Automation foi comprada pela Cadence Design Systems em 1990. Vejamos um pouco de sua sintaxe, que é bem parecida com a linguagem C:

- **Comentários:** // para comentar até o final da linha.
- **Estado lógico:**

- 1 - Um lógico
- 0 - Zero lógico
- Z - Alta impedância
- X - Valor indefinido (pode ser qualquer um)

- **Principais operadores:**

- + Soma binária
- - Subtração binária
- ~ Negação bit à bit
- | Negação bit à bit
- = Atribuição
- == Comparação

- **Exemplo de Código:**

```
module EXEMPLO_AND(x,y,out); //declaracao do modulo AND
input wire a, b; // entrada de dados
output wire s; // saida de dados
assign s = a & b; //AND com a chamada assign
endmodule // final do modulo AND
```

### 1.3 SystemVerilog



O SystemVerilog foi iniciado com a doação da Superlog language para a empresa Accellera em 2002. Ele é uma combinação de uma HDL e uma linguagem de verificação de hardware, é baseada em Vera, VHDL e Verilog. A sua sintaxe é uma extensão do Verilog, porém, ela oferece alguns itens adicionais, por exemplo:

- **Novos tipos de dados:**

- **String:** São vetores de caracteres. Exemplo:

```
string s1 = "Hello";
string s2 = "world";
string p = ".?!";
string s3 = {s1, " ", s2, p[2]}; //Concatena string
$display("%s",s3); //Resulta: "Hello, world!"
```

- **Tipos de dados enumerados:** Permite que palavras (não reservadas) possam ser assimiladas à valores crescentes. Exemplo:

```
typedef enum logic [2:0] {
    Vermelho, Verde, Azul, Magenta, Amarelo
} color_t;
```

- **Vetor dinâmico:** Um vetor que pode ser de tamanho variável. Exemplo:

```
int Vetor[]; //Vetor dinâmico que pode ser alocado posteriormente
```

- **Orientação à objetos:** As classes são suportadas no SystemVerilog que oferecem funções de manipulação semelhante ao C++ como encapsulamento, Polimorfismo, Herança, a palavra reservada "new" se refere ao construtor automático e a linguagem também possui um Coletor de Lixo para objetos não referenciados. Exemplo:

```
virtual class Memory;
    virtual function bit [31:0] read(bit [31:0] addr); endfunction
    virtual function void write(bit [31:0] addr, bit [31:0] data); endfunction
endclass
```

```
class SRAM #(parameter AWIDTH=10) extends Memory;
    bit [31:0] mem [1<<AWIDTH];
```

```
    virtual function bit [31:0] read(bit [31:0] addr);
        return mem[addr];
    endfunction
```

```
    virtual function void write(bit [31:0] addr, bit [31:0] data);
        mem[addr] = data;
    endfunction
```

```
endclass
```

## 2 Compiladores de Alto nível para FPGA

### 2.1 Cλash

O "clash" como é pronunciado, é uma linguagem de descrição de hardware funcional baseada na linguagem Haskell. O compilador Cλash transforma as descrições de alto nível para VHDL, Verilog ou SystemVerilog que são descrições de baixo nível. Exemplo de código:

```
module MAC where
import CLaSH.Prelude
ma acc (x,y) = acc + x * y
```

O código acima foi feito para um arquivo com nome MAC, que deve corresponder ao nome do módulo descrito na primeira linha. Na segunda linha é importada a biblioteca necessária para o funcionamento do programa. Na terceira linha é feita a função "ma" que recebe as variáveis x,y e acc e faz a operação  $acc + (x * y)$ . Exemplo de uso:

```
>>> ma 2 (6,8)
50
```

### 2.2 Chisel

Chisel é uma linguagem construtora de hardware desenvolvida em UC Berkeley que suporta hardware avançado usando geradores altamente parametrizados. Incorporada na linguagem Scala de programação, a Chisel utiliza hierarquia, orientação à objetos, possui também uma construção funcional combinada com tipos abstratos de dados resultando em uma tradução para a linguagem de baixo nível Verilog. Exemplo de código:

```
import chisel3._
import scala.collection.mutable.ArrayBuffer

/** multiplicador 4 por 4 usando tabela verdade.*/
class Mul extends Module {
  val io = IO(new Bundle {
    val x    = Input(UInt(4.W))
    val y    = Input(UInt(4.W))
    val z    = Output(UInt(8.W))
  })
  val muls = new ArrayBuffer[UInt]()
}
```

```

for (i <- 0 until 16)
  for (j <- 0 until 16)
    muls += (i * j).U(8.W)
val tbl = Vec(muls)
io.z := tbl((io.x << 4.U) | io.y)
}

```

## 2.3 MyHDL

O MyHDL é a tradução automática da linguagem Python em uma linguagem de descrição e verificação de hardware como VHDL ou Verilog, que é Open Source e foi desenvolvida por Jan Decaluwe. O MyHDL possui orientação à objeto desenvolvimento orientado a teste (TDD), testes funcionais, bibliotecas prontas com diversas funcionalidades implementadas (manipulação de texto, funções matemáticas, parte gráfica e etc.) e tudo mais que faz parte do universo Python está disponível para desenvolvimento de hardware com o MyHDL.

## 2.4 SystemC

O SystemC, desenvolvido pela Accellera, é um conjunto de classes e macros do C++ que possuem simulação de manipulação de eventos, bem como os tipos de dados oferecidos pelo C++. Em certos aspectos, a linguagem imita deliberadamente a semântica do VHDL e do Verilog. O SystemC também traz uma simulação de kernel [?]. Veja um exemplo de código de um somador:

```

#include "systemc.h"

SC_MODULE(adder){          // declaração do módulo (class)
  sc_in<int> a, b;          // portas
  sc_out<int> sum;

  void do_add(){            // processo
    sum.write(a.read() + b.read()); //or just sum = a + b
  }

  SC_CTOR(adder){           // construtor
    SC_METHOD(do_add);
    sensitive << a << b;
  }
};

```

## Referências

Disponível em: <<https://pt.stackoverflow.com/questions/216032/vhdl-%C3%A9-linguagem-de-programa%C3%A7%C3%A3o>> acesso em 01 set. de 2017

Disponível em: <[https://edisciplinas.usp.br/pluginfile.php/3020729/mod\\_resource/content/0](https://edisciplinas.usp.br/pluginfile.php/3020729/mod_resource/content/0)> acesso em 01 set. de 2017

Disponível em: <<https://www.embarcados.com.br/tutorial-de-verilog-operadores/>> acesso em 01 set. de 2017

Disponível em: <<http://www.verilog.com/>> acesso em 01 set. de 2017

Disponível em: <<https://www.embarcados.com.br/myhdlhardwarepython/>> acesso em 01 set. de 2017

Disponível em: <<http://www.myhdl.org/>> acesso em 01 set. de 2017

Disponível em: <<https://www.fpgarelated.com/showarticle/25.php>> acesso em 01 set. de 2017

Disponível em: <<http://www.accellera.org/downloads/standards/systemc>> acesso em 01 set. de 2017

Disponível em: <<https://chisel.eecs.berkeley.edu/>> acesso em 01 set. de 2017

Disponível em: <<http://www.clash-lang.org/>> acesso em 01 set. de 2017

Disponível em: <<https://hackage.haskell.org/package/clash-prelude-0.11.2/docs/CLaSH-Tutorial.html>> acesso em 01 set. de 2017