

VHDL Cheat Sheet (simplificado)

Cabeçalho de Arquivos

```
LIBRARY <biblioteca>;  
USE <biblioteca>.<pacote>.<item>;  
Importam bibliotecas e disponibilizam recursos.
```

Entidades

```
entity <entidade> is  
port (  
    <nome_do_sinal1>: modo tipo; ...  
    <nome_do_sinaln>: modo tipo);  
end entity;
```

Define as entrada e saídas de uma entidade.

modo: indica a direção do sinal:

- in** – para sinais de entrada (input);
- out** – para sinais somente de saída (output);

tipo: tipo do dado

- bit** e **bit_vector** – valores 0 ou 1
- std_logic** e **d** – valores digitais*
- vetores são definidos por: (**<N> downto 0**)

- integer** – valores numéricos inteiros
- inteiros tem um faixa: **range <min> to <max>**

*prefira o uso de **std_logic** na criação das entidades.

Arquiteturas

```
architecture <nome> of <entidade>  
<declarações>;  
begin  
    <operações>;  
end nome_arquitetura;
```

Declarações de Arquiteturas

Definições de outras entidades, sinais, etc.

```
component <entidade>  
    port original da entidade;  
end component;  
Definições interface de entidade a ser usada.
```

```
signal <nome_do_sinal1>: tipo;  
Definições de um sinal interno para a arquitetura.
```

```
constant <nome> : <tipo> := <valor>  
Definições de uma constante para a arquitetura.
```

```
variable <nome>: <tipo> [:=<valor>]  
Definições de uma variável (não é um sinal interno).
```

Operações de Arquiteturas

Define o comportamento lógico da entidade.

Operadores

Operadores lógicos: **and**, **or**, **xor**, **nand**, **nor**, **xnor**, **not**
Operadores relacionais: **=**, **/=**, **<**, **<=**, **>**, **>=**
Operadores de shift: **sll**, **srl**, **sla**, **sra**, **rol**, **ror**
Operadores aritméticos: **+**, **-**, *****, ******, **/**, **mod**, **abs**, **rem**
Operador de concatenação: **&**

Atribuições Concorrentes

- **Direta:**

```
<sinal> <= <expressão>;  
Redirecionamento de sinal (como uma conexão).
```

- **Condicional:**

```
<sinal> <= <expr> when <cond> else  
    <expr> when <cond> else  
    <expr>;
```

- **Seletor:**

```
with <sinal> select  
    sinal <= <expr> when <escolha>,  
    <expr> when <escolha>,  
    <expr> when others;
```

Port map:

```
<nome>:<entidade> port map (ports);  
Realiza o mapeamento das portas das entidades.
```

Processos Combinacionais

```
process(<sinais>)  
<declarações>;  
begin  
    <operações>;  
end process;
```

- **If-then-else:**

```
if <condição> then <atribuição>  
else <atribuição>  
end if;
```

- **Case:**

```
case <sinal>  
    when <cond> => <atrib>;  
    when <cond> => <atrib>;  
end case;
```

Processos Sequenciais.

```
process(clk) ...  
Sinal de clock é informado na lista de sinais.
```