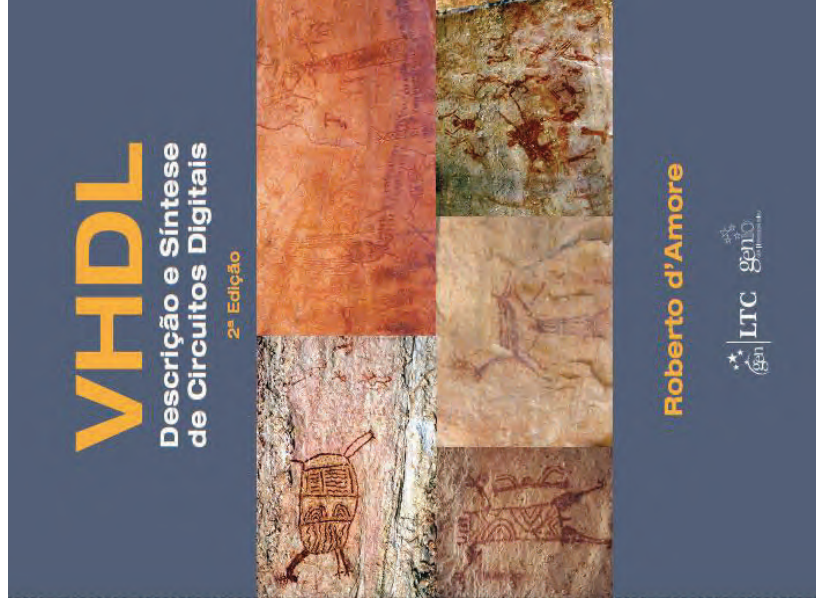


VHDL - Descrição e Síntese de Circuitos Digitais

Roberto d'Amore

Editora LTC www.ltceditora.com.br



Para informações adicionais, consulte: www.ele.ita.br/~damore/vhdl



Aspectos gerais da linguagem

- **Suporta diversos níveis de hierarquia**
 - uma descrição pode ser: conjunto de descrições interligadas
- **Estilo de uma descrição**
 - um circuito pode ser descrito de diversas maneiras
 - níveis de abstração (exemplo: comportamental - portas lógicas - rede de ligações)
- **Uma descrição**
 - pode mesclar diferentes níveis de abstração (em um mesmo código)
- **Ferramentas de síntese:**
 - suportam diferentes estilos de descrição
 - normalmente: modos preferenciais devem ser empregados



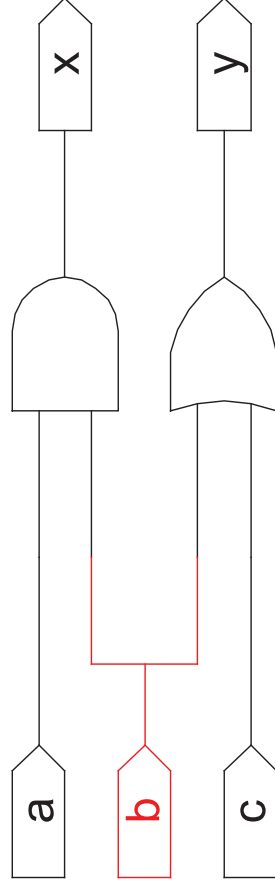
Aspectos gerais da linguagem

- **Linguagem concorrente**
 - ordem dos comandos: não importa
 - mudança de valor em um sinal:
 - acarreta a execução de todos os comandos envolvidos



- **Concorrência: exemplo**

- alteração do valor em **b**:
 - execução conjunta dos comandos nas linhas 8 e 9
- simulador
 - internamente executadas sequencialmente
 - é necessário um mecanismo interno para as avaliações (Capítulo 3)
 - a ordem da avaliação dos comandos é irrelevante
 - o resultado é sempre o mesmo

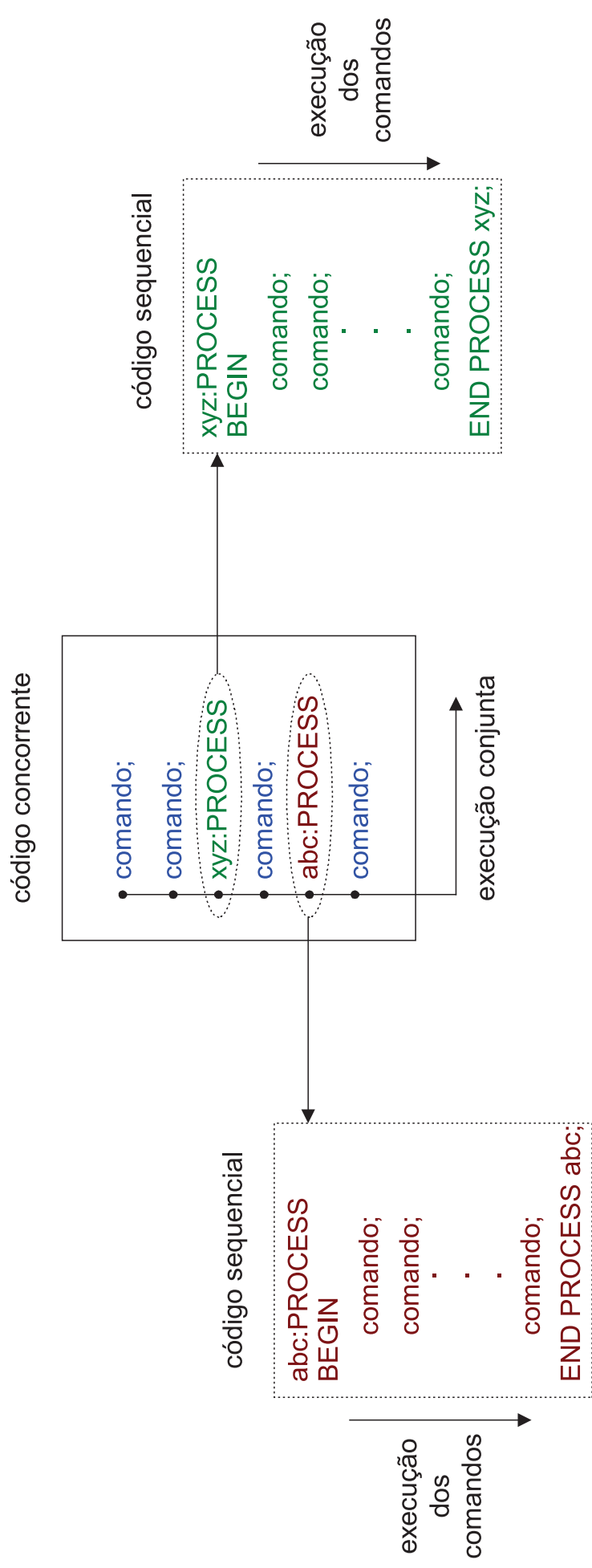


descrição VHDL

```
1 ENTITY portas IS
2   PORT (a, b, c : IN BIT;
3         x, y   : OUT BIT);
4 END portas;
5
6 ARCHITECTURE teste OF portas IS
7 BEGIN
8   x <= a AND b;
9   y <= c OR b;
10  END teste;
```

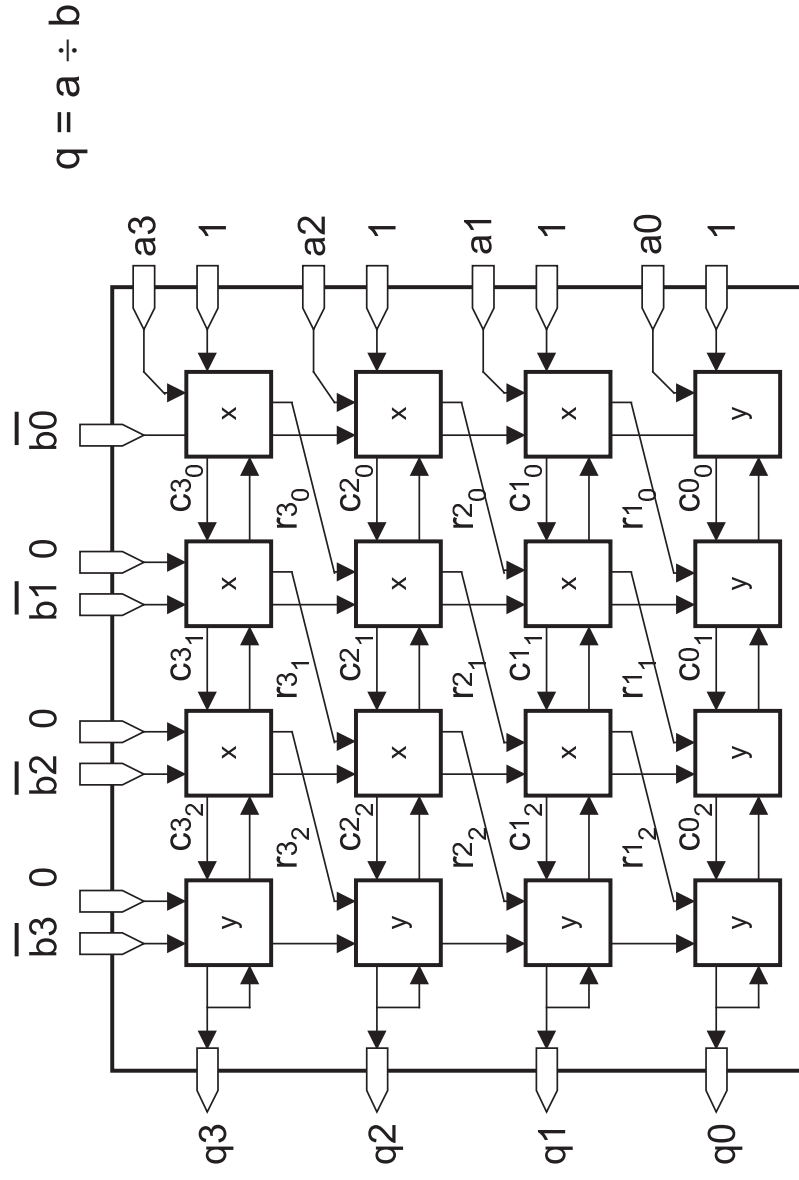
- **Comandos sequenciais:**

- somente em regiões delimitadas no código
- cada região é executada concorrentemente
- delimitado em regiões específicas:
 - subprogramas
 - processos
- comandos próprios nestas regiões



- **Subprogramas:** procedimento (procedure) e função (function)

- empregados em:
 - rotinas de conversão
 - descrição de uma parte do circuito
 - divisor paralelo (composto de várias células interligadas)
 - cada célula distinta → descrita por um subprograma



• Definição de biblioteca e pacote (library) (package)

- pacotes
 - definição de recursos de uso comum:
subprogramas, constantes, novos tipos etc.
 - evitam a repetição de definições comuns
 - exceto o pacote padrão (*standard package*), qualquer pacote deve ser declarado para se tornar visível

- bibliotecas

- local de armazenamento das informações compiladas
- biblioteca padrão de armazenamento: **work**
- diferentes locais podem ser definidos:
 - úteis para trabalhos em equipes (local comum de armazenamento)
 - organização do projeto



• Tipos

- Exemplo de tipos pré-definidos no pacote padrão (STD):

`bit`, `boolean`, `integer`

- tipo `bit`

exemplo de valores: `0` e `1`

- tipo `boolean`:

exemplo de valores: `true` e `false`

- tipo `std_logic` → um novo tipo definido no pacote `std_logic_1164`

exemplo de valores:

`0` `1` `U` (não inicializado) `X` (desconhecido) `Z` (alta impedância)



- **Definição de novos tipos**

- o usuário pode definir novos tipos

- exemplo: estados de uma máquina:

```
TYPE estado IS (parado, falha, correto, caso_1);
```

cada estado deve ser representado por um código

- exemplo de codificação:

```
parado = 0001, falha = 0010, correto = 0100, caso_1 = 1000
```

- forma de codificação dos estados:
executado pela ferramenta de síntese

- **Procedimentos de teste**

- uma descrição VHDL pode gerar estímulos de teste

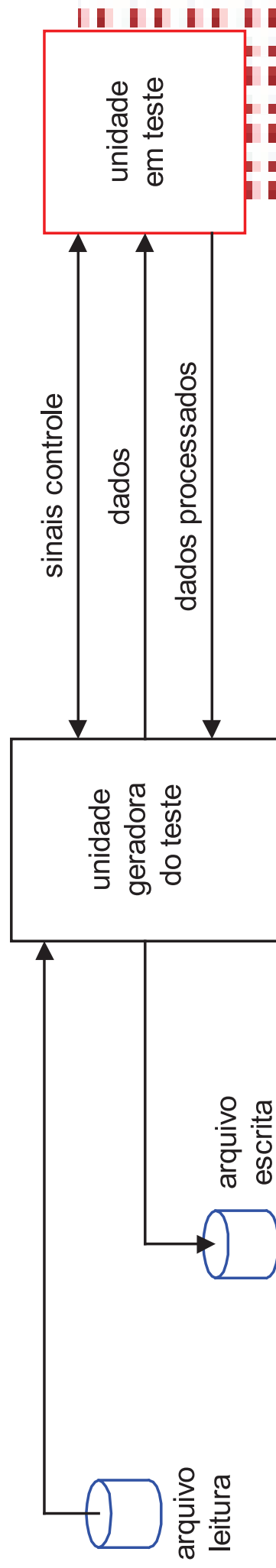
exemplo:



- verificação funcional da descrição
 - após a síntese: temporização disponível
 - verificação de restrições como:
 - hold time*
 - setup time*

- **Operações com arquivos**

- não suportadas por ferramentas de síntese
- são por demais abstratas
- aplicação: operações de teste de outras descrições



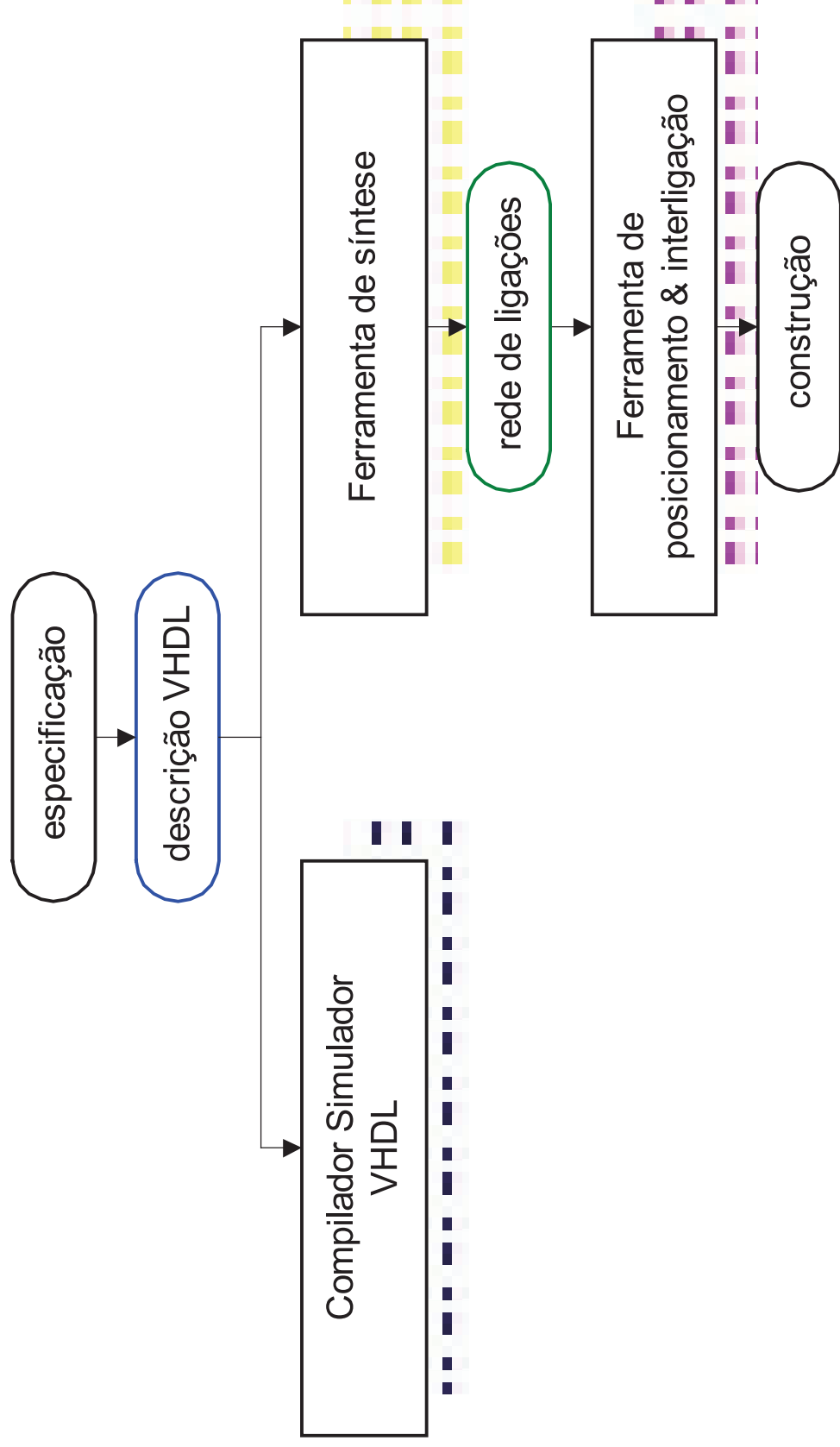
Síntese de circuitos

- **VHDL:** não foi concebida para síntese de circuitos
 - consequência:
 - *nem todas as construções da linguagem são suportadas na síntese*
- **Motivos da limitação:**
 - 1- falta de correspondência entre: construção \Leftrightarrow circuito real
 - exemplo: *flip-flop* com dois terminais de relógio
 - código pode ser simulado \Leftrightarrow não existe o *flip-flop*
 - 2- impossibilidade da síntese direta
 - exemplo: multiplicação de dois números reais
 - código pode ser simulado \Leftrightarrow circuito muito complexo



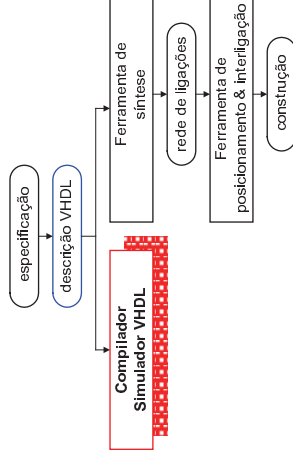
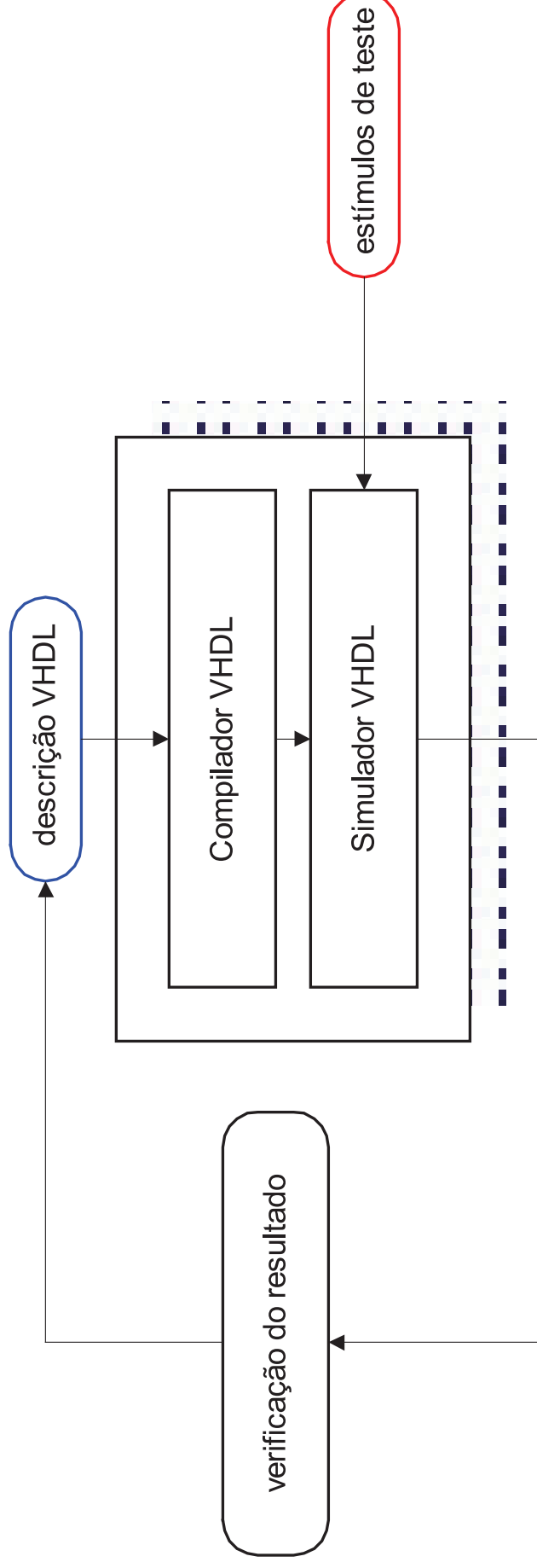
Etapas gerais de uma síntese

- * Elaboração da descrição → ferramenta de compilação
- * Síntese da descrição → ferramenta de síntese
- * Posicionamento e interligação → ferramenta de síntese/ferramenta de programação



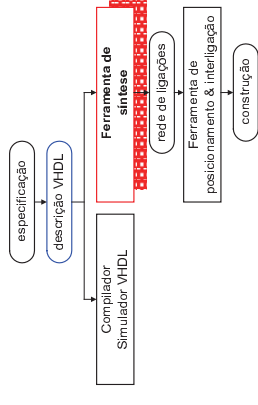
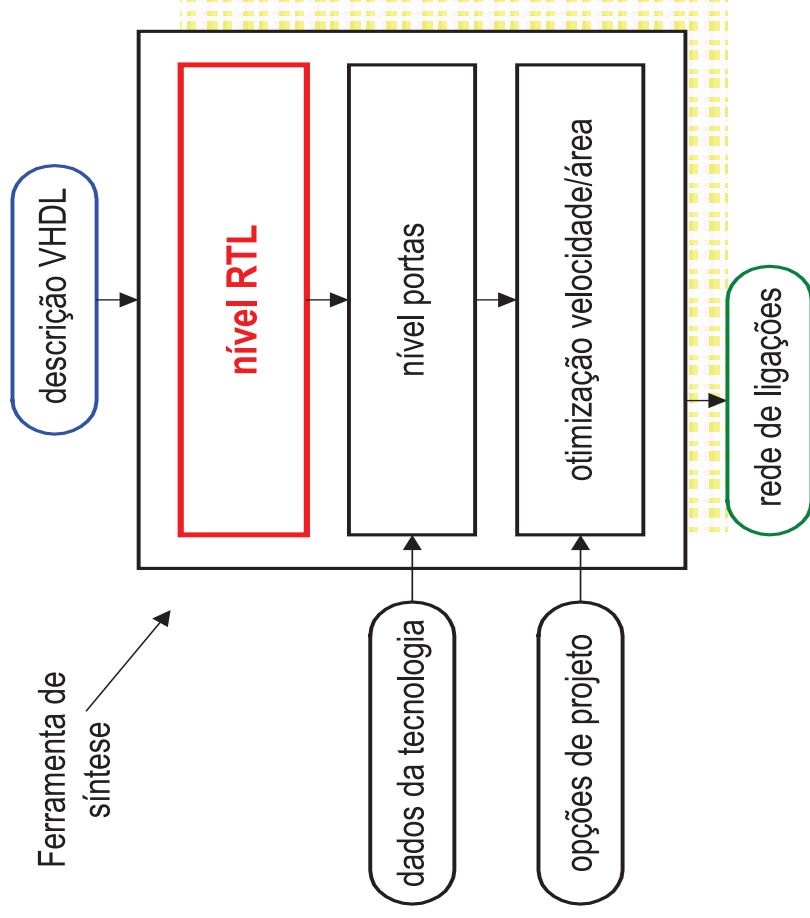
1- Elaboração da descrição

- um mesmo circuito:
 - várias formas de descrição possíveis
 - diferentes níveis de abstração
- proposta inicial da descrição:
 - pode conter estruturas muito abstratas para a síntese direta
- processo iterativo:
 - permite atingir o grau de detalhamento necessário para síntese



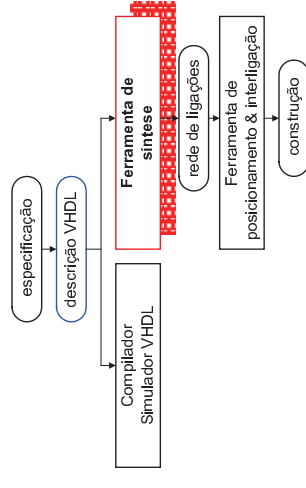
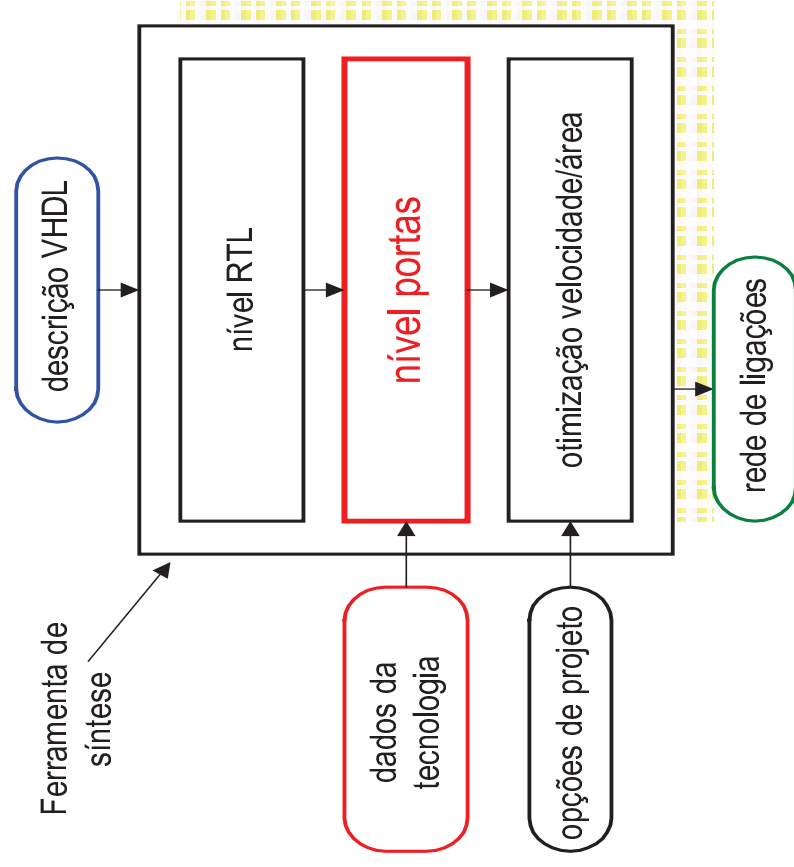
2- Síntese da descrição (verificação de erros – nível RTL)

- verificação de erros de sintaxe
- inferência das estruturas necessárias: gerado circuito nível RTL
 - emprega primitivas da ferramenta de síntese:
(portas lógicas, somadores, comparadores etc.)
 - circuito nível RTL: não é associado a nenhuma tecnologia



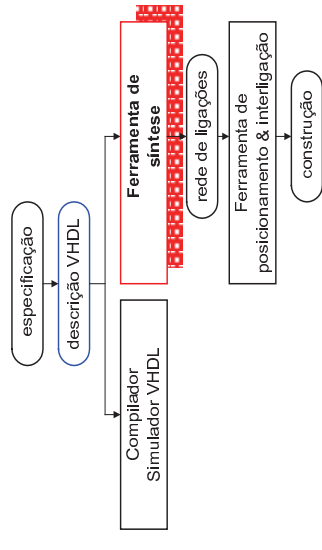
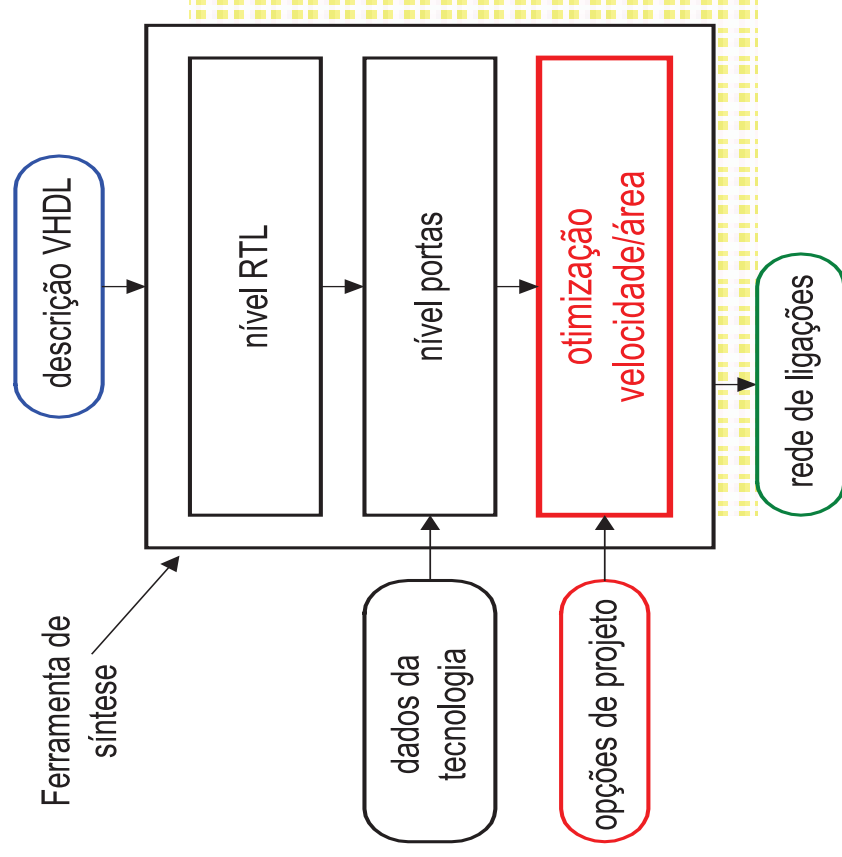
2- Síntese da descrição (nível portas)

- novo circuito considerando a tecnologia utilizada
- emprega: primitivas da tecnologia empregada
módulos otimizados fornecidos pelo fabricante
(exemplo de módulos: somadores, contadores)



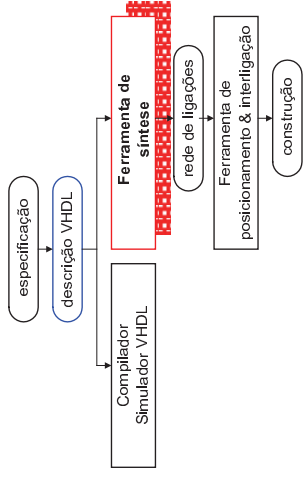
2- Síntese da descrição (otimização)

- definição de opções de projeto
velocidade/área (normalmente conflitantes)



2- Síntese da descrição (ilustração das operações)

- descrição do circuito
- circuito nível RTL sintetizado
- circuito nível portas (final)

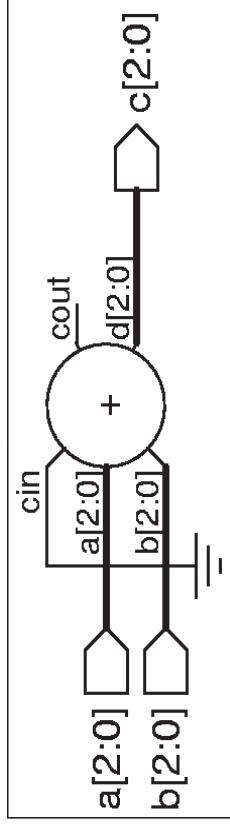


descrição VHDL

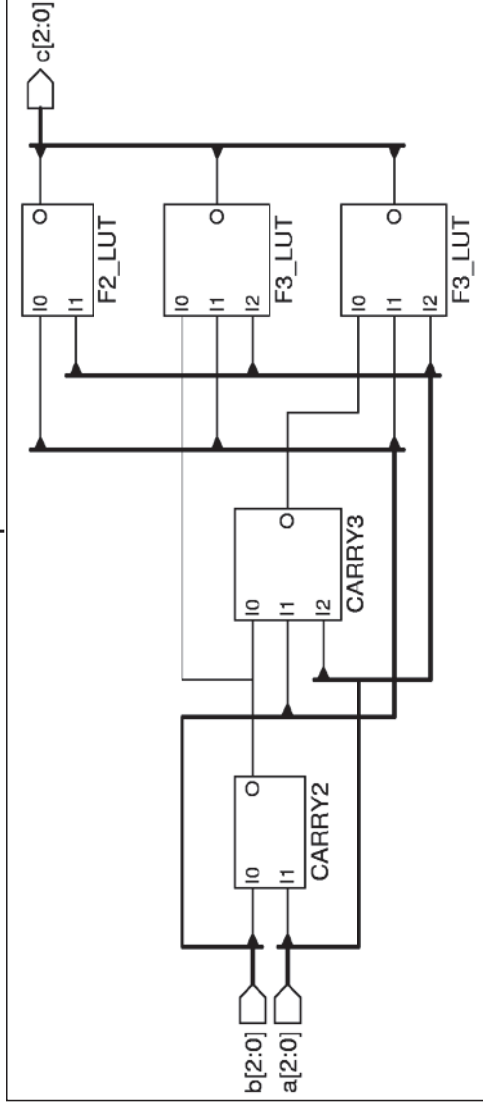
```
ENTITY soma IS
  PORT (a, b : IN INTEGER RANGE 7 DOWNTO 0;
        c : OUT INTEGER RANGE 7 DOWNTO 0);
END soma;

ARCHITECTURE teste OF soma IS
BEGIN
  c <= a+b;
END teste;
```

nível RTL



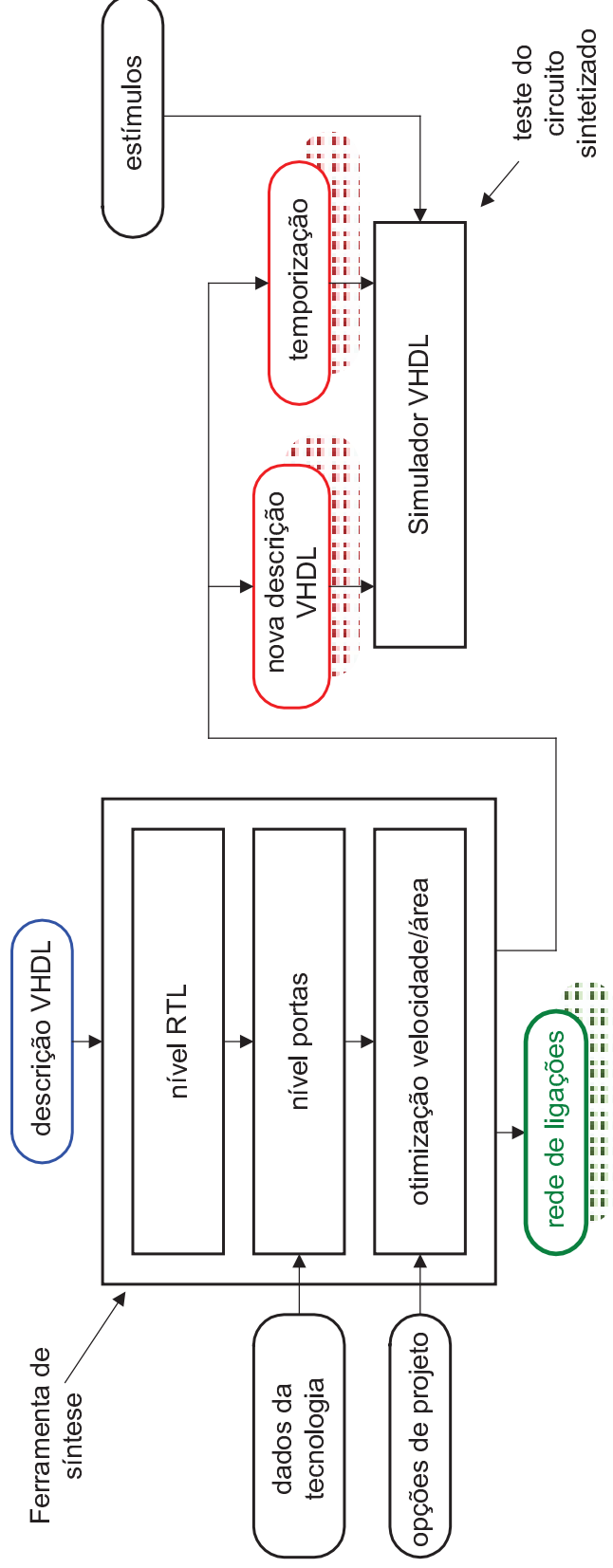
nível portas



otimização velocidade/área

2- Síntese da descrição (arquivos gerados)

- **rede de ligações**
- empregado pela próxima ferramenta
 - exemplo: formato EDIF
- **nova descrição VHDL**
- contém informações preliminares de temporização
- simulações contendo informações de temporização



3- Posicionamento e interligação (*placement routing*)

- ferramenta apropriada:
 - posiciona e interliga as primitivas/componentes
- dispositivo empregado na implementação:
 - *FPGA* - dispositivo lógico programável
 - *ASIC* - circuito integrado de aplicação específica
- arquivo de temporização mais preciso:
 - interligações definidas (e atrasos por elas gerados)
- novas simulações podem ser executadas

