

# Introdução ao Sistema da Synopsys

Simulação / Síntese

*Paulo Flores*

Outubro 1994

Este documento introduz as ferramentas da simulação e síntese da Synopsys através de um exemplo. São apresentados os vários passos necessários para o projecto de um circuito digital a partir da sua descrição em VHDL.

## 0 Configuração da área de trabalho

Para utilizar o software da Synopsys cada utilizador deve de executar um ficheiro de comandos de inicialização com o seguinte comando (ou coloca-lo no ficheiro `.cshrc`):

```
$ source /net/calvin/soft/eurochip/synopsys/synopsys.init
```

Este ficheiro de comandos define a *path* e algumas variáveis necessárias para a utilização das ferramentas da Synopsys.

As várias ferramentas da Synopsys podem ser configuradas através de ficheiros próprios que são lidos no inicio da execução da cada ferramenta. Existem três tipos de configuração possíveis:

1. Configuração a nível do sistema (ficheiros `$$SYNOPSYS/admin/setup/.synopsys*`).
2. Configuração a nível do utilizador (ficheiro `$$USER/.synopsys*`).
3. Configuração a nível do projecto (ficheiro `./ .synopsys*`).

O primeiro ficheiro a ser lido corresponde à configuração a nível sistema. Este ficheiro é definido pelo administrador e não pode ser alterado. Os ficheiros de configuração a nível de utilizador e nível projecto são lidos em seguida. Este ficheiros devem a reflectir, respectivamente, a configuração genérica do utilizador e especifica de cada projecto.

Os ficheiros de configuração necessários para as ferramentas de síntese e simulação são:

- `.synopsys_vss.setup` - Configuração da ferramenta de simulação.

Neste ficheiro definem-se as correspondências entre os nomes lógicos das bibliotecas de VHDL e a sua localização no sistema de ficheiros de Unix. Podem ainda ser definidas neste ficheiro outras variáveis para controlar o simulador de VHDL. Na figura 1 apresenta-se o exemplo de um ficheiro de configuração que define a resolução do simulador em 1nS e que mapeia a biblioteca de defeito de VHDL (`work`) na directoria `work_lib`. Este ficheiro pode ser copiado da directoria `~pff/exemplo` e alterado para reflectir a configuração do utilizador.

```
TIMEBASE = NS
TIME_RES_FACTOR = 1

DEFAULT      : ~pff/exemplo/work_lib
```

Figura 1: Ficheiro de configuração da ferramenta de simulação (`.synopsys_vss.setup`).

- `.synopsys_dc.setup` - Configuração da ferramenta de síntese.

Neste ficheiro é necessário definir o nome do projectista, o caminho para as bibliotecas necessárias à ferramenta e quais os ficheiros que definem a tecnologia e os símbolos a utilizar. Na figura 2 apresenta-se um ficheiro de configuração que pode ser copiado da directoria `~pff/exemplo` e alterado para reflectir a configuração do utilizador.

```
designer = "Paulo Flores";

search_path = { . \
                synopsys_root + "/local/x3000/libraries" \
                synopsys_root + "/xilinx/libraries" \
                synopsys_root + "/libraries/syn"} ;

target_library = {class.db} ;
symbol_library = {class.sdb generic.sdb} ;
synthetic_library = {standard.sldb, \
                    dw01.sldb, dw02.sldb, dw03.sldb, \
                    xilinx.sldb};
```

Figura 2: Ficheiro de configuração da ferramenta de síntese (`.synopsys_dc.setup`).

# 1 Análise e simulação da descrição VHDL

## 1. Descrição do circuito

A descrição do circuito deve ser feita de forma a que cada par entidade-arquitetura seja descrito num ficheiro com o mesmo nome da entidade. Na figura 3 apresenta-se o circuito exemplo que vai ser usado neste documento, um decodificador de 2 para 4. Esta descrição encontra-se no ficheiro `descodificador.vhd`, que pode ser copiado da directoria `~pff/exemplo`.

```
entity descodificador is
    port(entrada: in bit_vector(1 downto 0);
         saida: out bit_vector (3 downto 0));
end descodificador;

architecture exemplo of descodificador is
begin

    process (entrada)
    begin
        case entrada is
            when "00" =>
                saida <= "0001";
            when "01" =>
                saida <= "0010";
            when "10" =>
                saida <= "0100";
            when "11" =>
                saida <= "1000";
        end case;
    end process;

end exemplo;
```

Figura 3: Código VHDL de um decodificador de 2 para 4.

## 2. Análise da descrição VHDL

A análise de um ficheiro contendo uma descrição VHDL pode ser feita utilizando a ferramenta `vhdlan` ou o seu equivalente gráfico `gvan`. Assim, a análise para a biblioteca VHDL de defeito (`work`) do decodificador apresentado, pode ser obtida como o seguinte comando:

```
$ vhdlan descodificador.vhd
```

Algumas opções podem ser dadas na linha de comandos das quais se destacam:

- l Se existirem erros, produz um ficheiro auxiliar, de extensão `.lis`, com a listagem do código VHDL e respectivos erros.
- spc Realiza simultaneamente análise da descrição para a ferramenta de simulação e de síntese. São geradas mensagens de aviso para as construções que não

puderem ser sintetizadas.

### 3. Descrição dos estímulos de simulação (código de teste)

A forma mais “simples” e normalizada de descrever os estímulos de simulação consiste na utilização da própria linguagem VHDL. Assim, deve ser criada um entidade de teste cuja arquitectura instancia o componente a testar e gera o estímulos necessários à sua verificação.

Na figura 4 apresenta-se o código de teste utilizado para verificar a descrição do descodificador. Na arquitectura desta entidade de teste apresenta-se adicionalmente a forma de criar um relógio de período 20 nS.

```
entity descodificador_tb is
end descodificador_tb;

architecture for_test of descodificador_tb is
  component test
    port(entrada: in bit_vector(1 downto 0);
         saida: out bit_vector (3 downto 0));
  end component;

  signal ent: bit_vector(1 downto 0);
  signal sai: bit_vector(3 downto 0);
  signal clock: bit := '0';

  for DUT: test use entity work.descodificador(exemplo);
begin
  DUT: test
    port map(ent, sai);

  clock <= not clock after 10 ns;

  dados: process
  begin
    wait for 10 ns;
    ent <= "00", "01" after 20 ns, "10" after 40 ns, "11" after 80 ns;
    wait for 100 ns;
    assert false
      report "Fim da simulacao";
  end process dados;

end for_test;
```

Figura 4: Descrição VHDL para testar o descodificador.

O código de teste deve ser analisado de forma idêntica à que foi descrita na alínea anterior.

### 4. Invocação do simulador

A simulação de uma entidade VHDL pode ser realizada através da ferramenta `vhdlbxb`. Esta ferramenta apresenta também facilidades que auxiliam a detecção de erros na descrição VHDL do circuito. Para a simulação do código de teste do exemplo apresentado deve-se dar o seguinte comando<sup>1</sup>:

```
$ vhdlbxb descodificador_tb
```

A ferramenta `vhdlbxb` cria uma janela que se encontra dividida em duas grandes zonas (janela `Vhdlbxb` da figura 5). A zona superior destina-se a visualizar o código VHDL que vai ser simulado e a zona inferior é à interacção com o simulador. Nesta zona são apresentados alguns resultados do simulador e podem ser dados comandos pelo utilizador através do teclado.

Se o ficheiro de simulação for lido correctamente aparecerá na parte superior da janela do simulador a descrição correspondente ao código de teste.

#### 5. Seleção dos sinais a visualizar

A indicação à ferramenta de quais os sinais a visualizar durante a simulação é feita através do comando `trace`. Este pode ser dado através da linha de comandos (comando `trace` seguido dos sinais a visualizar), ou através da interface gráfica (selecção gráfica do sinal na zona superior da janela seguido da selecção do botão `Trace`). Os sinais seleccionados vão sendo acrescentado numa janela de visualização de formas de ondas .

#### 6. Simulação

Para inicializar a simulação deve-se seleccionar o botão `Run`. Opcionalmente pode ser indicado durante quanto tempo se pretende simular o circuito.

Os resultados da simulação podem ser verificado na janela de visualização de formas de ondas. Na figura 5 apresenta-se as janelas de criada pela ferramenta `vhdlbxb` e a janela de visualização de formas de onda, após ter sido simulado o código de teste.

#### 7. Sair do simulador

Para sair do simulador e fechar a janela `vhdlbxb` selecciona-se o primeiro menú (`Execute`) e escolhe-se a última opção (`Quit`).

## 2 Síntese do circuito

#### 1. Invocação da ferramenta de síntese

A síntese de uma descrição VHDL é feita pela ferramenta `design_analyzer`. Assim, deve ser dado o seguinte comando:

```
$ design_analyzer &
```

---

<sup>1</sup>O “parâmetro” fornecido à ferramenta de simulação é o nome da entidade que se pretende simular e não o nome do ficheiro onde esta se encontra descrita.

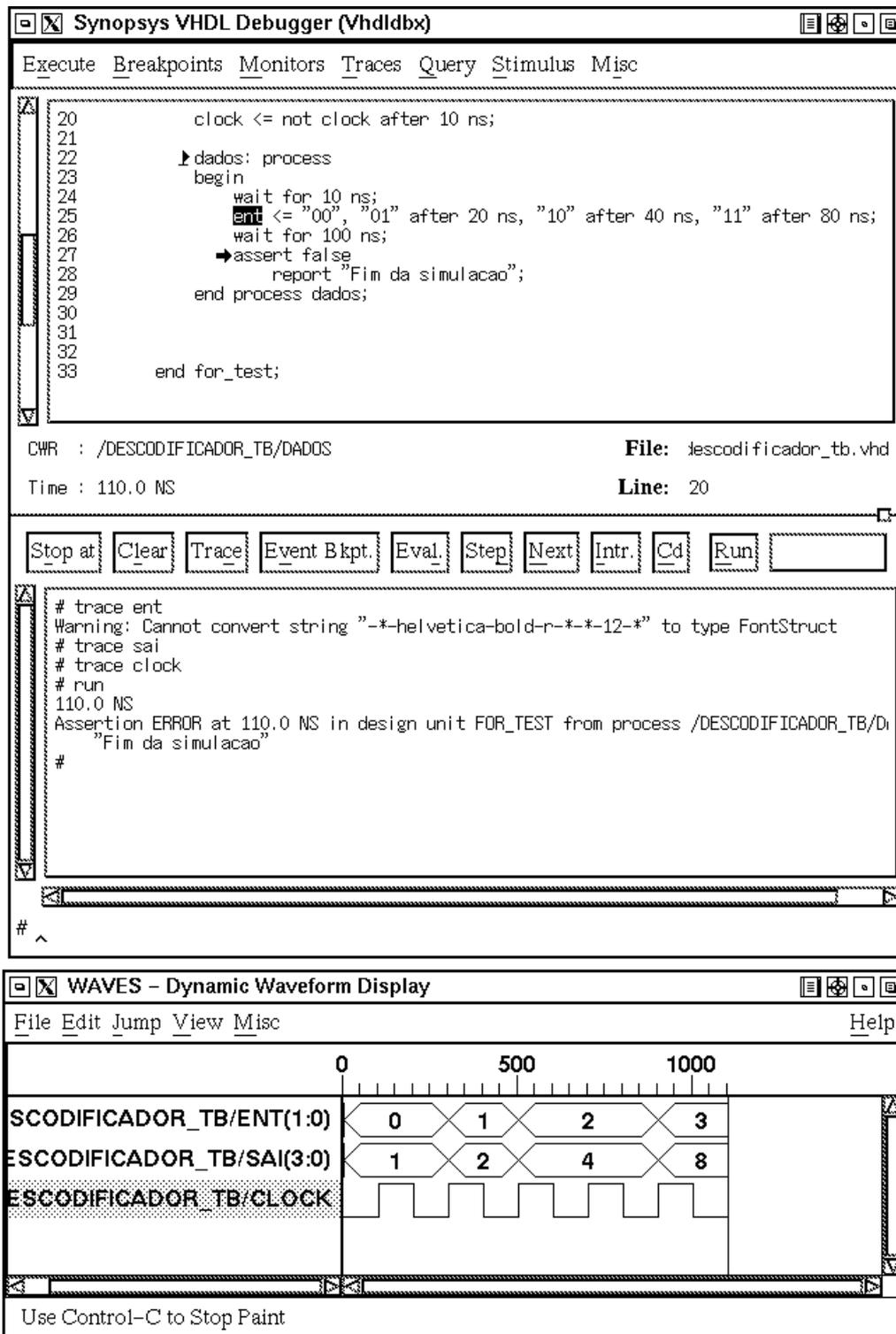


Figura 5: Janelas do simulador e do visualizador de formas de ondas

A figura 6 apresenta a janela criada por esta ferramenta através da qual, utilizando uma interface gráfica, são dados os comandos necessários para realizar a síntese do circuito.

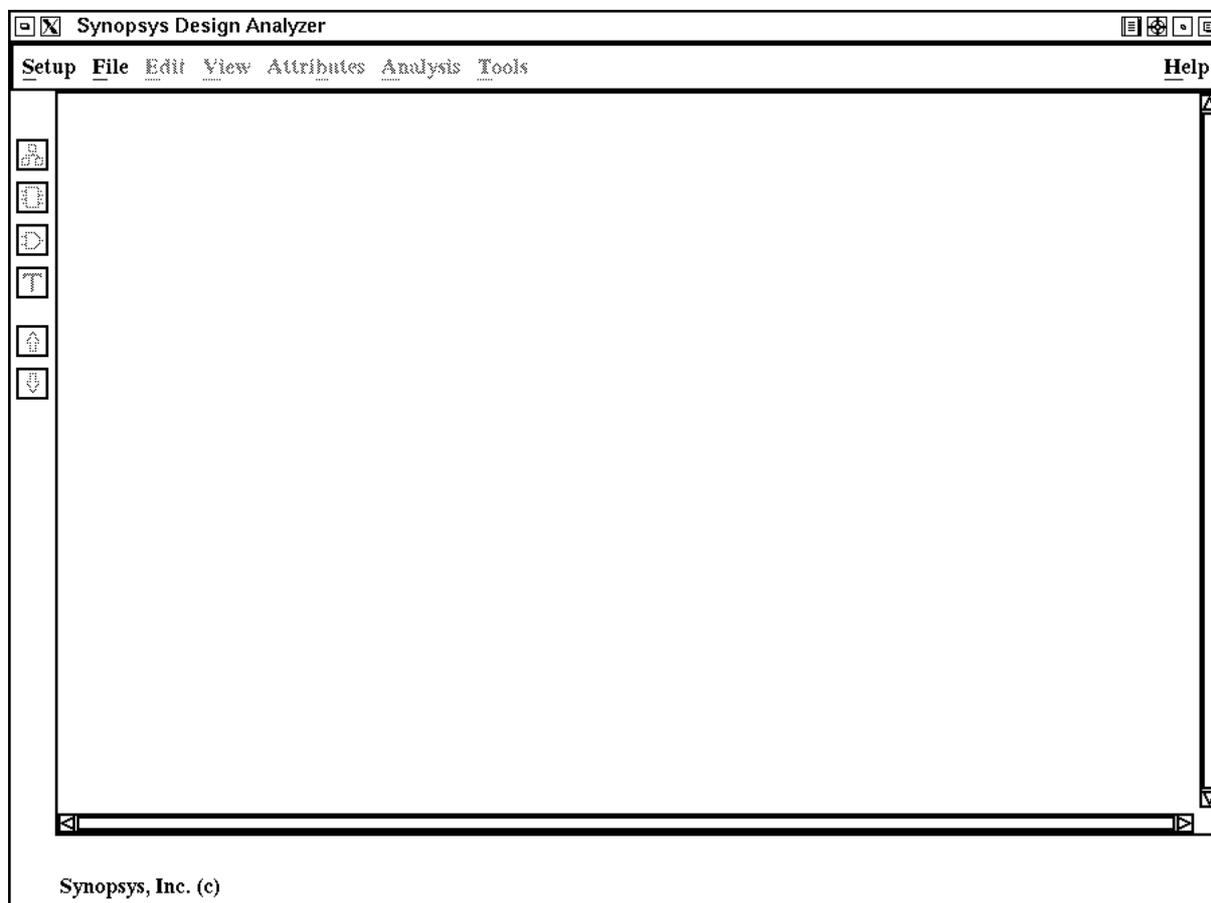


Figura 6: Janela criada pela ferramenta de síntese(design\_analyzer).

## 2. Seleção da tecnologia

Antes de se realizar a síntese da descrição VHDL é necessário indicar à ferramenta qual a tecnologia em que se pretende obter o circuito sintetizado.

Para indicar a biblioteca de tecnologia à ferramenta selecciona-se do menú **Setup** a opção **Defaults...** Uma nova janela é criada em que o campo correspondente à **Target Library** deve indicar o nome do ficheiro da tecnologia desejada e o campo **Symbol Library** deve indicar o ficheiro de símbolos correspondente à tecnologia escolhida.

NOTA: O ficheiro de configuração `.synopsys_dc.setup` pode ser alterado para que o valor por defeito do parâmetro **Target Library** e **Symbol Library** seja o pretendido.

## 3. Leitura da descrição VHDL

A leitura de uma descrição em VHDL pode ser realizada de duas formas:

(a) Através dos comandos **Analyze** e **Elaborate**

O comando **Analyze** realiza a análise da descrição do circuito para uma dada biblioteca VHDL. A selecção do circuito e da biblioteca é feita através uma janela de interacção criada pela ferramenta. Após a escolha do ficheiro com a descrição do circuito (`descodificador.vhd`) e da biblioteca de VHDL (`work`) seleccione o botão **OK**. Uma nova janela é criada para indicar o resultado da análise da descrição. Para fechar esta janela seleccione o botão **Cancel**.

O comando **Elaborate** “carrega” uma descrição do circuito de uma biblioteca de VHDL para a ferramenta de síntese. Na janela de interacção criada selecciona-se a biblioteca de VHDL (`work` ou `default`) e depois uma das descrições (por entidade-arquitectura) que tenha sido analisada para essa biblioteca (`descodificador(exemplo)`). Depois de seleccionar o botão **OK** uma nova janela é criada que indica o resultado da elaboração da descrição.

NOTA: Se para a simulação VHDL a descrição do circuito foi analisada com a opção `-spc`, o passo correspondente ao comando **Analyze** não necessita de ser executado na ferramenta de síntese.

(b) Através do comando de **Read**

O comando **Read** lê uma descrição VHDL de um circuito para a ferramenta de síntese num único passo. Este comando realiza os passos equivalentes aos comandos **Analyze** e **Elaborate** sem no entanto colocar a descrição analisada em nenhuma biblioteca VHDL.

A leitura com sucesso da descrição VHDL do circuito é indicada pelo aparecimento de um icon na ferramenta de síntese como o apresentado na figura 7.

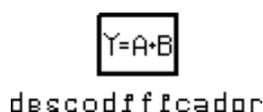


Figura 7: Icon que representa a descrição VHDL do circuito.

#### 4. Caracterização do circuito e imposição de restrições

Para que a ferramenta de síntese possa otimizar o circuito de acordo com as características e/ou restrições impostas ao circuito, estas devem ser indicadas à ferramenta através das várias opções do menú **Attributes**.

Se ao circuito não forem impostas restrições, o resultado de síntese apresentará um circuito otimizado em área.

#### 5. Síntese do circuito

Para realizar a síntese do circuito selecciona-se o icon que representa o circuito pretendido, e escolhe-se a opção **Design Optimization** do menú **Tools**. Na janela criada, onde podem ser seleccionadas várias opções relacionadas com a optimização

do circuito, seleccione o botão OK. Na nova janela criada é indicada a evolução do processo de síntese. Para fechar esta janela seleccione o botão Cancel.

Se o processo de síntese do circuito for realizado com sucesso, o ícone que representa o circuito é alterado para uma porta lógica (ver figura 8).

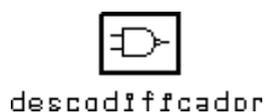


Figura 8: Ícone que representa o circuito sintetizado.

## 6. Visualização do circuito e geração de relatórios

Para visualizar o circuito sintetizado deve-se realizar um duplo *click*, primeiro sobre o ícone do circuito (para visualizar o símbolo que representa o circuito) e depois sobre o seu símbolo (para visualizar o esquema lógico do circuito). Na figura 9 apresenta-se o circuito sintetizado para o decodificador.

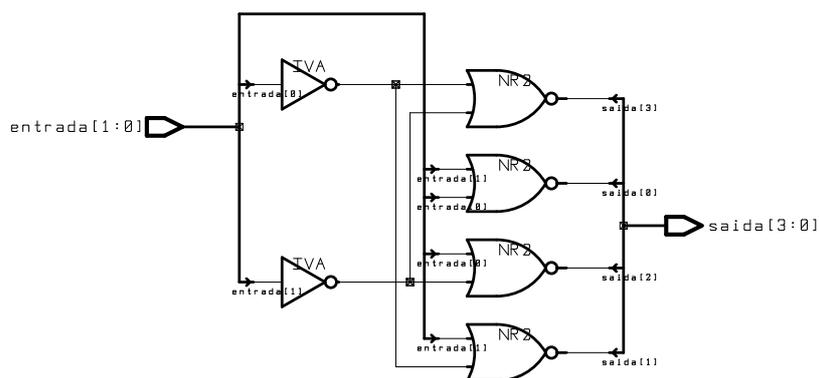


Figura 9: Circuito lógico sintetizado para o decodificador.

Alguns relatórios sobre o circuito podem ser obtidos através da selecção da opção Report do menu Analysis. A janela criada permite escolher o tipo de relatório pretendido.

## 7. Gravar e sair da ferramenta

Antes de sair da ferramenta de síntese o circuito sintetizado deve ser guardado em disco. Para gravar o circuito seleccione do menu File a opção Save As. Na janela criada deve ser indicado o nome do ficheiro em que o circuito vai ser gravado e o seu formato. Os vários formatos existentes podem ser seleccionados através do botão que apresenta por defeito o valor DB (formato interno da Synopsys).

Finalmente, para sair da ferramenta de síntese seleccione do menu File a opção Quit.