VHDL

Simulação & TestBench

4 de outubro de 2015

Rafael Corsi Ferrão - IMT

rafael.corsi@maua.br
http://www.maua.br

CONTEÚDO

CONTEÚDO

VERIFICAÇÃO

Dois tipos de verificação podem ser realizadas:

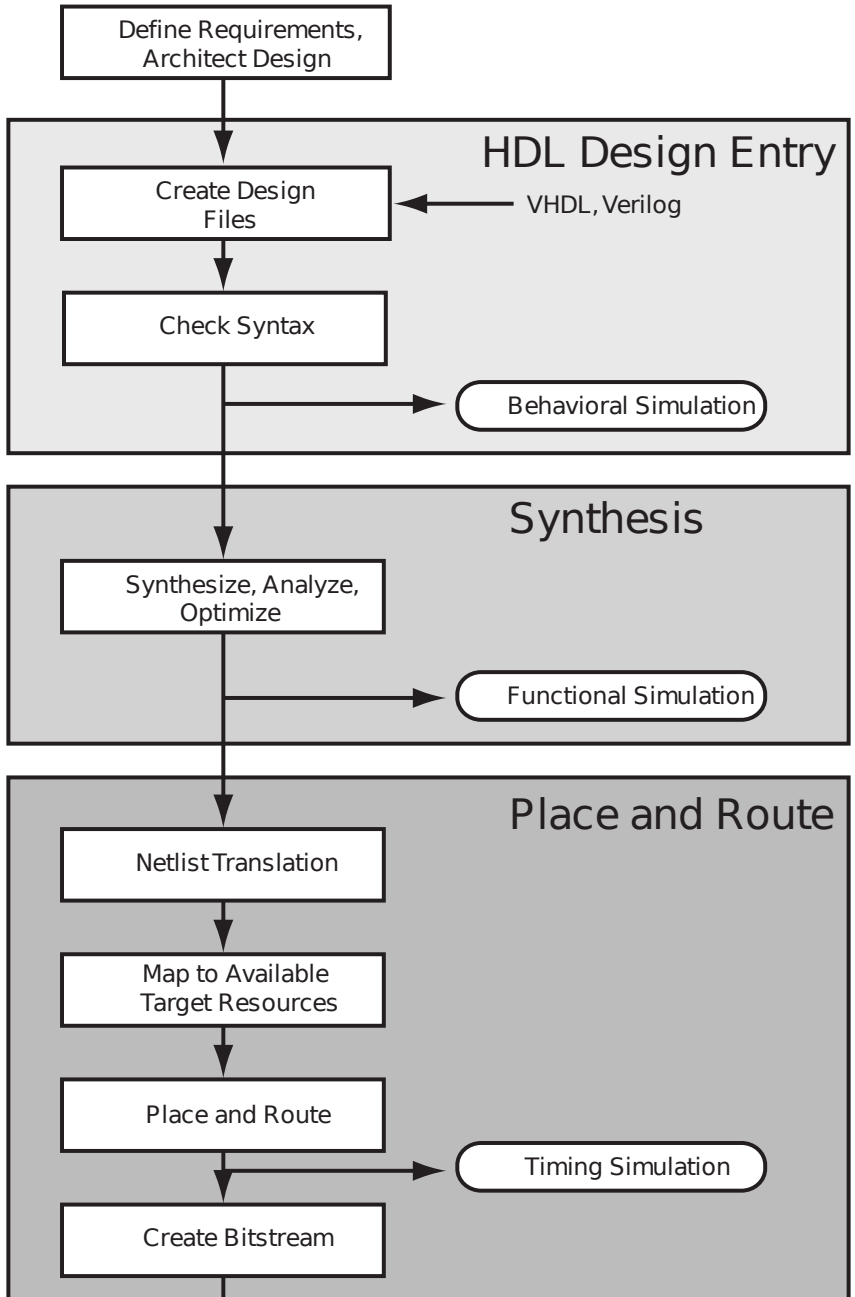
- Embarcar o projeto na pastilha e testar
  - processo demorado devido ao tempo de copilação
  - sem acesso aos sinais internos
  - verificação 100% real
- Simulação
  - processo rápido
  - com acesso aos sinais internos e externos
  - necessita de um código para exitar a entidade sob teste
  - não necessariamente 100% real

SIMULAÇÃO

A simulação pode ser realizada em três estágios diferentes do projeto :

- RTL
  - Verificamos aqui o funcionamento do código (simulação comportamental)
- Síntese
  - A simulação nesse estágio já leva em consideração os elementos internos da FPGA (simulação funcional)
- Place & route (Implementação)
  - No último estágio de simulação além dos elementos é verificado a questão de tempo (simulação de tempo)

SIMULAÇÃO


Define Requirements, Architect Design
HDL Design Entry
Create Design Files
VHDL, Verilog
Check Syntax
Behavioral Simulation
Synthesis
Synthesize, Analyze, Optimize
Functional Simulation
Place and Route
Netlist Translation
Map to Available Target Resources
Place and Route
Timing Simulation
Create Bitstream

CONTEÚDO

ARQUIVOS DE SIMULAÇÃO

Existem dois principais diferentes configurações de simulação :

- waveform
  - forma de onda criada via interface gráfica - difícil manutenção
- testbench
  - forma de onda criada via código VHDL - fácil manutenção

Definição

TestBench: ou cenário de testes é uma metodologia de verificação do projeto fornecendo uma forma de excitação dos sinais via código VHDL

CONTEÚDO

Os seguintes comandos são dedicados a simulação :

- WAIT
- WAIT ON
- WAIT FOR
- WAIT ON

e o seguinte tipo de dado :

- TIME

WAIT

Os comandos Waits são sequenciais e devem ser utilizados somente dentro de processos.

WAIT

O comando WAIT suspende a execução de um processo

WAIT

Os comandos Waits são sequenciais e devem ser utilizados somente dentro de processos.

WAIT

O comando WAIT suspende a execução de um processo

WAIT ON

Fornece um mecanismo similar a lista de sensibilidade

WAIT

Os comandos Waits são sequenciais e devem ser utilizados somente dentro de processos.

WAIT

O comando WAIT suspende a execução de um processo

WAIT ON

Fornece um mecanismo similar a lista de sensibilidade

WAIT UNTIL

O processo é suspendido enquanto a expressão booleana não for satisfeita

WAIT

Os comandos Waits são sequenciais e devem ser utilizados somente dentro de processos.

WAIT

O comando WAIT suspende a execução de um processo

WAIT ON

Fornece um mecanismo similar a lista de sensibilidade

WAIT UNTIL

O processo é suspendido enquanto a expressão booleana não for satisfeita

WAIT FOR

O processo é suspenso por um período de tempo

EXEMPLO

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity tb_wait is
end tb_wait;

architecture Behavioral of tb_wait is

---------------------------
-- sinais
---------------------------
    signal sig1  : std_logic := '0';
    signal sig2  : std_logic_vector(1 downto 0) := (others => '0');
    signal sig3  : integer := 0;

begin
```

EXEMPLO - CONTINUAÇÃO

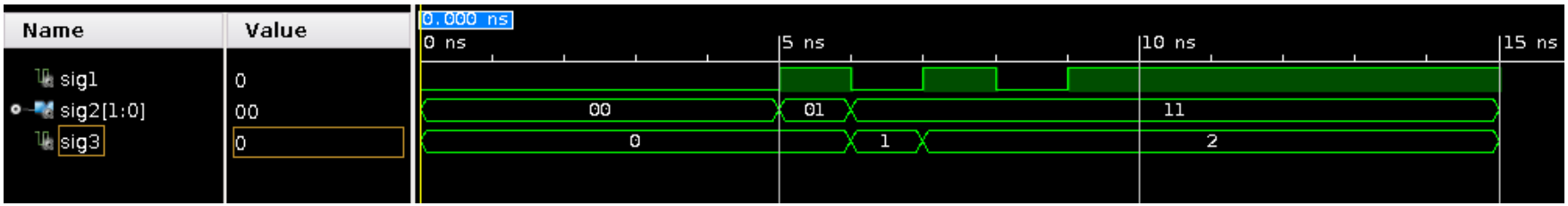
```
sig1_p : process
begin
    sig1 <= '0';
    wait for 5 ns;
    sig1 <= '1';
    wait for 1 ns;
    sig1 <= '0';
    wait for 1 ns;
    sig1 <= '1';
    wait for 1 ns;
    sig1 <= '0';
    wait for 1 ns;
    sig1 <= '1';
    wait;
end process;

sig2_p : process
begin
    sig2 <= "00";
    wait on sig1;
    sig2 <= "01";
    wait on sig1;
    sig2 <= "11";
    wait;
```

```
sig3_p : process
begin
    sig3 <= 0;
    wait until sig2 = "11";
    sig3 <= sig3 + 1;
    wait for 1 ns;
    sig3 <= sig3 + 1;
    wait;
end process;

end Behavioral;
```

WAVEFORM GERADO



Note que usamos um processo sem lista de sensibilidade, isso significa que o processo será executado sempre, sem necessitar da alteração de nenhum sinal

É comum no teste de um projeto a necessidade de geração de um estímulo para o clock, segue exemplo de como gerarmos esse sinal :

```
--------------------------
-- clk
--------------------------
process
begin
  clk <= not clk;
  wait for 1 ns; -- meio periodo
  clk <= not clk;
  wait for 1 ns; -- meio periodo
end process;
```

Podemos definir uma constante para facilitar a geração do clock :

```
architecture tb of tb_wait is

  signal half_period : time := 1 ns;

begin

-- clk
process
begin
  clk <= not clk;
  wait for half_period ns; -- meio periodo
  clk <= not clk;
  wait for half_period ns; -- meio periodo
end process;
```

CLK

CONTEÚDO

FERRAMENTAS DE SIMULAÇÃO

- ModelSim Mentor: É a ferramenta de simulação mais completa do mercado suporta praticamente todos os fabricantes e todas e possui integrabilidade com as ferramentas de desenvolvimento
- Vivado Simulator : ferramenta integrada no ambiente Vivado, simula somente FPGAs Xilinx

BIBLIOTECAS ESPECÍFICAS DE SIMULAÇÃO

Caso o projeto possua alguma instância de uma célula específica da Xilinx (como por exemplo a simulação de sinais LVDS) necessitamos incluir uma nova biblioteca que possa modelos de simulação dessas instâncias:

```
use UNISIM.VComponents.all;
```

COMENTÁRIOS

- TestBenchs podem ser tão ou mais complexos que o próprio projeto, não só exitando a entidade sob teste mas também verificando suas respostas.
- projetos grandes/críticos possuem equipes exclusivas para escreverem esses testes
- é bastante usual a escrita e leitura de arquivos de texto no ambiente de simulação.
- nunca deixe de testar suas entidades, faça um TB individual para cada entidade e depois para o conjunto de componentes (toplevel)