



Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Curso de Graduação em Ciências da Computação



Sistemas Digitais

INE 5406


Aula 5-P

Testbenchs

Profs. José Luís Güntzel , Rafael Luiz Cancian

{guntzel,cancian}@inf.ufsc.br

Testbenchs

- 
- Há várias formas de verificar o funcionamento de um projeto de sistema digital.
 - A simulação usando testbechs é uma das formas mais comuns.
 - Um testbench é uma especificação VHDL que é simulada por um simulador VHDL.

Testbenchs

- Um testbench é uma descrição em VHDL que contém:

Uma entidade de teste,

Um componente sob teste (UUT),

Um gerador de estímulos para a UUT,

Ferramentas para monitorar respostas da UUT.

Testbenchs

- A entidade de teste é uma entidade vazia, sem sinais de entrada e saída.
- Na arquitetura da entidade de teste é instanciada a entidade sob teste (UUT), que é o componente que será simulado.
- Na arquitetura também são especificados sinais internos que correspondem ao port da UUT e os estímulos para esses sinais

Testbenchs

- Usando VHDL:

Estímulos complexos podem ser gerados mais facilmente,

Construções específicas para simulação podem ser usadas para monitorar as respostas da UUT e fornecer um feedback da simulação ao projetista.

Construções VHDL para Testbenchs


• Assert

Testa uma condição booleana. Se for falsa, apresenta uma mensagem na tela de simulação.

A severidade da não-conformidade pode ser definida como “note”, “warning”, “error” ou “failure”.

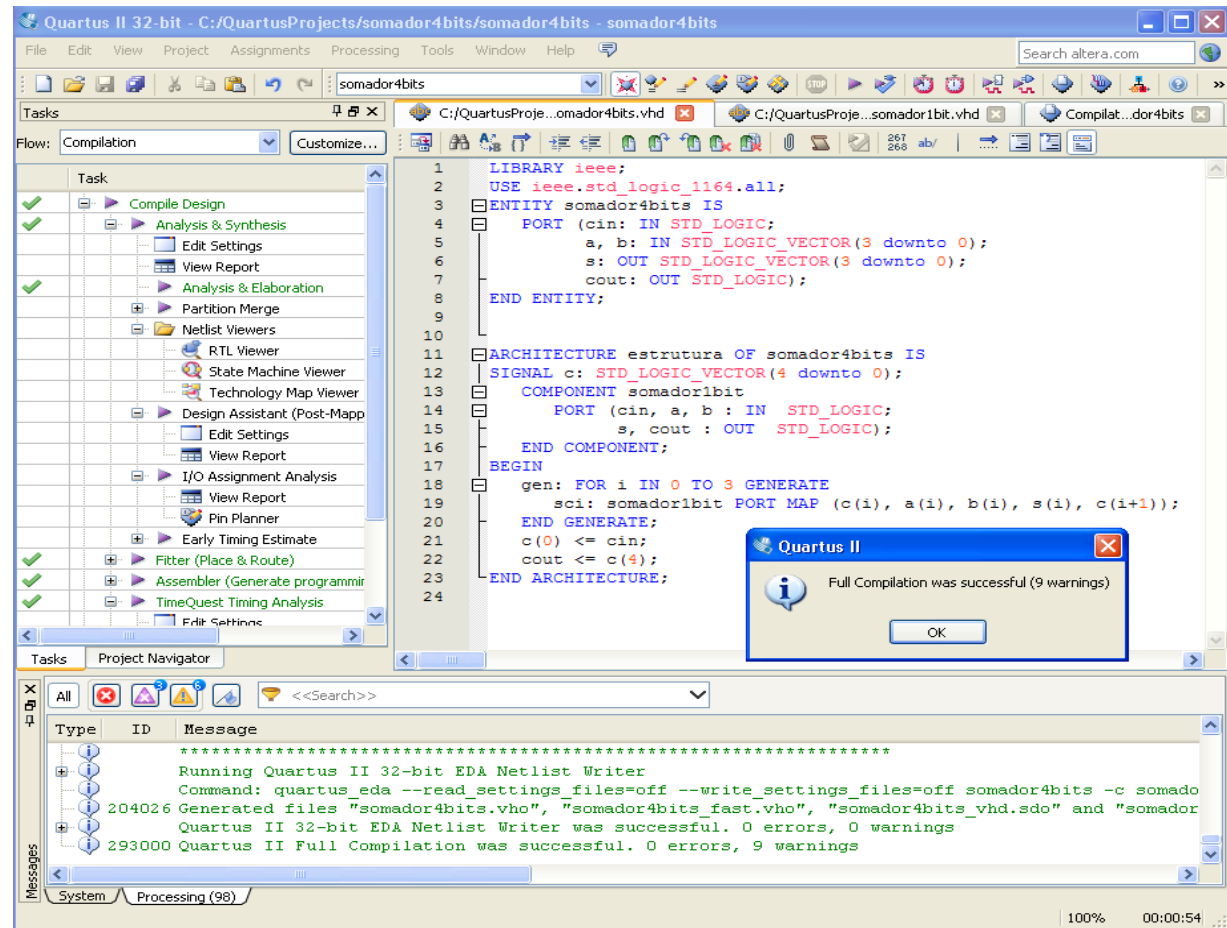
```
CHECK_SETUP: process (CLK, D)
begin
    if (CLK'event and CLK = '1') then
        assert D'stable(SETUP_TIME)
            report "Setup Violation..."
            severity warning;
    end if;
end process;
```

**Verifica se as
restrições temporais
foram satisfeitas**



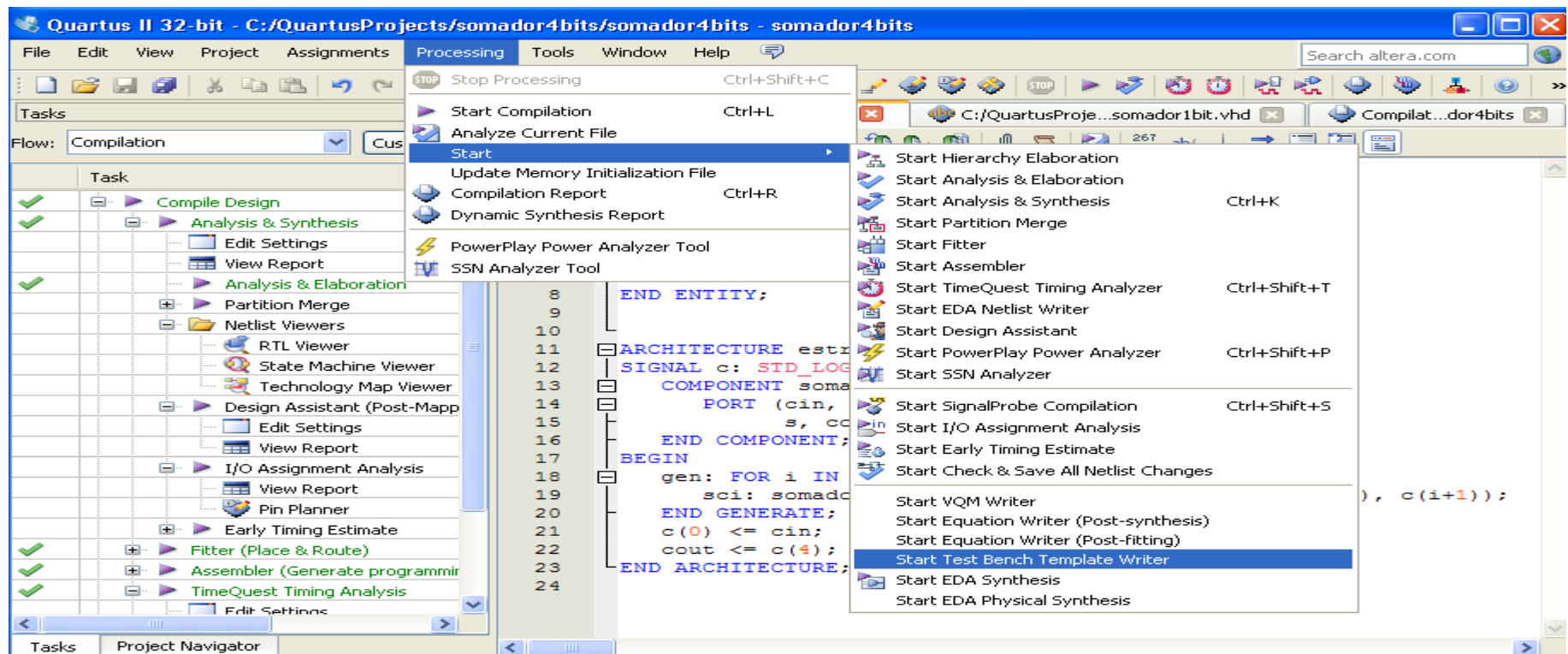
Roteiro para criação de testbenchs

- Crie seu projeto normalmente e compile-o com sucesso.
- Exemplo: somador de 4 bits (Aula 10P)



Roteiro para criação de testbenchs

- Acesse o menu “Processing” → “Start” → “Start Test Bench Template Writer”



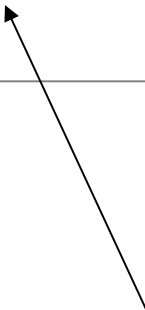
Roteiro para criação de testbenchs

- O arquivo gerado tem o nome da entidade top-level e extensão “vht” (Test Bench Output File), e fica armazenado na pasta “simulation/modelsim”.
- Acesse o menu “File” → “Open”, selecione arquivos do tipo “*.vht” e abra o arquivo “simulation/modelsim/somador4bits.vht”

Estrutura do testbench

- O testbench criado tem a seguinte estrutura:

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY somador4bits_vhd_tst IS  
END somador4bits_vhd_tst;
```



**Entidade de teste vazia,
sem sinais de entrada ou
De saída.**

Estrutura do testbench

```
ARCHITECTURE somador4bits_arch OF somador4bits_vhd_tst IS
```

```
-- constants
```

```
-- signals
```

```
SIGNAL a : STD_LOGIC_VECTOR(3 DOWNT0 0);
```

```
SIGNAL b : STD_LOGIC_VECTOR(3 DOWNT0 0);
```

```
SIGNAL cin : STD_LOGIC;
```

```
SIGNAL cout : STD_LOGIC;
```

```
SIGNAL s : STD_LOGIC_VECTOR(3 DOWNT0 0);
```

```
COMPONENT somador4bits
```

```
PORT (
```

```
  a : IN STD_LOGIC_VECTOR(3 DOWNT0 0);
```

```
  b : IN STD_LOGIC_VECTOR(3 DOWNT0 0);
```

```
  cin : IN STD_LOGIC;
```

```
  cout : OUT STD_LOGIC;
```

```
  s : OUT STD_LOGIC_VECTOR(3 DOWNT0 0)
```

```
);
```

```
END COMPONENT;
```

```
BEGIN
```

INE/CTC/UFSC

Sistemas Digitais - semestre 2011/1

slide 10P.11

**Sinais internos
correspondentes
aos da entidade
sob teste**

**Um componente
que é a entidade
sob teste (UUT),
O top-level do
projeto sendo
simulado.**

Prof. José Luís Güntzel

Estrutura do testbench

```
END COMPONENT;  
BEGIN
```

```
    i1 : somador4bits  
    PORT MAP (  
-- list connections between master ports and signals  
        a => a,  
        b => b,  
        cin => cin,  
        cout => cout,  
        s => s  
    );
```

Uma instância da entidade sob teste (UUT), chamada de “i1” com port map para sinais internos com os mesmos nomes.

Estrutura do testbench

```
init : PROCESS
-- variable declarations
BEGIN
    -- code that executes only once
WAIT;
END PROCESS init;
```

```
always : PROCESS
-- optional sensitivity list
-- ( )
-- variable declarations
BEGIN
    -- code executes for every event on sensitivity list
WAIT;
END PROCESS always;
END somador4bits_arch;
```

Processos para gerar estímulos para a entidade sob teste, e que podem conter construções VHDL como FOR, IF, CASE, etc para estímulos mais complexos e também comandos como ASSERT para monitorar as respostas da UUT.

Roteiro para criação de testbenchs

- Modifique os processos para geração de estímulos, fazendo com que

Todas as combinações possíveis de entradas sejam geradas ao somador4bits.

Uma mensagem de erro seja apresentada caso o resultado não seja o esperado, em qualquer combinação.

Roteiro para criação de testbenchs

```
always : PROCESS  
BEGIN
```

```
    fcin: for var_cin in 0 to 1 loop  
        if var_cin = 0 then cin <= '0';  
        else cin <= '1';  
        end if;
```

```
    fa: for var_a in 0 to 2**4-1 loop
```

```
        a <= std_logic_vector(to_unsigned(var_a, a'length));
```

```
    fb: for var_b in 0 to 2**4-1 loop
```

```
        b <= std_logic_vector(to_unsigned(var_b, b'length));
```

```
        wait for 15ns;
```

```
        assert(s=std_logic_vector(to_unsigned(var_a+var_b+var_cin,s'length)))
```

```
            report "Erro na soma" severity error;
```

```
    end loop;
```

```
end loop;
```

```
end loop;
```

```
END PROCESS always;
```

```
END somador4bits_arch;
```

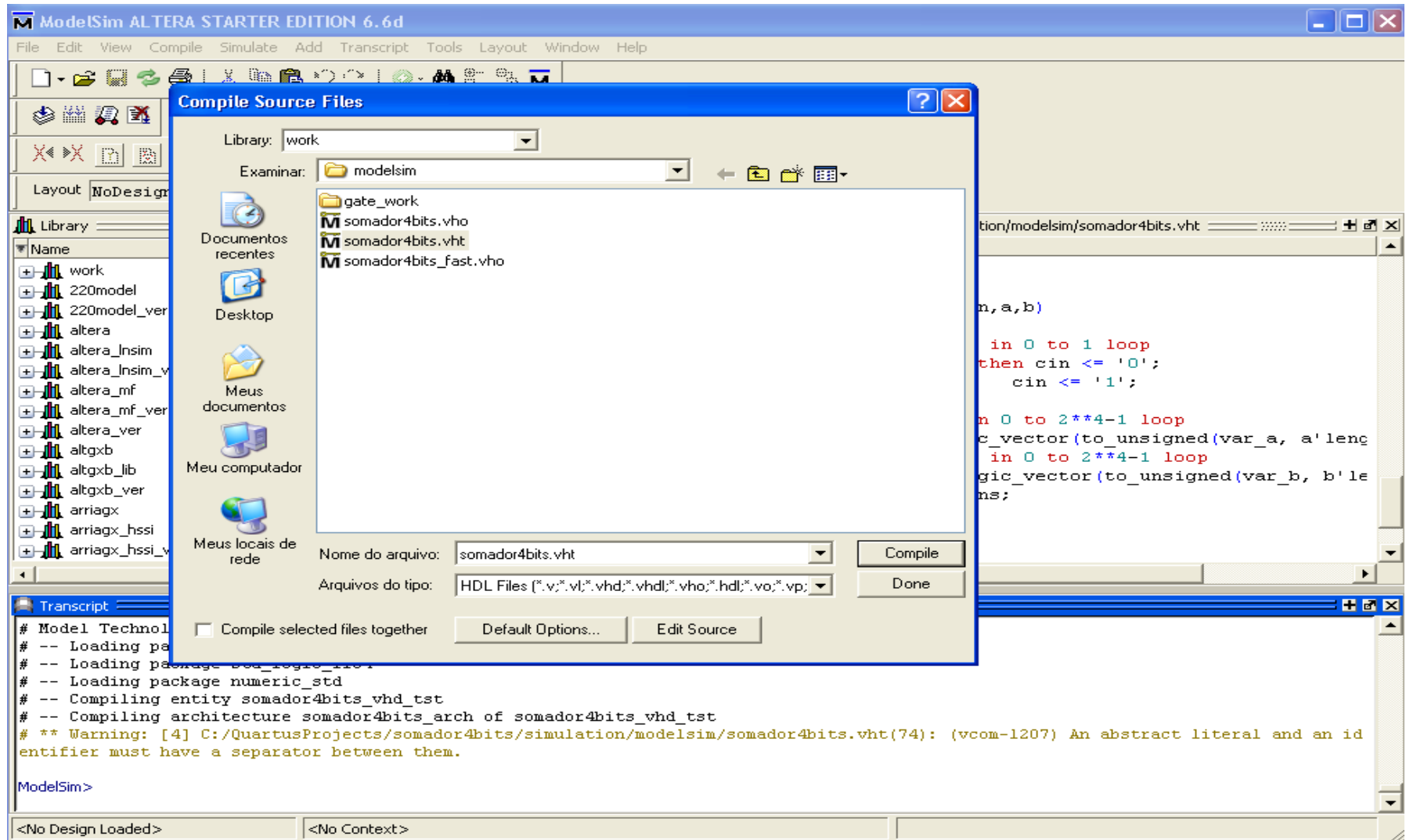
É preciso converter
de “integer” para
“std_logic_vector”



Roteiro para criação de testbenchs

- Salve o arquivo de testbench e invoque o modelsim-altera.
- Assim que o modelsim-altera abrir, selecione o menu “Compile” → “Compile”, e então selecione o arquivo de testbench “somador4bits.vht” e pressione “Compile”.
- Se o testbench estiver correto, ele será compilado com sucesso. Fique atento ao log que aparece.

Roteiro para criação de testbenchs

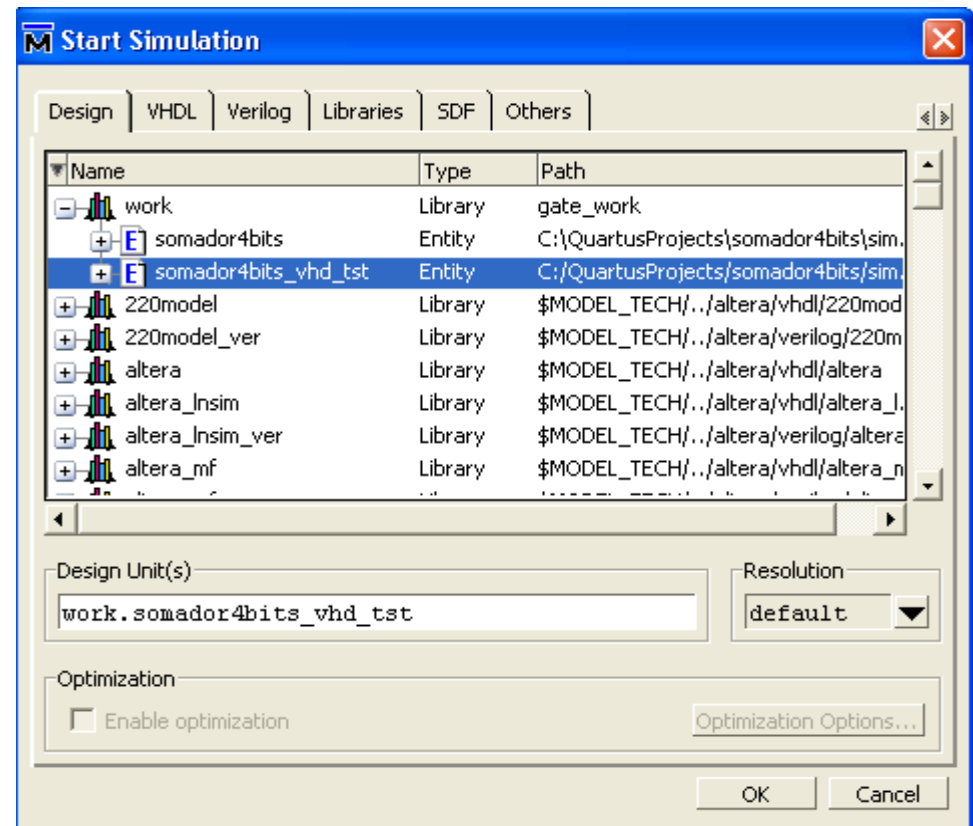


Roteiro para criação de testbenchs

- Se a compilação foi bem sucedida, pressione “Done”. Caso contrário, volte ao Quartus e corrija os erros de sintaxe”.
- Acesse o menu “Simulate” → “Start Simulation”.

Roteiro para criação de testbenchs

- Na janela que aparece, selecione “work” e então “somador4bits_vhd_tst”, que é a entidade de teste, e não “somador4bits”, que é a top-level do projeto.

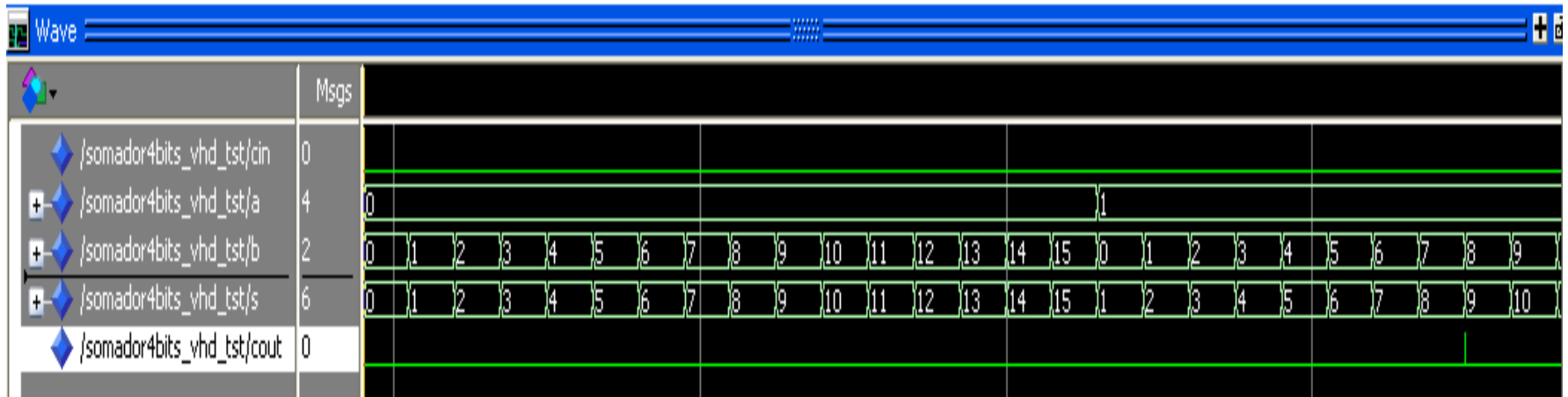


Roteiro para criação de testbenchs

- Proceda com a simulação normalmente, incluindo os sinais na forma de onda e ajustando as opções através do menu
“Simulate” → “Runtime Options”
- E então executando a simulação com
“Simulate → “Run” → “Run 100”
- Só não é necessário criar um arquivo de estímulos e executá-lo, pois os estímulos estão no próprio testbench.

Roteiro para criação de testbenchs

- O diagrama de forma de onda mostra os resultados da simulação, e fica visível a geração de todas as combinações possíveis dos sinais de estímulos,



Roteiro para criação de testbenchs