

Aula 02 - Verificação de Hardware 1/3 - Test-bench

Prof. Sergio R. M. Canovas

PCS - Departamento de Engenharia de Computação e Sistemas Digitais
Escola Politécnica da Universidade de São Paulo

Agosto, 2020

Agenda

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

1 Introdução

2 Simulação

3 Exemplo: Test-bench de somador de 4 bits

Ao final da aula você saberá:

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- O que é a etapa de verificação de projetos de sistemas digitais, bem como sua importância;
- O que é a técnica de simulação;
- O que é um *test-bench*;
- Como construir um *test-bench* em VHDL, de forma programática, para um circuito combinatório simples;
- Como executar a simulação de um *test-bench* utilizando a ferramenta GHDL.

Introdução

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

Introdução

Introdução

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- **Verificação** é uma das etapas chave em um projeto de sistema digital [1].7;
- Ao invés de testar um dispositivo real, a verificação consiste em analisar e tirar conclusões a partir de modelos computacionais deste dispositivo antes de fabricá-lo;
- Não é viável nem rápido construir protótipos físicos a todo passo de evolução de um projeto apenas para verificar se ele está correto e funciona conforme o esperado;

Introdução

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Sistemas de segurança crítica (*safety critical*) como equipamentos médicos podem causar perda de vidas se não se comportarem de acordo com sua especificação funcional;
- Mesmo em sistemas não críticos, falhas após a entrega resultam em *recalls* de produtos e, portanto, prejuízos financeiros e de reputação para a empresa;
- Conclusão: a necessidade de garantir o correto funcionamento de um sistema influencia a metodologia de seu projeto;

Introdução

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Em cada passo do projeto, o projetista precisa ter certeza de que o modelo reflete a intenção original e executa o comportamento pretendido de forma correta, eficiente e segura;
- Isso é obtido pela atividade de **verificação**;

Introdução

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Um **modelo** é um artefato correspondente a uma abstração de uma realidade, existente ou a ser construída, que captura os detalhes correspondentes a um certo **nível de abstração**;
- Um diagrama lógico, por exemplo, é um modelo de um circuito lógico;
- Em seu nível de abstração, ele é suficiente para identificar as entradas, saídas e operações lógicas realizadas ao longo dos caminhos do circuito, mas não captura os detalhes de operação dos transistores que compõem as portas lógicas;

Introdução

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- A verificação pode ser realizada em modelos de qualquer nível de abstração do sistema (sistema, processador, lógico e de circuito);
- Existem muitos tipos de modelos à nossa disposição, cada um mais adequado (ou não) para uma certa etapa e visão de projeto;
- Certos tipos de modelos podem ser simulados com propósitos de estudar e avaliar diferentes aspectos do dispositivo projetado;

Introdução

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- As técnicas de verificação de modelos podem ser divididas em dois grupos:
 - 1 Simulação;
 - 2 Métodos formais.
- Nesta aula iniciaremos a técnica de **simulação**.

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

Simulação

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Métodos de verificação baseados em simulação consistem em checar um conjunto de propriedades no modelo de implementação que se deseja verificar;
- O **estímulo** corresponde à entrada a ser aplicada ao modelo de implementação a ser simulado;
- O comportamento esperado é checado pelo **monitoramento** das saídas do modelo simulado;
- A simulação é o método de verificação de sistemas mais utilizado [1].7;

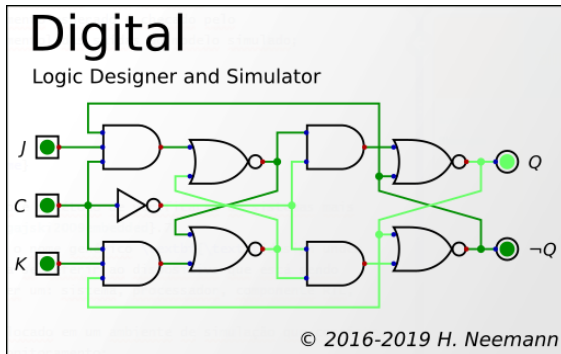
Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Um exemplo de *software* de **simulação interativa** é o Digital;



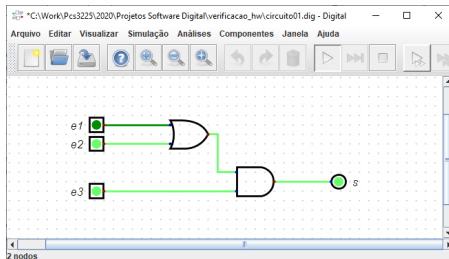
Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Os circuitos são desenhados em um ambiente visual;
- O botão *play* coloca o circuito em modo de execução;
- O usuário clica nas entradas para alterar seu valor lógico, e visualiza o valor lógico de cada sinal intermediário e também das saídas;



Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- A simulação interativa é um meio muito didático e útil para circuitos pouco complexos;
- Uma desvantagem é a dificuldade para efetuar simulações longas e repetitivas de forma automatizada;
- É necessário haver um suporte, de algum modo, para *scripts* de teste;

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Utiliza-se o nome genérico *design under test* (DUT) para se referir ao dispositivo que está sendo simulado. Pode ser um: sistema, processador, componente RTL, etc.;
- O DUT é colocado em um ambiente de simulação que consiste em estímulos e monitoramento;

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

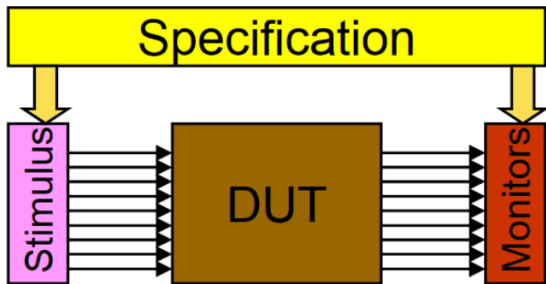


Figura: Ambiente de Simulação - Fonte: [1].7 - Figura 7.1

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Os estímulos correspondem a um conjunto de valores que são aplicados como entradas no DUT;
- Essas entradas disparam eventos e computações dentro do DUT, conforme descrito pelo seu modelo;
- Esses eventos e computações resultam em valores de saída;
- É tarefa do simulador manter registro desses eventos, processá-los conforme determina a semântica do modelo e calcular os valores de saída adequadamente;

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- A principal responsabilidade do elemento monitor é garantir que os valores de saída correspondam aos valores esperados;
- Durante a simulação, essas saídas podem variar com o tempo. Assim, o monitor deve conhecer todos os valores esperados de saída com seu respectivo tempo de ocorrência;
- O monitor deve indicar um erro se o valor obtido na simulação não bater com o esperado, o que será avaliado pelo projetista;

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Um par **estímulo / saída esperada** é chamado de caso de teste (*test-case*);
- Um conjunto de casos de teste compõe o ***test-bench***, ou **bancada de testes**, que consiste em um modelo de um ambiente de simulação do DUT;

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

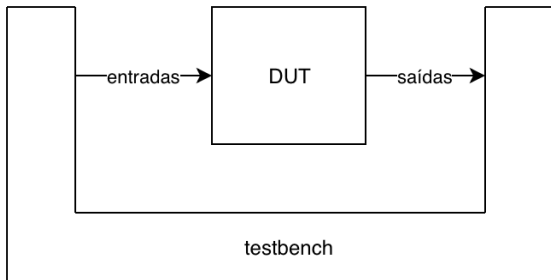


Figura: *Test-Bench* - Fonte: https://balbertini.github.io/vhdl_testbench-pt_BR.html

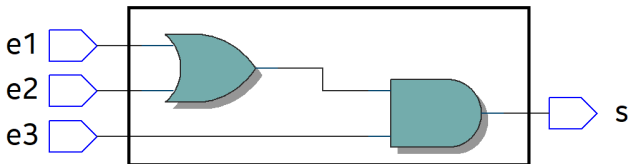
Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Retomando o exemplo simples de um circuito que implementa uma lógica combinatória com entradas $e1$, $e2$ e $e3$ e uma saída s dada por $s = (e1 \vee e2) \wedge e3$;



Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Para testar esse sistema, qual seria um caso de teste possível?

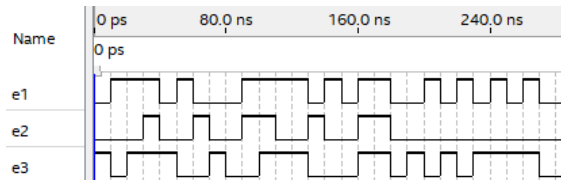
Simulação

Introdução

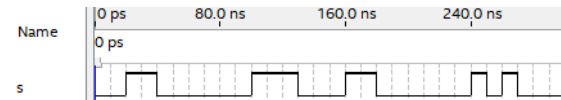
Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Poderíamos enviar o seguinte estímulo:



- E sabemos qual é a saída esperada:



Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Para simulá-lo, inicialmente precisamos de uma descrição VHDL do próprio sistema, que será nosso DUT;

```
1  entity MeuSistema is
2      port (
3          e1,e2,e3: in bit;
4          s: out bit
5      );
6  end MeuSistema;
7
8  architecture MeuSistema_Arch of MeuSistema is
9  begin
10     s <= (e1 or e2) and e3;
11 end;
```

- Mas como construir um *test-bench* para esse sistema?

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Lembre-se da figura do *slide* 21: o *test-bench* também pode ser visto como um sistema, englobando o DUT, sem entradas e sem saídas;
- Também podemos descrevê-lo em VHDL e simulá-lo;
- Ele envia estímulos ao DUT, coletando as saídas e comparando-as com os valores esperados (monitoramento);
- O *test-bench* é um *hardware* projetado para testar outro *hardware*;

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- O VHDL possui a instrução **assert ... report**, que permite verificar uma condição e emitir uma mensagem caso essa condição não seja satisfeita. Essa mensagem não tem propósitos de sintetização, mas sim de simulação;
- Podemos usar o **assert ... report** dentro de um **process** que envia os estímulos, alertando o usuário que está executando a simulação em caso de situações não esperadas;

- Esta seria a entidade do *test-bench* (vazia!):

```
1  entity MeuSistema_tb is
2  end MeuSistema_tb;
3
```

- Como seria sua *architecture*?

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Primeiramente, a *architecture* do *test-bench* instancia o DUT, mapeando (conectando) suas portas a sinais internos (linha 19):

```
4  architecture MeuSistema_tb_arch of MeuSistema_tb is
5
6      signal e1_tb: bit;
7      signal e2_tb: bit;
8      signal e3_tb: bit;
9      signal s_tb: bit;
10
11     component MeuSistema
12     port (
13         e1,e2,e3: in bit;
14         s: out bit
15     );
16     end component;
17 begin
18
19     ms0: MeuSistema port map (e1 => e1_tb, e2 => e2_tb, e3 => e3_tb, s => s_tb);
20
```

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- A *architecture* contém um *process* gerador de estímulos, que na simulação é executado sequencialmente, e devemos utilizá-lo para gerar os sinais de nosso interesse;
- O **wait for** instrui o simulador para aguardar um tempo de simulação, em que consideramos atrasos esperados, antes de fazer a verificação da saída do DUT;

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

```
21      GeraEstimulos: process
22      begin
23          e1_tb <= '0';
24          e2_tb <= '0';
25          e3_tb <= '0';
26          wait for 10 ns;
27          assert s_tb = '0' report "Saída inesperada" severity error;
28
29          wait for 20 ns;
30
31          e2_tb <= '1';
32          e3_tb <= '1';
33          wait for 10 ns;
34          assert s_tb = '1' report "Saída inesperada" severity error;
35
36          wait;    -- espera em definitivo, concluindo a simulação
37      end process;
38  end;
```

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- O simulador avisará se alguma situação não esperada aconteceu;
- Você pode ainda utilizar um visualizador de ondas para conferir os sinais em cada instante de tempo, facilitando a depuração;
- Experimente introduzir um defeito nesse sistema (ex. troque a porta OR por uma AND) e veja como o *test-bench* acusa o problema;
- O código apresentado não corresponde ao estímulo do *slide* 24. Tente criar um *test-bench* que reproduza aquele estímulo como exercício;

Simulação

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- O exemplo apresentado é simples demais, e ainda é combinatório, ou seja, a saída em um determinado instante depende apenas das entradas naquele instante (salvo atrasos de propagação);
- É fácil fazer um *test-bench* que preveja todas as situações possíveis, mas em um sistema de maior complexidade isso é inviável;
- Basta considerar um circuito sequencial com um número grande de entradas e saídas. As saídas dependem de todo o histórico das entradas;
- É importante que um *test-bench* seja eficiente em capturar possíveis *bugs* no modelo do DUT;

Exemplo: Test-bench de somador de 4 bits

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

Exemplo: Test-bench de somador de 4 bits

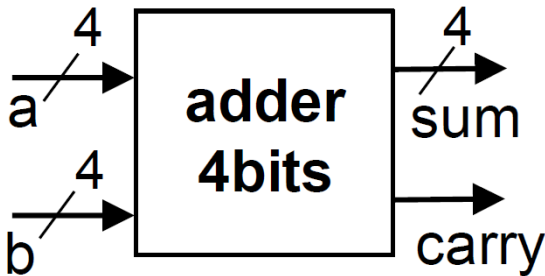
Exemplo: Test-bench de somador de 4 bits

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Exemplo: *test-bench* de um somador de 4 bits;
- Não é necessário conhecer a descrição VHDL do DUT para elaborar um *test-bench*, basta conhecer a entidade (entradas e saídas) e casos de teste;



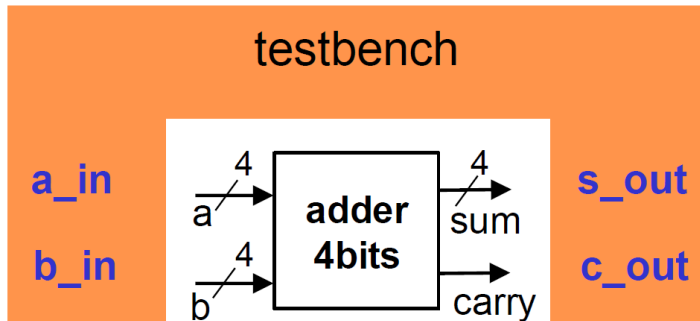
Exemplo: Test-bench de somador de 4 bits

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- O *test-bench* é um *hardware* que “envolve” o somador (DUT);



Exemplo: Test-bench de somador de 4 bits

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

```
-- Testbench para somador
```

```
library IEEE;
```

```
use IEEE.numeric_bit.all;
```

```
entity testbench is
```

```
-- Sempre vazio:
```

```
end testbench;
```

```
architecture testb of testbench is
```

```
-- DUT: component
```

```
component adder4bits is
```

```
port(
```

```
  a,b:  in   bit_vector  (3 downto 0);
```

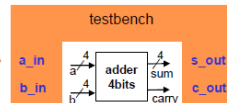
```
  sum:  out  bit_vector  (3 downto 0);
```

```
  carry: out  bit );
```

```
end component;
```

```
signal a_in, b_in, s_out:  bit_vector  (3 downto 0);
```

```
signal c_out :  bit ;
```



Exemplo: Test-bench de somador de 4 bits

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

```
-- Conectar DUT
DUT: adder4bits port map(a_in, b_in, s_out, c_out);

process -- Circuito com sequência de testes
begin
    a_in <= "0000";
    b_in <= "0000";
    wait for 1 ns; -- espera estabilizar e verifica saída
    assert(c_out & s_out = "00000") report "Fail 0+0" severity error;

    a_in <= "0001";
    b_in <= "0001";
    wait for 1 ns; -- espera estabilizar e verifica saída
    assert(c_out & s_out = "00010") report "Fail 1+1" severity error;

    a_in <= "1000";
    b_in <= "1000";
    wait for 1 ns; -- espera estabilizar e verifica saída
    assert(c_out & s_out = "10000") report "Fail 8+8" severity error;
```

Exemplo: Test-bench de somador de 4 bits

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

```
a_in <= "1111";  
b_in <= "0001";  
wait for 1 ns; -- espera estabilizar e verifica saída  
assert(c_out & s_out ="10000") report "Fail F+1" severity error;  
  
a_in <= "1110";  
b_in <= "0111";  
wait for 1 ns; -- espera estabilizar e verifica saída  
assert(c_out & s_out ="10101") report "Fail E+7" severity error;  
  
a_in <= "0111";  
b_in <= "0011";  
wait for 1 ns; -- espera estabilizar e verifica saída  
assert(c_out & s_out ="01010") report "Fail 7+3" severity error;
```

Exemplo: Test-bench de somador de 4 bits

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

```
a_in <= "1100";  
b_in <= "0101";  
wait for 1 ns; -- espera estabilizar e verifica saída  
assert(c_out & s_out ="10001") report "Fail C+5" severity error;  
  
-- Limpa entradas (opcional)  
a_in <= "0000";  
b_in <= "0000";  
  
-- Informa fim do teste  
assert false report "Test done." severity note;  
wait; -- Interrompe execução  
end process;  
end testb;
```


Exemplo: Test-bench de somador de 4 bits

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits

- Exercício: um *test-bench* para um somador de 64 bits que testa todas as combinações é viável? Explique o raciocínio.

Obrigado!



Universidade de São Paulo



DEPARTAMENTO DE ENGENHARIA DE
COMPUTAÇÃO E SISTEMAS DIGITAIS

PCS

Referências

Introdução

Simulação

Exemplo:
Test-bench de
somador de 4
bits



D. Gajski, S. Abdi, A. Gerstlauer, and G. Schirner.
Embedded System Design: Modeling, Synthesis and Verification.
Springer US, 2009.