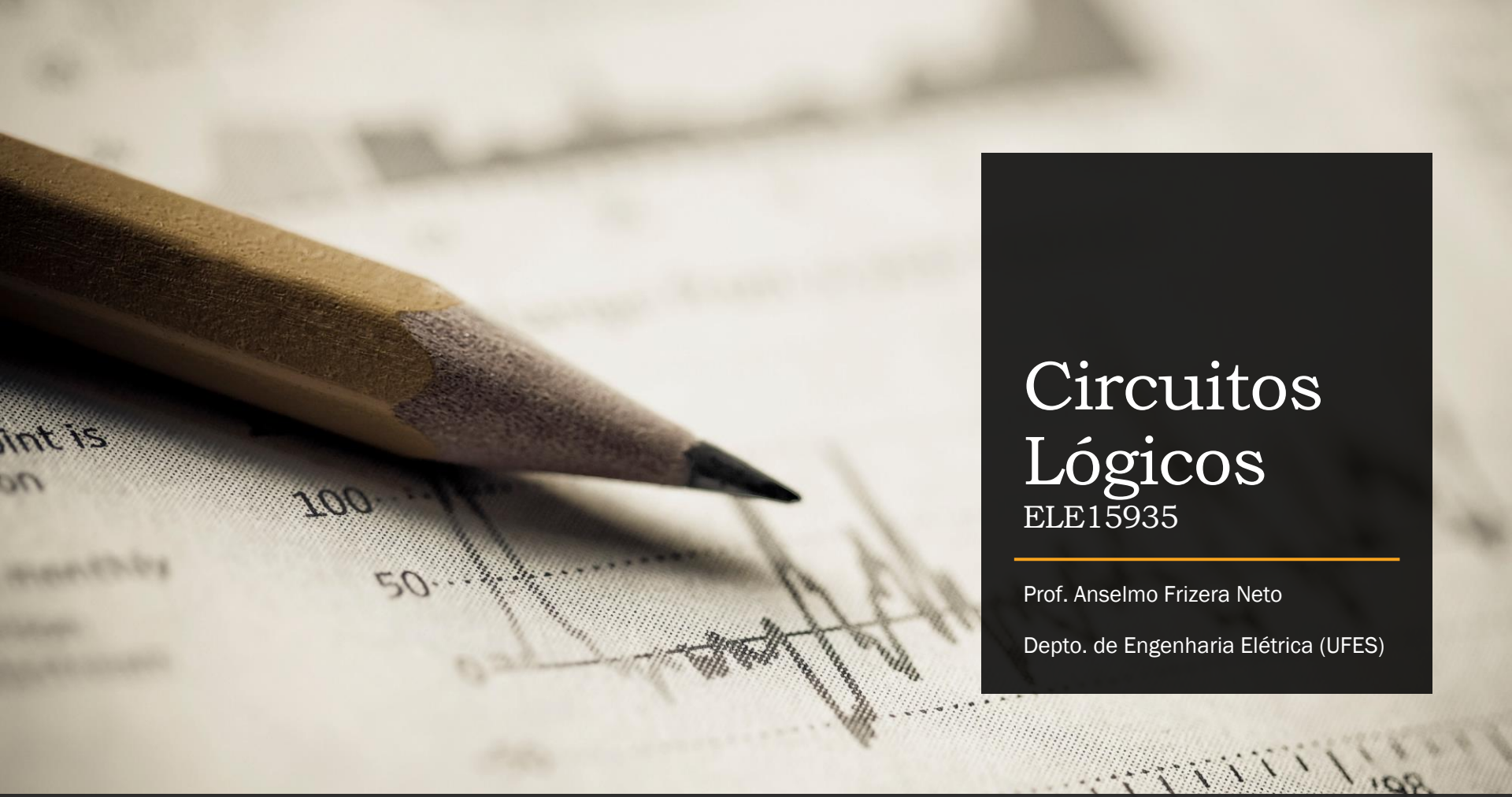


Circuitos Lógicos

ELE15935

Prof. Anselmo Frizera Neto

Depto. de Engenharia Elétrica (UFES)

Lab. 2. Apresentação da FPGA e *Testbench*

Tópicos da aula de hoje:

- Apresentação (resumida) da FPGA Xilinx Artix 7 XC7A100T CSG324C speed -1
- Apresentação da placa Nexys A7 FPGA Trainer Board
- Realização de *Testbench* para o comparador de 2 bits apresentado na aula passada

Bibliografia:

- F. Vahid, “Sistemas digitas” - Capítulos 7 e 9
- P. Chu, “FPGA Prototyping by VHDL Examples”

FPGA:
Macro células

Módulos construídos a nível de transistor

Incorporam componentes frequentemente usados no chip da FPGA

Blocos de propósito especial

- Módulos de memória
- Suporte de clock
- Multiplicadores combinacionais
- Interface serial de alta velocidade (como PCI e Gigabit ethernet controller)
- Processador (por exemplo, ARM A9)

FPGA

A FPGA usada neste curso é a Xilinx Artix 7 XC7A100T-1CSG324C disponível na placa Nexys A7 da Digilent

FPGA Xilinx Artix 7 XC7A100T CSG324C speed -1

Características

- 15,850 *Slices* de lógica programável, cada um com quatro LUTs de 6 entradas e 8 flip-flops
- 1,188 Kbits de *block* RAM
- Seis circuitos de gerenciamento de clock, com phase-locked loop (PLL)
- 240 DSP *slices*
- *Clock* interno com velocidade excedendo 450 MHz
- Conversor analógico-digital (XADC) interno dois canais, 1 MSPS

Product Variant	Nexys A7-100T
FPGA Part Number	XC7A100T-1CSG324C
Look-up Tables (LUTs)	63,400
Flip-Flops	126,800
Block RAM	1,188 Kb
DSP Slices	240
Clock Management Tiles	6

FPGA
Xilinx Artix 7 XC7A100T CSG324C speed -1

Artix-7 FPGA Feature Summary

Table 4: **Artix-7 FPGA Feature Summary by Device**

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices[2]	Block RAM Blocks[3]			CMTs[4]	PCIe[5]	GTPs	XADC Blocks	Total I/O Banks[6]	Max User I/O[7]
		Slices[1]	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)						
XC7A12T	12,800	2,000	171	40	40	20	720	3	1	2	1	3	150
XC7A15T	16,640	2,600	200	45	50	25	900	5	1	4	1	5	250
XC7A25T	23,360	3,650	313	80	90	45	1,620	3	1	4	1	3	150
XC7A35T	33,280	5,200	400	90	100	50	1,800	5	1	4	1	5	250
XC7A50T	52,160	8,150	600	120	150	75	2,700	5	1	4	1	5	250
XC7A75T	75,520	11,800	892	180	210	105	3,780	6	1	8	1	6	300
XC7A100T	101,440	15,850	1,188	240	270	135	4,860	6	1	8	1	6	300
XC7A200T	215,360	33,650	2,888	740	730	365	13,140	10	1	16	1	10	500

Notes:

1. Each 7 series FPGA slice contains four LUTs and eight flip-flops; only some slices can use their LUTs as distributed RAM or SRLs.
2. Each DSP slice contains a pre-adder, a 25 x 18 multiplier, an adder, and an accumulator.
3. Block RAMs are fundamentally 36 Kb in size; each block can also be used as two independent 18 Kb blocks.
4. Each CMT contains one MMCM and one PLL.
5. Artix-7 FPGA Interface Blocks for PCI Express support up to x4 Gen 2.
6. Does not include configuration Bank 0.
7. This number does not include GTP transceivers.

FPGA
Xilinx Artix 7 XC7A100T CSG324C speed -1

Algumas características da arquitetura do CLB incluem:

- Tabelas de look-up de 6 entradas
- Capacidade de memória dentro da LUT
- Funcionalidade de registro e registro de deslocamento
- As LUTs podem ser configuradas como uma LUT de 6 entradas (ROMs de 64 bits) com uma saída ou como duas LUTs de 5 entradas
- LUTs (ROMs de 32 bits) com saídas separadas, mas com endereços ou entradas lógicas comuns.
- Cada saída LUT pode ser opcionalmente registrada em um *flip-flop*.

FPGA
Xilinx Artix 7 XC7A100T CSG324C speed -1

Alguns dos pontos chave da arquitetura de gerenciamento de clock incluem:

- Buffers e roteamentos de alta velocidade para distribuição de clock com baixa distorção (*low-skew*)
- Síntese de frequência e deslocamento de fase
- Geração de clock com *jitter* e filtragem de *jitter*
- Possui 6 blocos de gerenciamento de relógio (*clock management tiles* - CMTs), cada um consistindo em um gerenciador de relógio de modo misto (*mixed-mode clock manager* - MMCM) e um *phase-locked loop* (PLL).

FPGA
Xilinx Artix 7 XC7A100T CSG324C speed -1

Destaques da funcionalidade DSP incluem:

- Processador de sinal de alta resolução (48 bits) com multiplicadores/acumuladores 25x18 bits em complemento de dois
- Pré-somador de economia de energia para otimizar aplicações de filtro simétrico
- Recursos avançados: *pipelining* opcional, ULA opcional e barramentos dedicados para cascateamento

Destaques da funcionalidade de entrada/saída incluem:

- Tecnologia SelectIO de alto desempenho com suporte para DDR3 de 1.866 Mb/s
- Capacitores de desacoplamento de alta frequência dentro do pacote para maior integridade do sinal
- Impedância controlada digitalmente que pode ser 3 estados para operação de E/S de baixa potência e alta velocidade

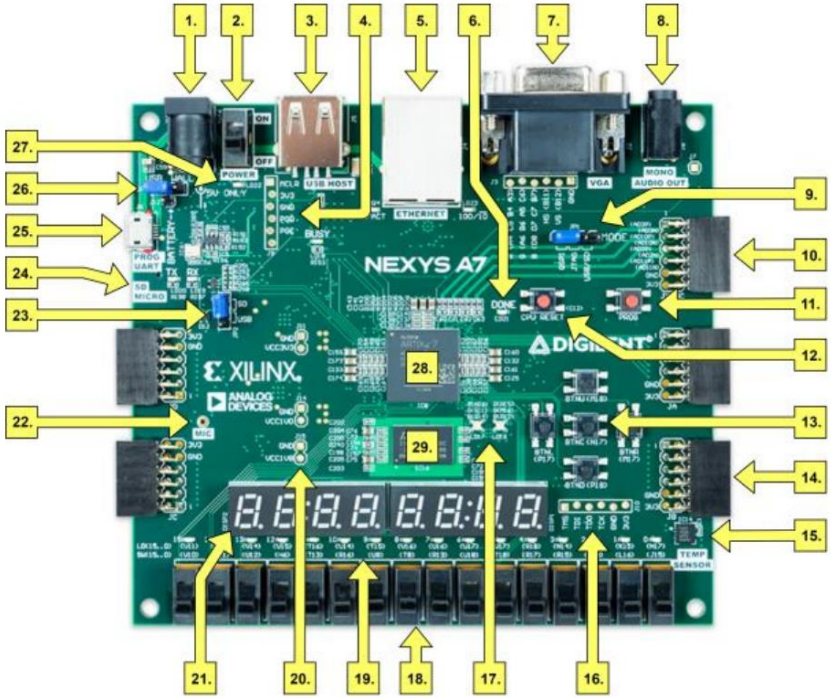
Destaques da arquitetura XADC (conversor analógico para digital) incluem:

- Conversor analógico-digital (ADCs) duplos de 12 bits 1 MSPS
- Opção de referência externa ou no chip
- Sensores de temperatura no chip (erro máximo de ±4°C) e fonte de alimentação (erro máximo de ±1%)
- Acesso JTAG contínuo a medições ADC

Nexys A7 FPGA Trainer Board

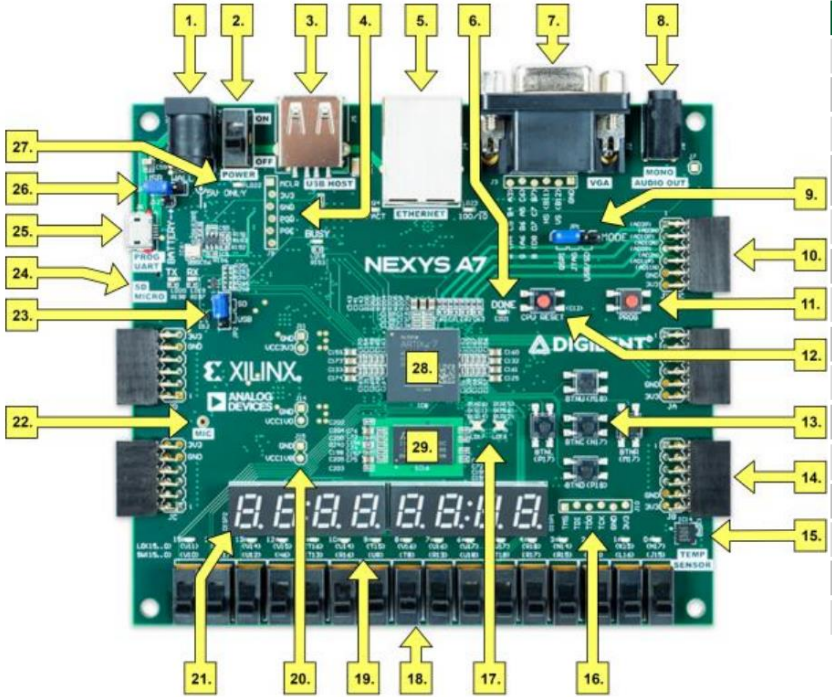
Esta é a placa Nexys A7 que contém a FPGA Xilinx Artix 7 XC7A100T-1CSG324C

Nexys A7 FPGA Trainer Board



Callout	Component Description
1	Power jack
2	Power switch
3	USB host connector
4	PIC24 programming port (factory use)
5	Ethernet connector
6	FPGA programming done LED
7	VGA connector
8	Audio connector
9	Programming mode jumper
10	Analog signal Pmod port (XADC)
11	FPGA configuration reset button
12	CPU reset button (for soft cores)
13	Five pushbuttons
14	Pmod port(s)
15	Temperature sensor

Nexys A7 FPGA Trainer Board



Callout	Component Description
16	JTAG port for (optional) external cable
17	Tri-color (RGB) LEDs
18	Slide switches (16)
19	LEDs (16)
20	Power supply test point(s)
21	Eight digit 7-seg display
22	Microphone
23	External configuration jumper (SD / USB)
24	MicroSD card slot
25	Shared UART/ JTAG USB port
26	Power select jumper and battery header
27	Power-good LED
28	Xilinx Artix-7 FPGA
29	DDR2 memory

Nexys A7 FPGA Trainer Board

Características da FPGA

- Clock interno com capacidade para 450MHz
- Conversor analógico-para-digital on-chip (XADC)
- Programável por JTAG e Flash

Memória

- 128MiB DDR2
- Serial Flash
- Slot de cartão micros

Alimentação

- Via USB ou fonte externa com tensão entre 4.5V e 5.5V

Nexys A7 FPGA Trainer Board

Áudio e Vídeo

- Saídas VGA12-bit para cor RGB além de hsync e vsync
- Saída de áudio PWM (*pulse-width modulated*)
- Microfone com saída digital em formato PDM (*pulse density modulated*)

USB e Ethernet

- 10/100 Ethernet PHY
- USB-JTAG circuitaria de programação
- USB-UART bridge
- USB HID (*Human interface device*) Host para mouse, teclado e *pendrive*

Nexys A7 FPGA Trainer Board

Dispositivos de entrada e saída simples

- 16 chaves (*slide switch*)
- 16 LEDs
- 2 LEDs RGB (Digilent strongly recommends the use of pulse-width modulation (PWM) when driving the tri-color LEDs
- 2 displays de 7 segmentos de 4 dígitos

Sensores Adicionais

- Acelerômetro de 3 eixos (usando protocolo SPI - *serial peripheral interface*)
- Sensor de temperatura (usando protocolo I^2C - *Inter-Integrated Circuit*)

Nexys A7 FPGA Trainer Board

Conectores de expansão

- Conector Pmod para sinais XADC
- 4 Conectores Pmod fornecendo 32 sinais de entrada e saída

A Placa Nexys A7 com a FPGA XC7A100T é compatível com o software Vivado® Design Suite bem como com o ISE® da Xilinx

A Xilinx oferece a versão WebPACK™ sem custo destes softwares

Nesse curso, vamos usar a versão 2017.4 disponível no seguinte link:

- https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html

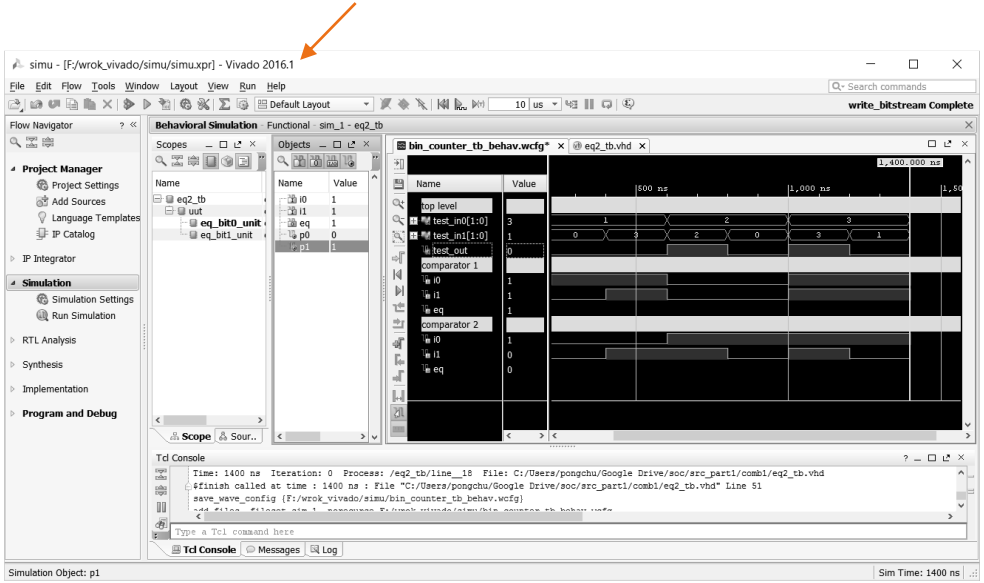
Mas como fazer para simular ou testar nosso código sem uma FPGA?

O Vivado Design Suite integra o simulador dentro do framework!

Tipos de simulações:

- comportamentais, pós-síntese e pós-implementação.

Janela com uma área de simulação típica:



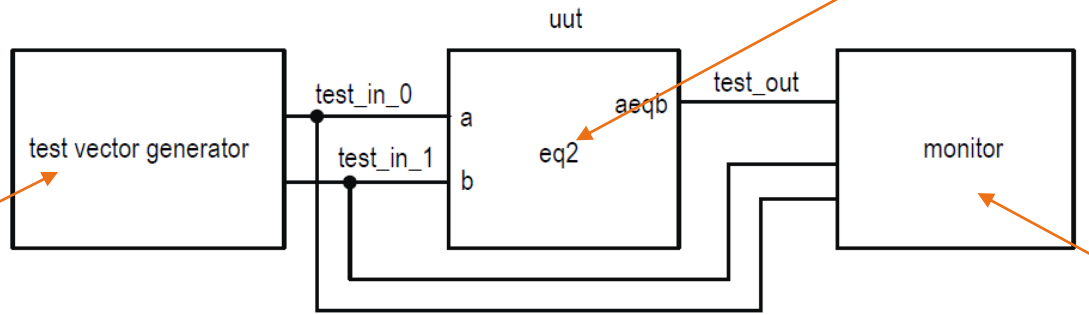
Testbench

Um testbench é um programa em VHDL que funciona como uma bancada de testes “virtual”

Testbench inclui:

- O circuito para ser testado (uut: unit under test)
- Os estímulos de entrada (por exemplo, algo como um gerador de função)
- Um monitor de saída para verificar o resultado (opcional)

Testbench pode ser executado por um simulador VHDL



Instanciação do componente eq2

Processo VHDL
Padrões de entrada para o teste

Forma de onda no simulador

Testbench para o comparador de 2 bits

Instanciando comparador de 2 bits

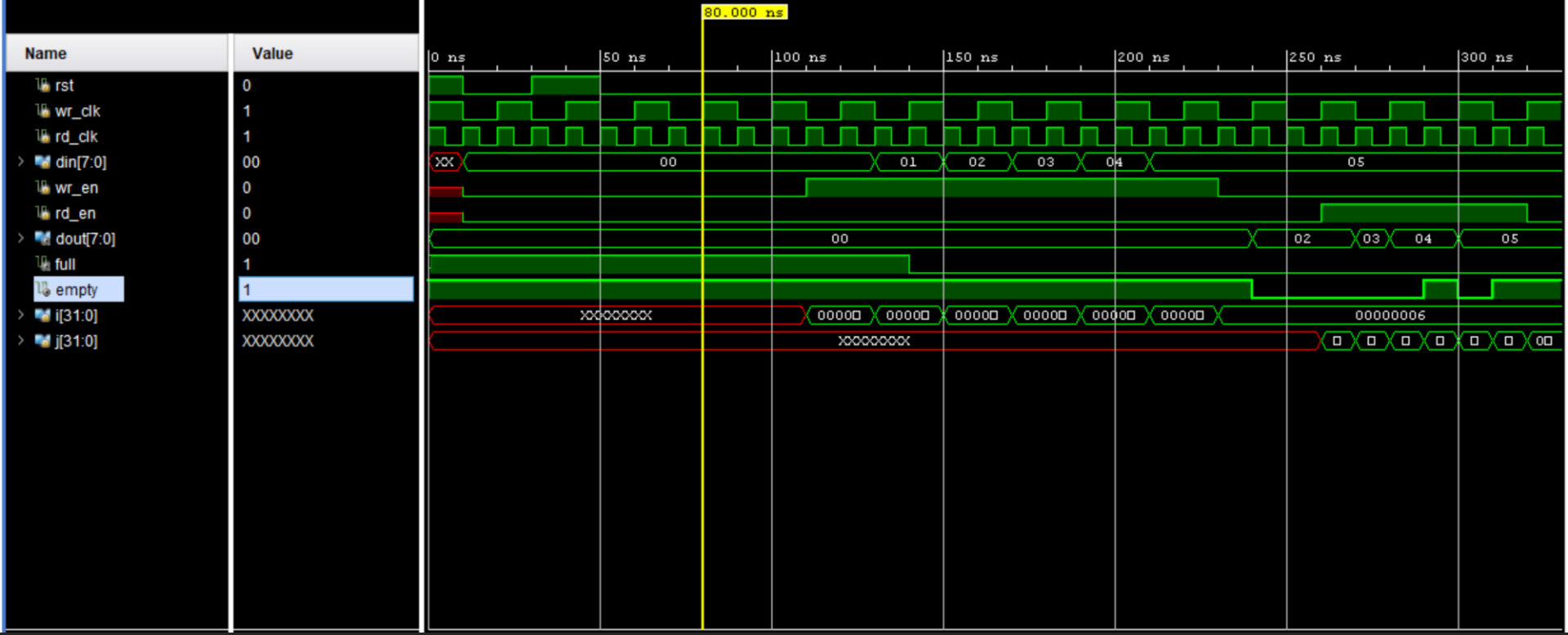
Process?

Wait?

Assert?

```
library ieee;
use ieee.std_logic_1164.all;
entity eq2_tb is
end eq2_tb;

architecture tb_arch of eq2_tb is
   signal test_in0 : std_logic_vector(1 downto 0);
   signal test_in1 : std_logic_vector(1 downto 0);
   signal test_out : std_logic;
begin
   -- instantiate the circuit under test
   uut: entity work.eq2(struc_arch)
      port map(
         a    => test_in0,
         b    => test_in1,
         aeqb => test_out
      );
   -- test vector generator
   process
   begin
      -- test vector 1
      test_in0 <= "00";
      test_in1 <= "00";
      wait for 200 ns;
      -- test vector 2
      test_in0 <= "01";
      test_in1 <= "00";
      wait for 200 ns;
      -- test vector 3
      test_in0 <= "01";
      test_in1 <= "11";
      wait for 200 ns;
      -- test vector 4
      test_in0 <= "10";
      test_in1 <= "10";
      wait for 200 ns;
      -- test vector 5
      test_in0 <= "10";
      test_in1 <= "00";
      wait for 200 ns;
      -- test vector 6
      test_in0 <= "11";
      test_in1 <= "11";
      wait for 200 ns;
      -- test vector 7
      test_in0 <= "11";
      test_in1 <= "01";
      wait for 200 ns;
      -- terminate simulation
      assert false
         report "Simulation Completed"
         severity failure;
   end process;
end tb_arch;
```



Roteiro prático para criar uma simulação simples

VIVADO 2017.4

Criando uma simulação...

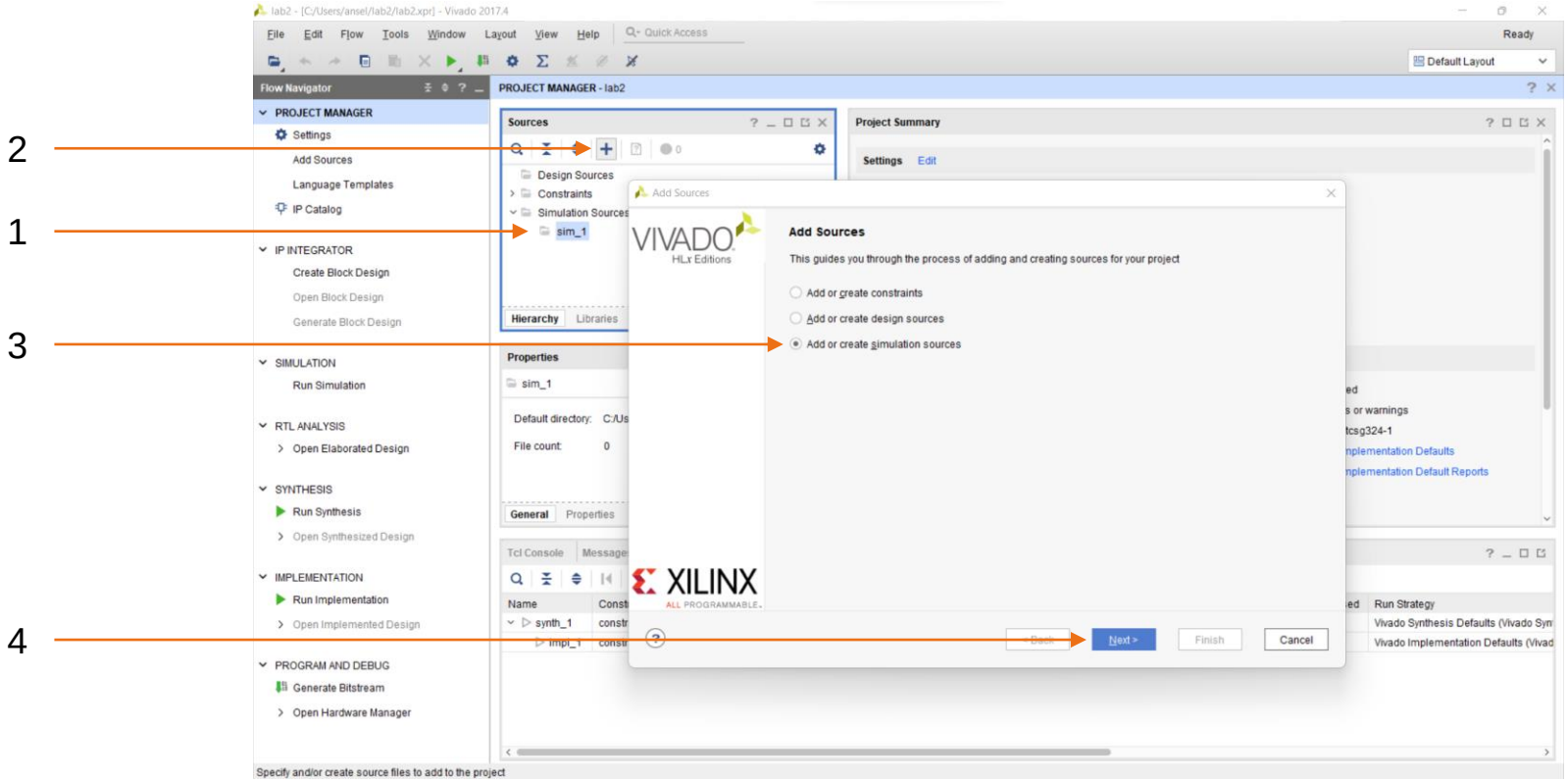
No Vivado Design Suite, o fluxo de simulação simplificado consiste nas seguintes etapas principais:

1. Crie um projeto de design
2. Adicione ou crie códigos de design e testbench
3. Faça a simulação inicial
4. Personalize a subjanela de forma de onda e simule novamente (opcional)

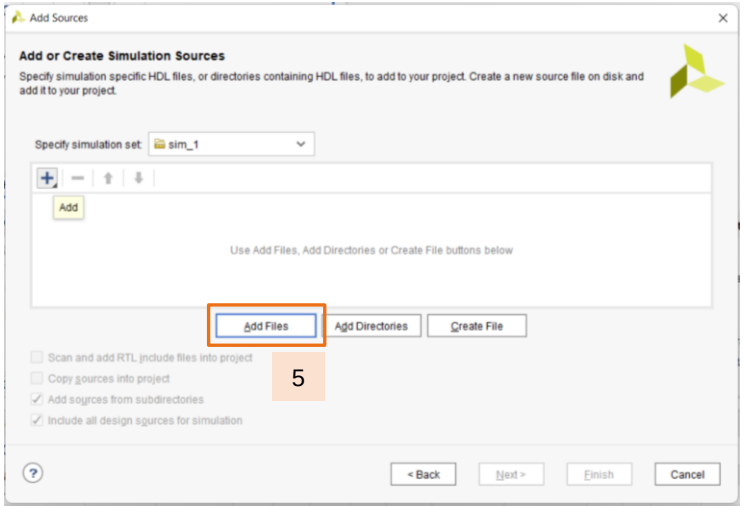
Passo1. Crie um projeto de design

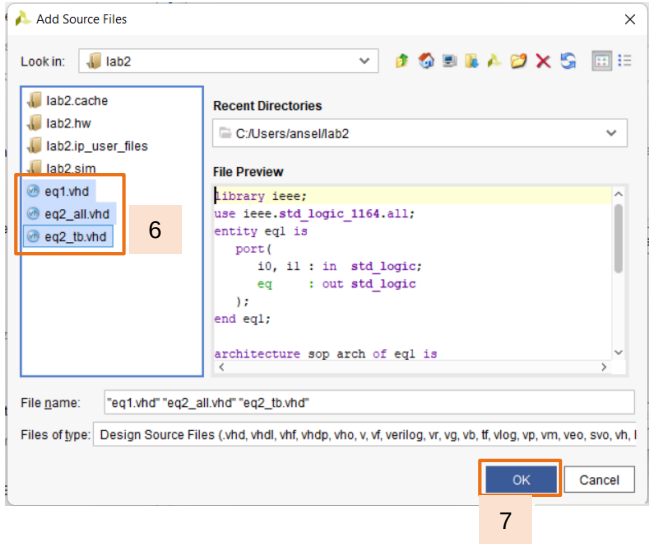
Fazer o mesmo processo da aula anterior ...

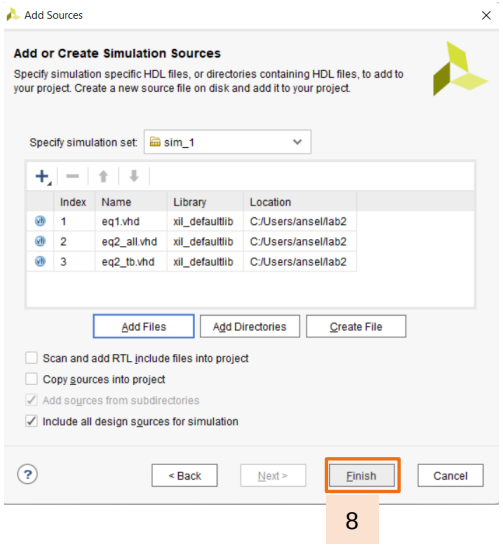
Passo 2. Adicione ou crie códigos de design e testbench



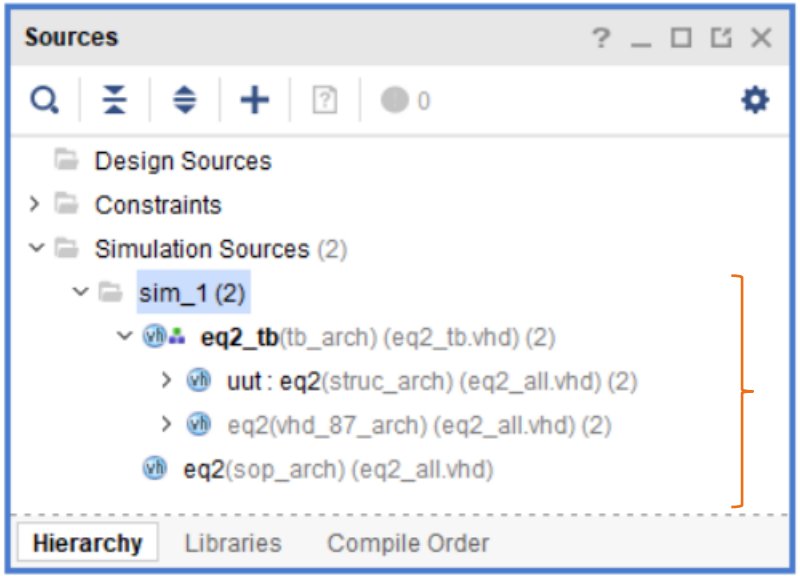
Passo 2. Adicione ou crie códigos de design e testbench





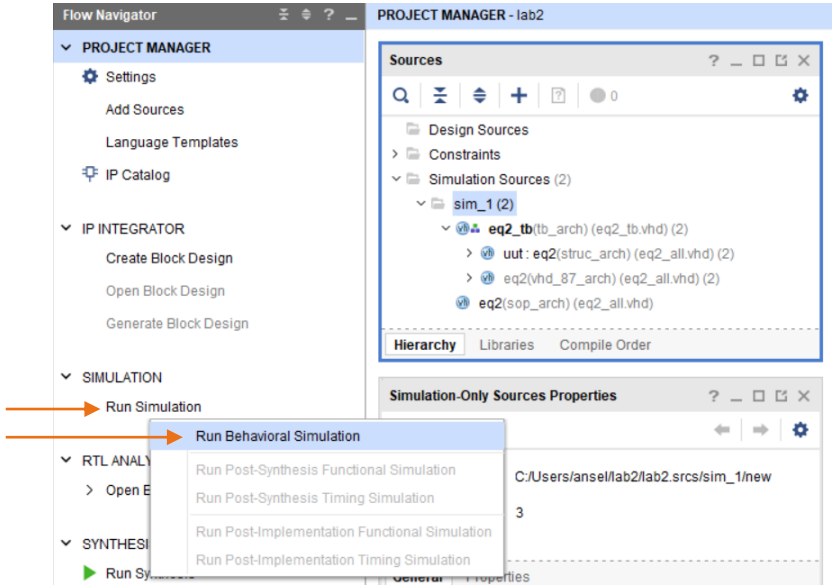


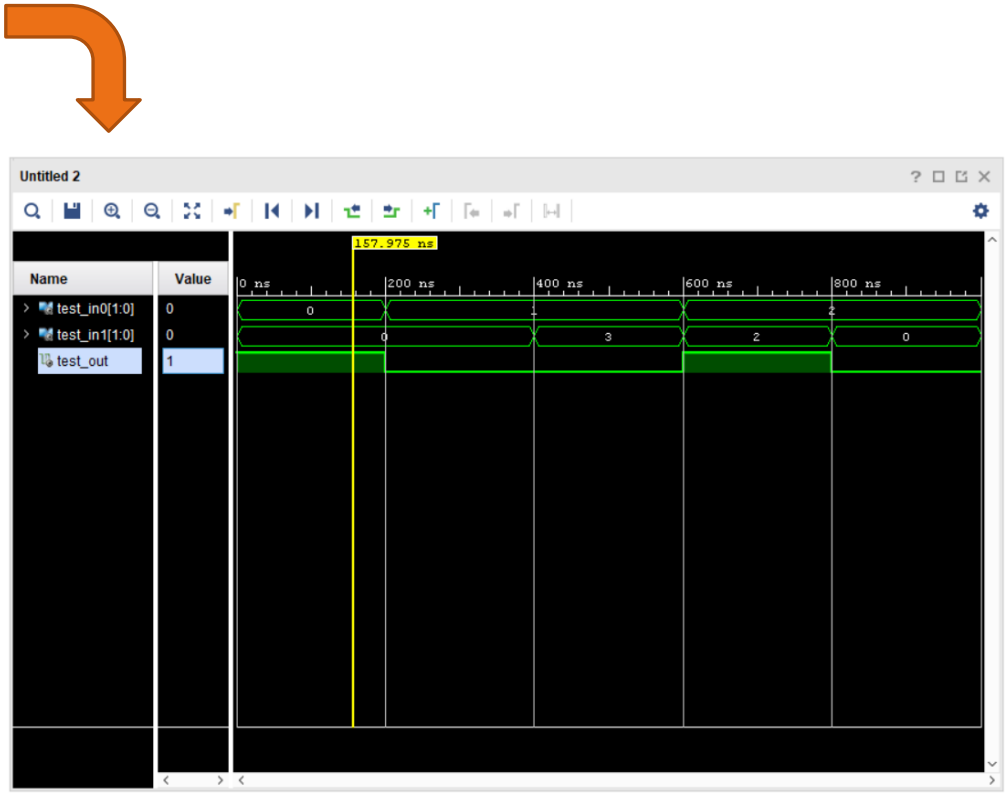
Passo 2. Adicione ou crie códigos de design e testbench



Resultado final

Passo 3. Faça a simulação inicial





Passo 4. Personalize a subjanela de forma de onda e simule novamente (opcional)

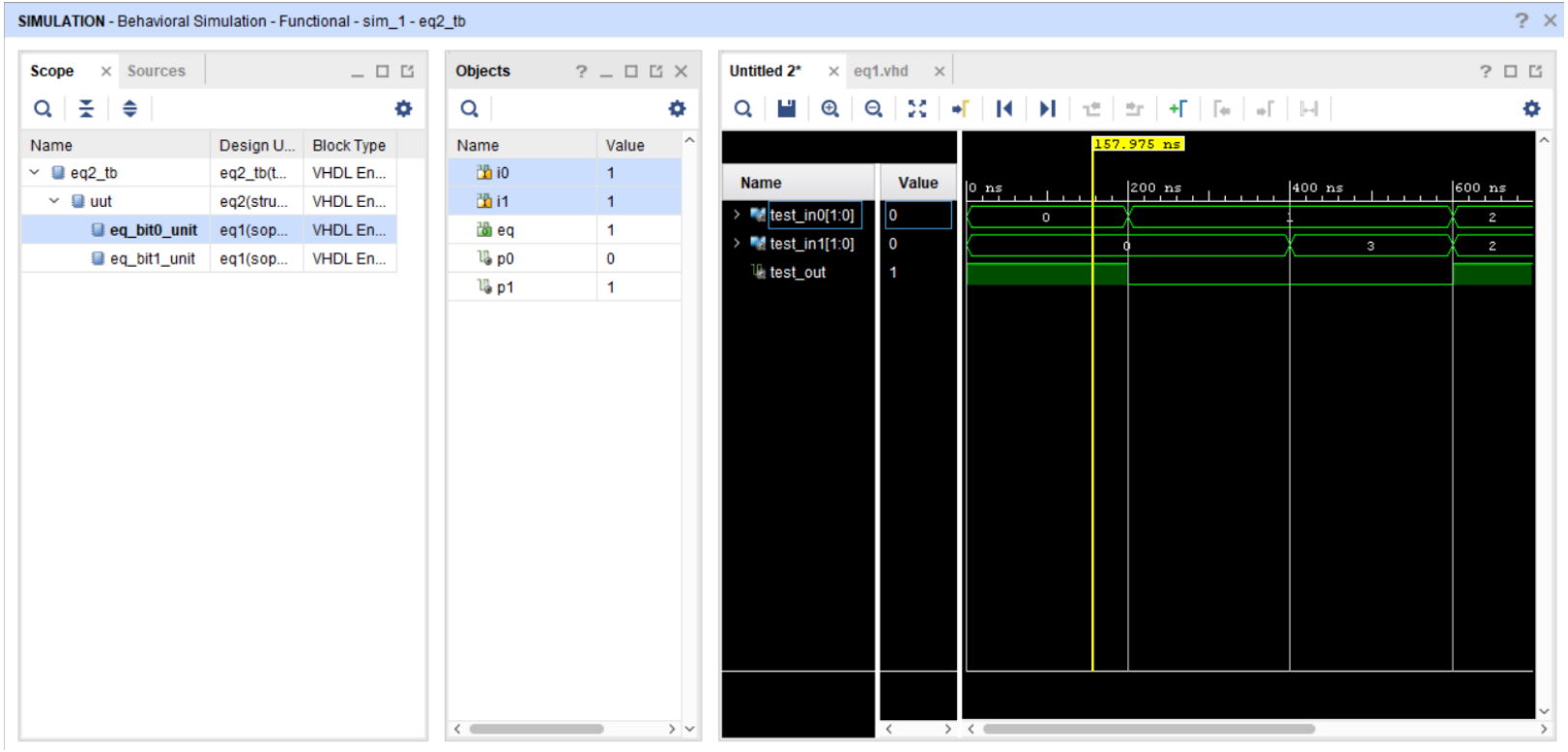
Frequentemente é necessário examinar sinais em módulos de nível inferior. Sinais adicionais podem ser adicionados e a subjanela Waveform pode ser personalizada de acordo.

Por exemplo, queremos verificar as operações dos dois comparadores de um bit adicionando seus sinais de E/S à subjanela. Isso pode ser feito da seguinte forma:

1. Na subjanela *Scopes*, expanda o ícone eq2_testbench e, em seguida, o ícone uut e, em seguida, realce a instância da unidade eq_bit0. A subjanela *Objects* é atualizada de acordo e lista as portas de E/S e os sinais internos da instância da unidade eq_bit0.

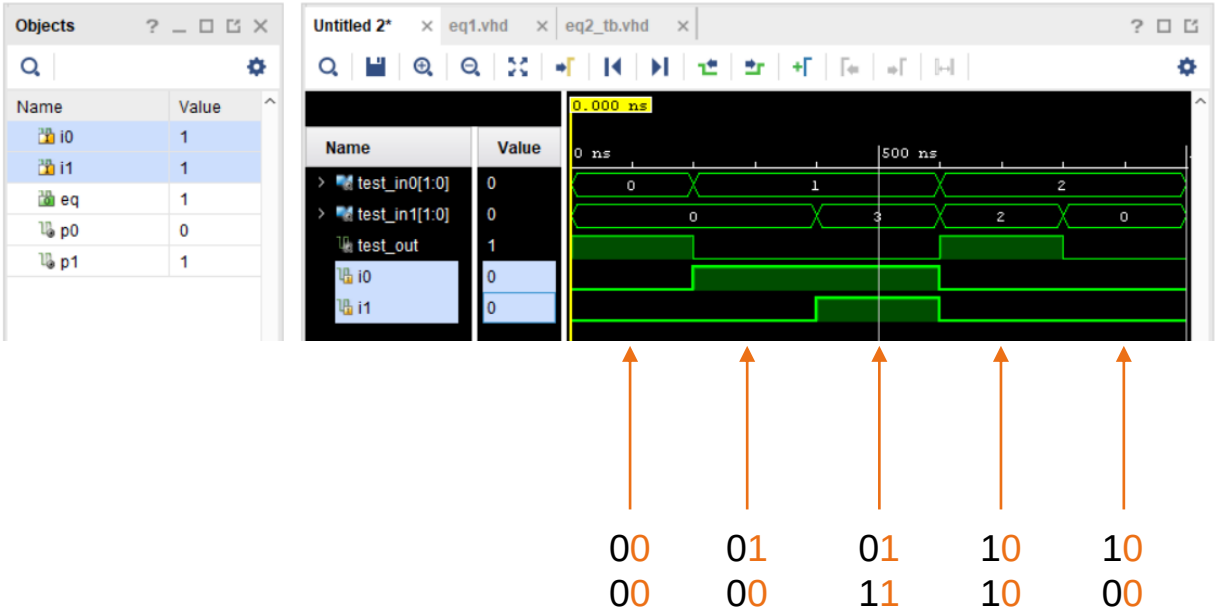
2. Na subjanela *Objects*, selecione i0, i1 e eq e depois arraste e solte-os na subjanela *Waveform*.

Passo 4. Personalize a subjanela de forma de onda e simule novamente (opcional)



Passo 4. Personalize a subjanela de forma de onda e simule novamente (opcional)

Resultado:

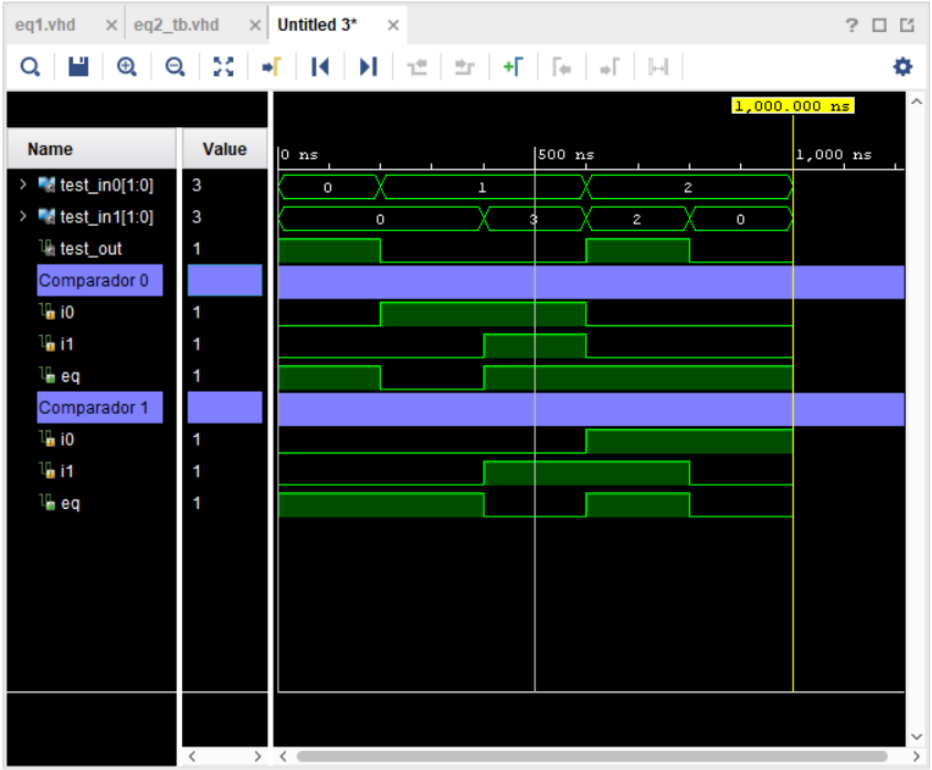


Passo 4. Personalize a subjanela de forma de onda e simule novamente (opcional)

Podemos refazer esse procedimento para observar entradas e saídas de um dos comparadores, por exemplo.

Outras personalizações podem ser executadas → explorar o ambiente para conhecer mais opções

- Dica: adicionar um divisor para organizar a simulação



Relaunch Simulation...