



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA
UNIVERSIDADE NOVA DE LISBOA



Xilinx_ISE_DS_Win_14.7

SIMULAÇÃO COM FICHEIRO VHDL

Simulação com um ficheiro VHDL

O esquemático criado poderá ser testado recorrendo a um ficheiro VHDL. Para isso cria-se um ficheiro VHDL (tal como se criou o ficheiro esquemático). O procedimento está ilustrado nas seguintes figuras. Na Figura 1 define-se o tipo de ficheiro a criar:

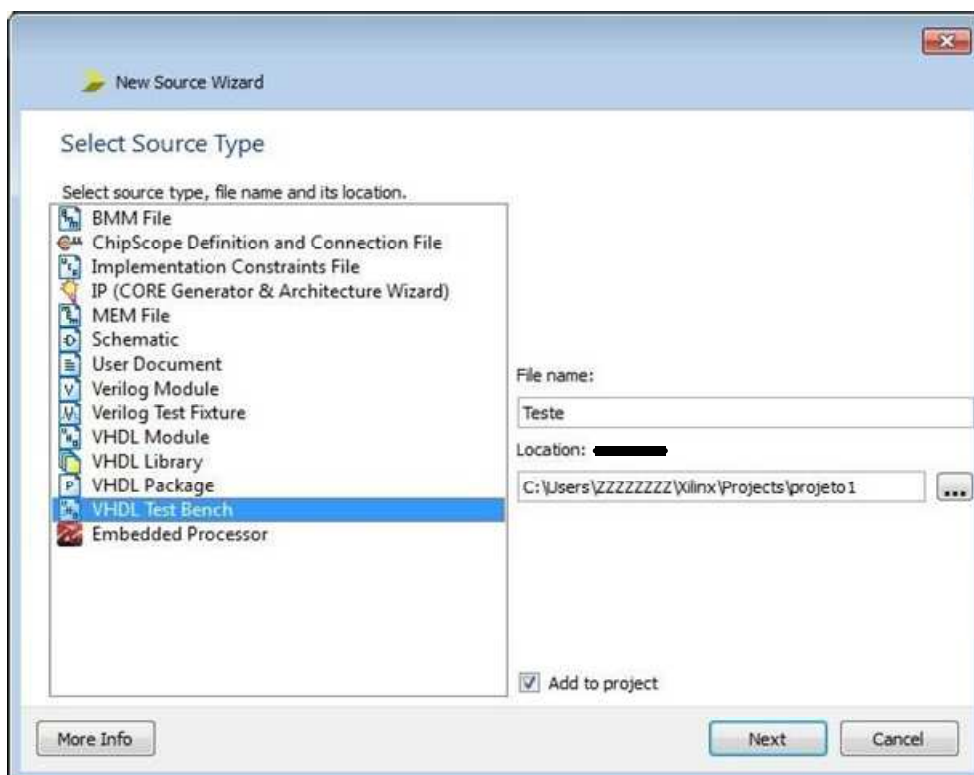


Figura 1-Criar ficheiro VHDL Test Bench

Em seguida escolhemos a qual dos esquemas do projeto queremos relacionar este ficheiro VHDL, podendo assim testar várias partes do projeto separadamente. Neste caso apenas temos um esquemático a testar, Figura 2.

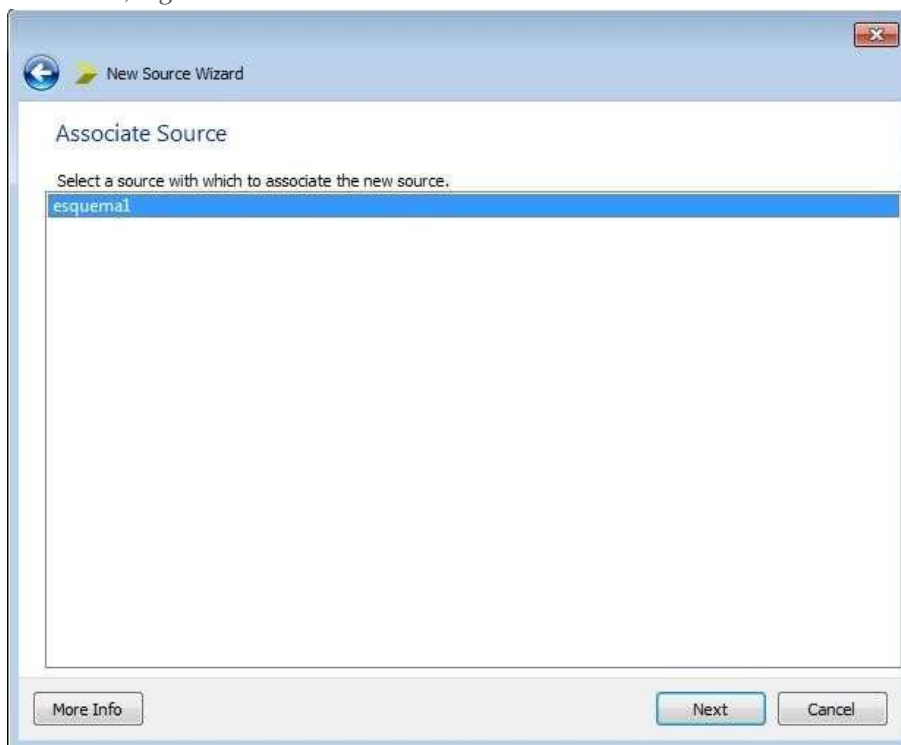


Figura 2-Associação ao esquemático

Confirmam-se as opções escolhidas, Figura 3

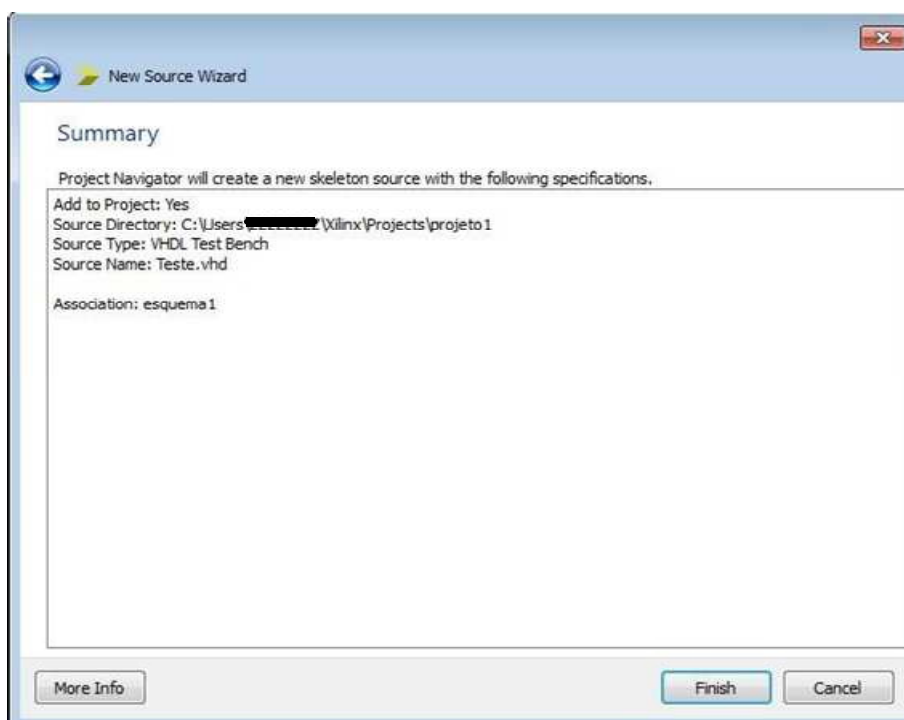


Figura 3-Confirmação das configurações do ficheiro VHDL

Abrirá uma nova divisão, no xilinx, correspondente ao ficheiro VHDL, como exemplificado na Figura 4.

```

1  -- Vhdl test bench created from schematic C:\Users\...Xilinx\Projects\projeto1\esquema1.sch - Wed Jul 31 16:05:53 20
2  --
3  -- Notes:
4  -- 1) This testbench template has been automatically generated using types
5  -- std_logic and std_logic_vector for the ports of the unit under test.
6  -- Xilinx recommends that these types always be used for the top-level
7  -- I/O of a design in order to guarantee that the testbench will bind
8  -- correctly to the timing (post-route) simulation model.
9  -- 2) To use this template as your testbench, change the filename to any
10 -- name of your choice with the extension .vhd, and use the "Source->Add"
11 -- menu in Project Navigator to import the testbench. Then
12 -- edit the user defined section below, adding code to generate the
13 -- stimulus for your design.
14 --
15 LIBRARY ieee;
16 USE ieee.std_logic_1164.ALL;
17 USE ieee.numeric_std.ALL;
18 LIBRARY UNISIM;
19 USE UNISIM.Vcomponents.ALL;
20 ENTITY esquema1_esquema1_sch_tb IS
21 END esquema1_esquema1_sch_tb;
22 ARCHITECTURE behavioral OF esquema1_esquema1_sch_tb IS
23
24     COMPONENT esquema1
25     PORT( A : IN STD_LOGIC;
26           B : IN STD_LOGIC;
27           C : IN STD_LOGIC;
28           Z : OUT STD_LOGIC);
29     END COMPONENT;
30
31     SIGNAL A : STD_LOGIC;

```

Figura 4-Ficheiro .vhd

Neste ficheiro será descrito a evolução temporal das entradas. Para exemplificar consideremos o esquemático apresentado no tutorial “Criar um projeto”.

Na Figura 5 demonstra-se, destacado a verde, um exemplo de evolução temporal em linguagem VHDL. A vermelho destacam-se os separadores selecionados para poder ser feita a simulação: deverá ser selecionado o ficheiro esquemático e a opção de Simulation.

Antes da simulação deverá ser feita uma confirmação ao código desenvolvido fazendo duplo click em *Behavioral Check Model*, na janela destacada a preto.

Se não existirem erros (que aparecerão na janela destacada a azul), o esquemático está pronto a ser testado.

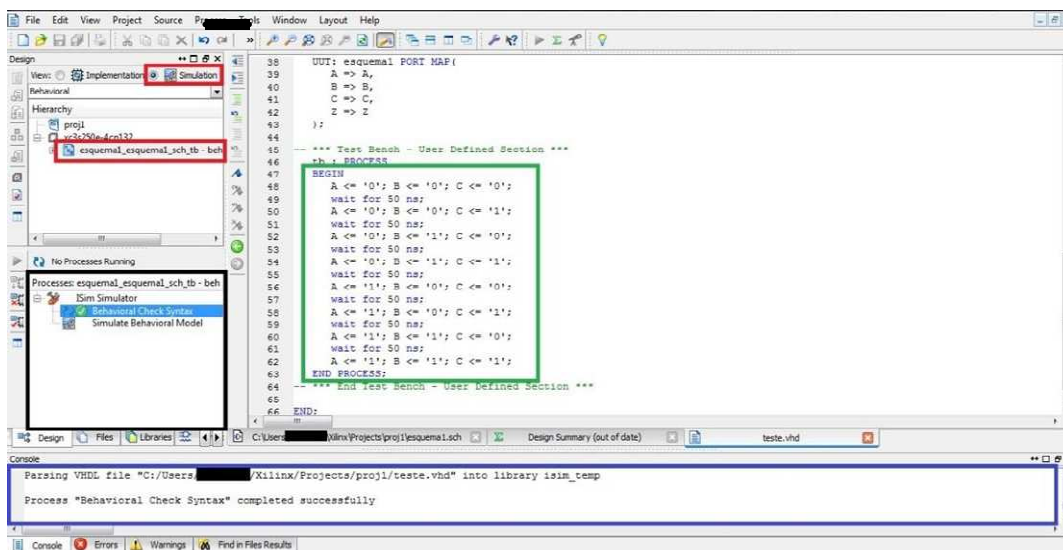


Figura 5: Preparação da simulação

Duplo click em Simulate Behavioral Model e aparecerá uma janela com a simulação, Figura 6:

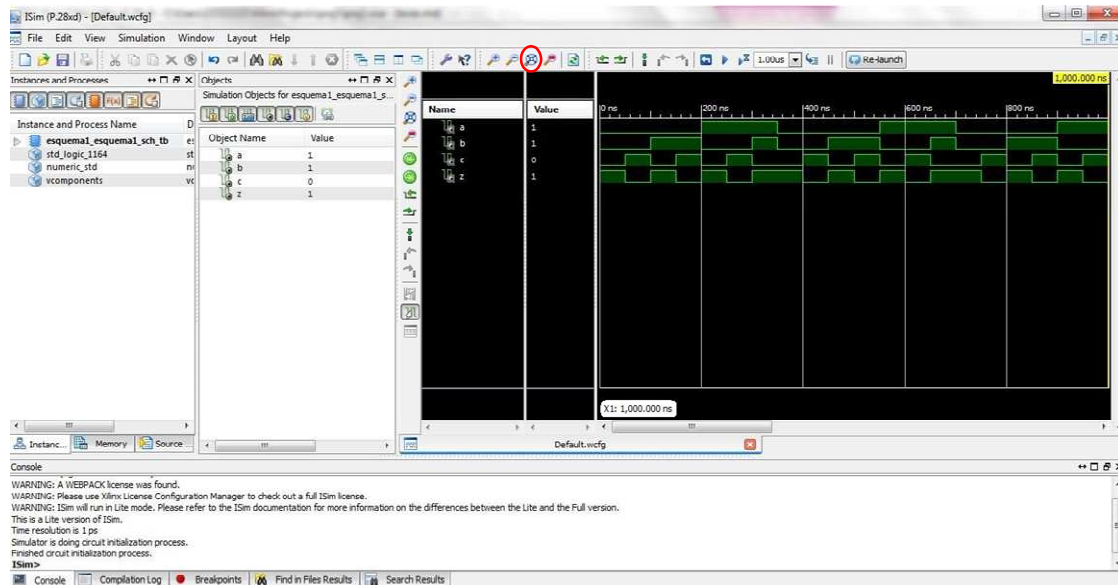


Figura 6-Simulação temporal

[Nota]: caso não se consiga visualizar o gráfico devido á escala de tempo, clicar com o botão do lado direito em cima do gráfico e escolher a opção To Full View, realçado a vermelho na figura acima, a escala de tempo será ajustada automaticamente.

Outro exemplo de código de teste, neste caso para um circuito com um flip-flop, onde seria preciso implementar vetores para as variáveis de entrada e um ciclo para variar o clock periodicamente.

```

43 );
44
45 -- *** Test Bench - User Defined Section ***
46
47 -----
48
49 --criar vectores para a entrada
50 A <= '0','1' after 150 ns,'0' after 350 ns,'1' after 450 ns;
51
52
53 --ciclo que cria o clock, sinal este que varia de 100 em 100 ns, neste caso.
54 tb : PROCESS
55 BEGIN
56   CLK <= '0';
57   WAIT for 100 ns; --   esperar 100 ns
58   CLK <= '1';
59   WAIT for 100 ns; --   esperar 100 ns
60 END PROCESS;
61
62 -----
63 -- *** End Test Bench - User Defined Section ***
64
65 END;
66

```

Figura 7-Peparação da Simulação

A simulação resultante, Figura 8.



Figura 8-Simulação temporal

Quando os sistemas são mais complexos existem saídas de um componente que se ligam a entradas de outro. Estas não aparecem na simulação pois não são saídas do sistema mas sim sinais internos. Muitas vezes é necessário ter também acesso a essas entradas/saídas.

Segue-se um exemplo na Figura 9 de um esquemático com sinais internos:

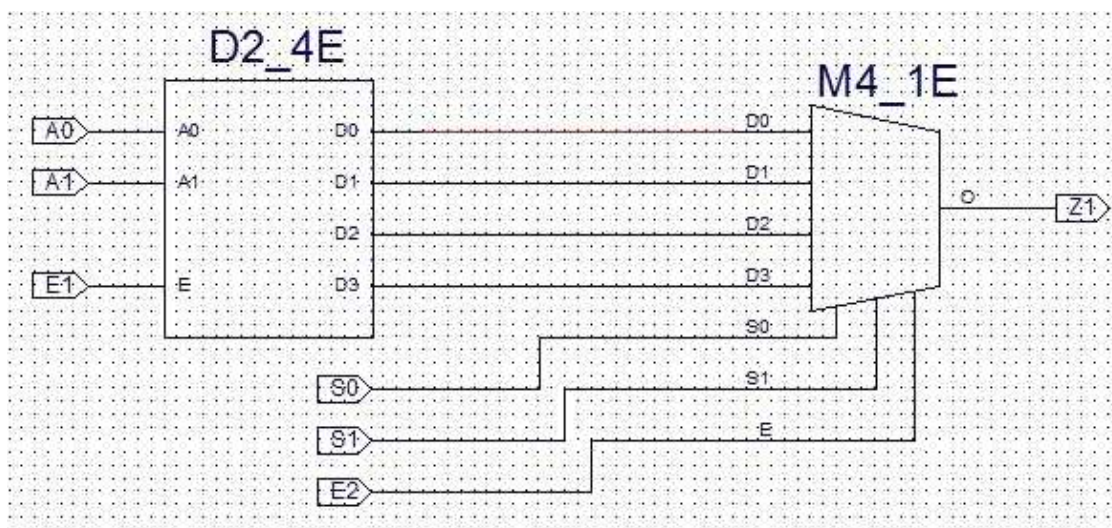


Figura 9-Esquemático com sinais internos

Os sinais Do, D1, D2 e D3 são sinais internos que não aparecem automaticamente na simulação.

É possível alterar os nomes desses sinais para nomes mais convenientes fazendo duplo clique em cima do fio correspondente ao sinal, como demonstra a Figura 10.

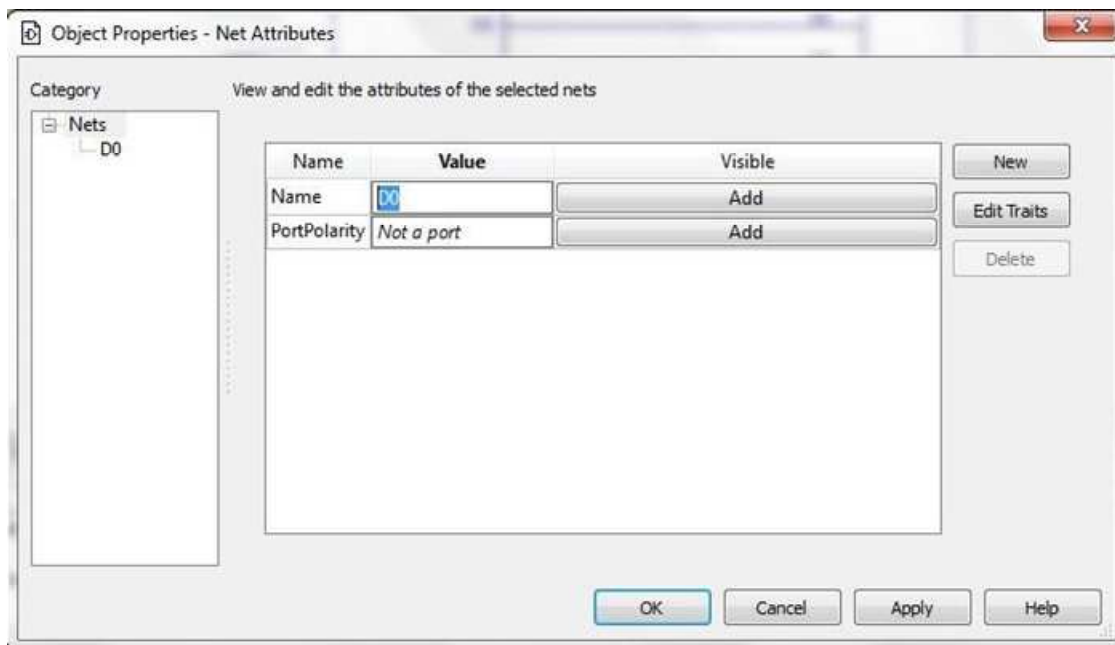


Figura 10-Alterar nome de um sinal interno

Fazer a simulação como demonstrado anteriormente.

Para ter acesso aos sinais internos selecionar a opção UUT da janela esquerda da simulação. A opção é demonstrada na Figura 11:

[Nota]: fazer clique único na opção UUT

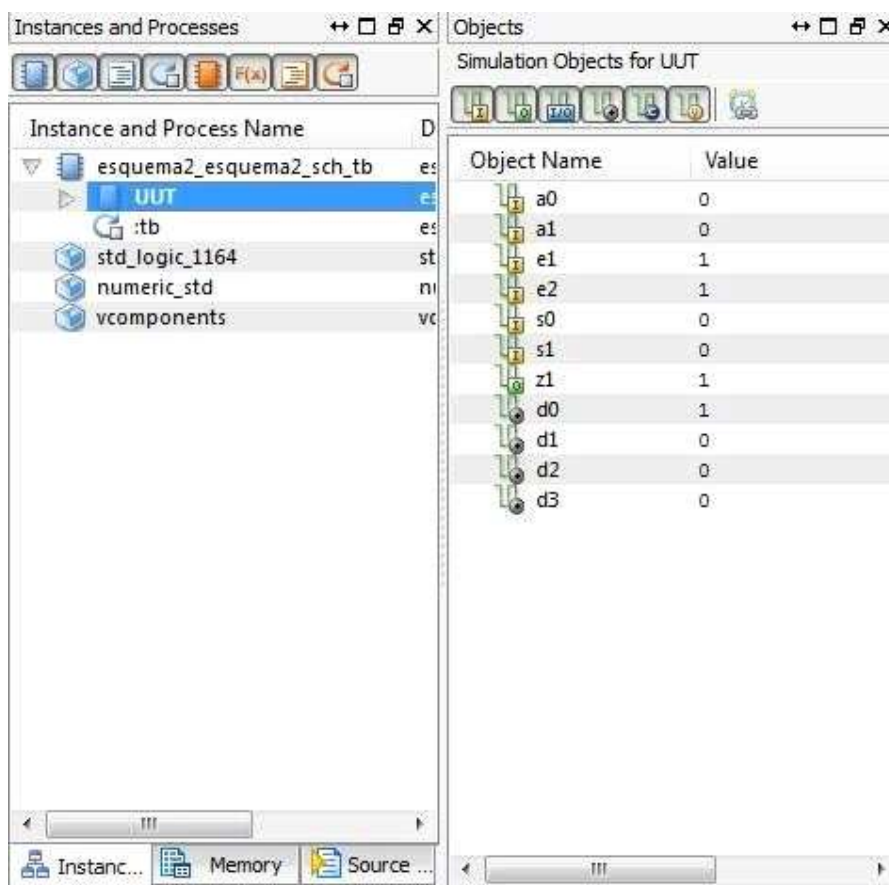


Figura 11- Seleção da opção UUT

Como se pode observar na Figura 9 os sinais internos estão agora disponíveis na janela *Simulation Objects for UUT*, do lado direito.

Falta agora poder observar a evolução temporal dos sinais internos. Para isso arrasta-se o sinal para a zona do gráfico como demonstra a Figura 12:

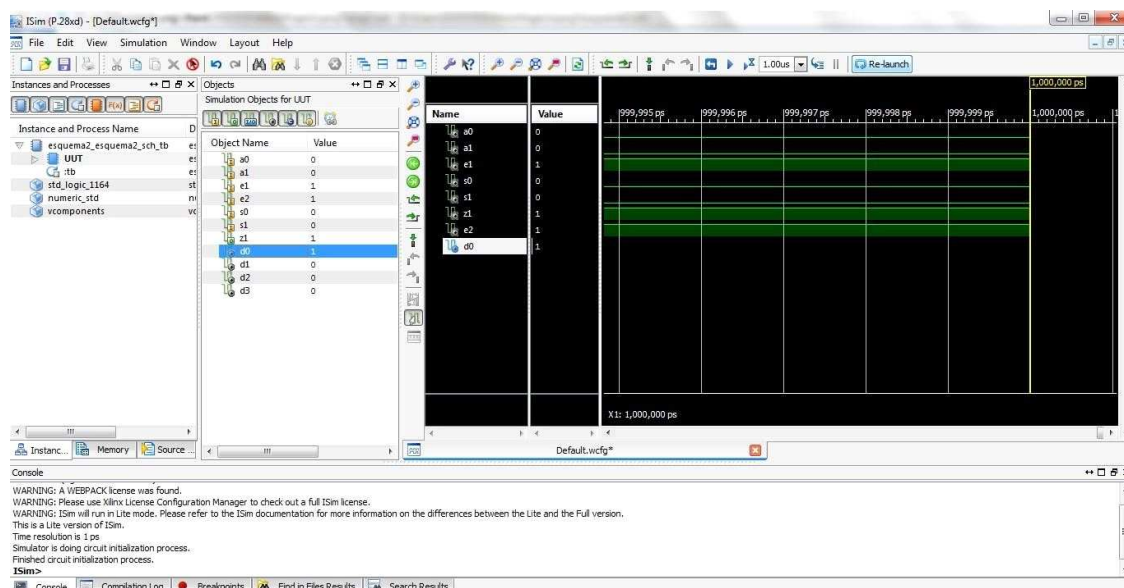


Figura 12- Sinal interno

Apesar do sinal agora estar na janela junto aos sinais que aparecem na simulação, este ainda não aparece. Para aparecer é necessário fazer *Re-launch*. Na figura 12 está destacado o botão da opção *Re-launch*.

Temos agora acesso a evolução temporal do sinal interno como demonstra a Figura 13:

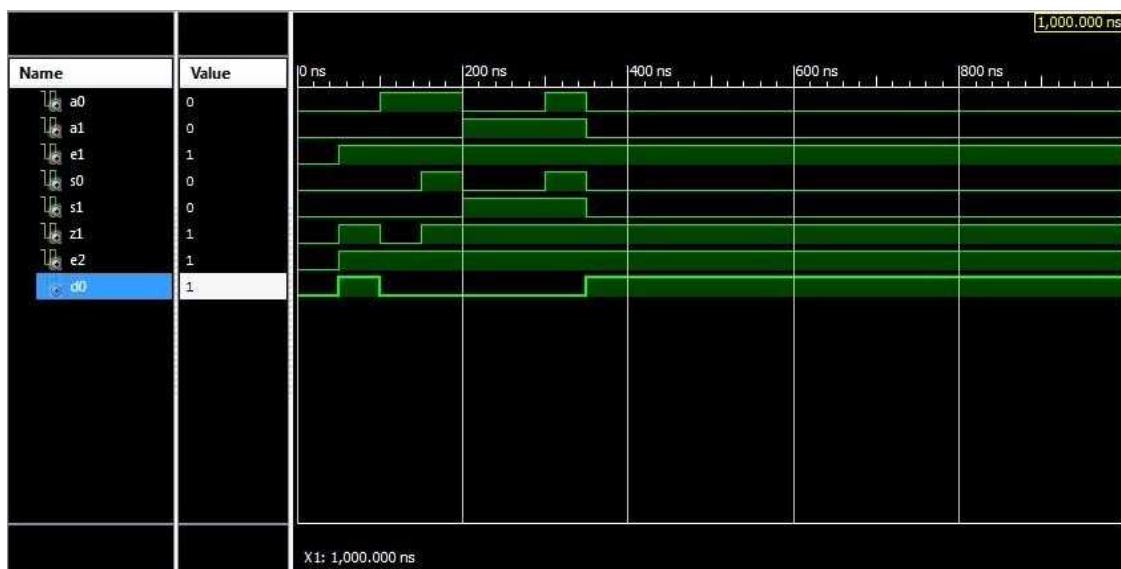


Figura 13- Evolução temporal do sinal interno