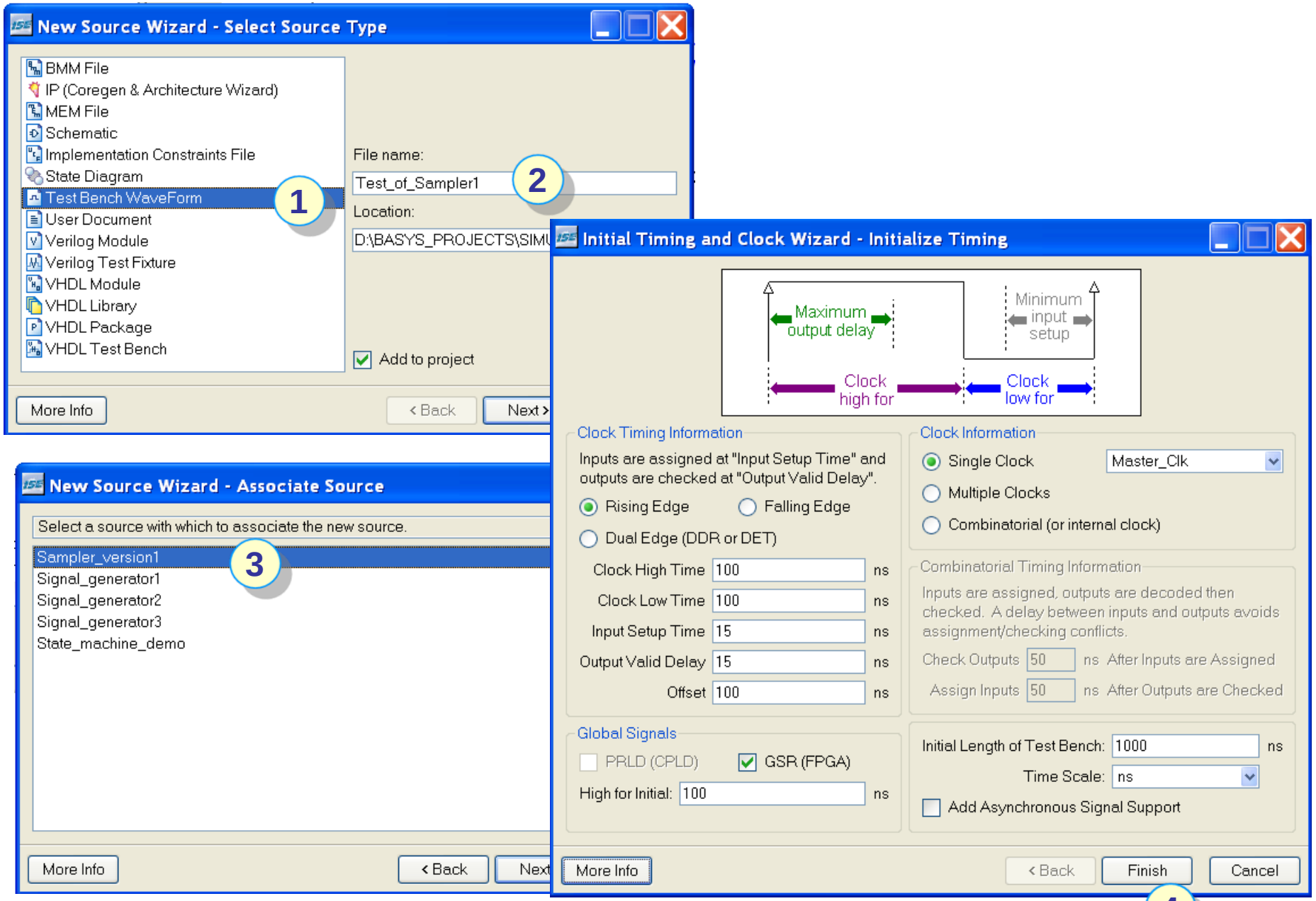
VHDL for simulation – Code for synthesize

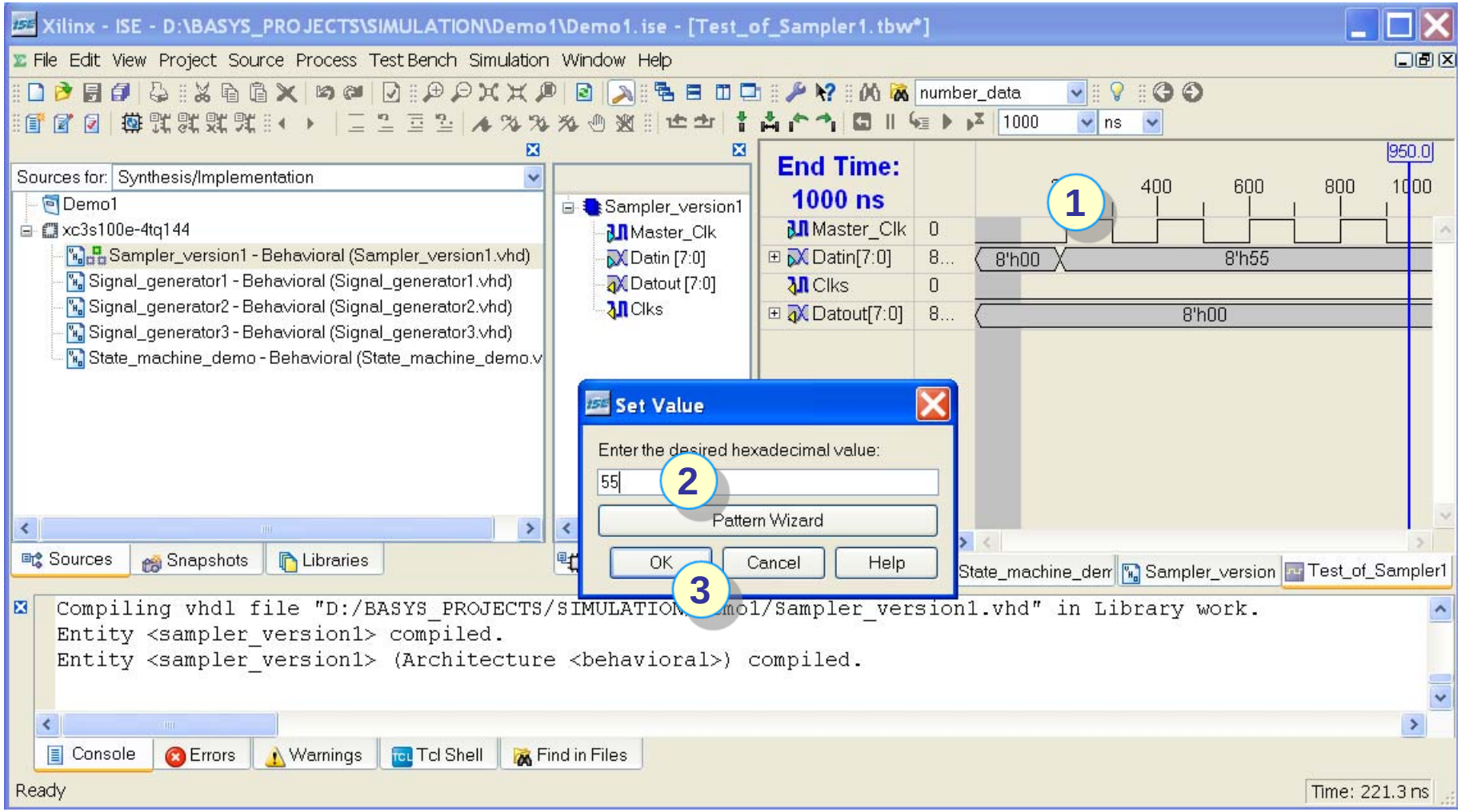
```
8  entity Sampler_version1 is
9      Port ( Master_Clk : in  STD_LOGIC;
10            Datin :      in  STD_LOGIC_VECTOR (7 downto 0);
11            Datout :     out STD_LOGIC_VECTOR (7 downto 0);
12            Clks :       out STD_LOGIC);
13 end Sampler_version1;
14
15 architecture Behavioral of Sampler_version1 is
16    Signal   Q: STD_LOGIC_VECTOR (7 downto 0);
17    Signal Enb: STD_LOGIC;
18 begin
19
20    Clks   <= not Enb;
21    Datout <= not Q;    -- Alternative Q+1
22
23    process( Master_Clk)
24       variable Scale: integer range 0 to 3 := 1;
25    begin
26       if rising_edge( Master_Clk) then
27          if Scale > 0 then
28             Scale := Scale-1;
29             Enb   <= '0';
30          else
31             Scale := 2;
32             Enb   <= '1';
33          end if;
34       end if;
35    end process;
36
37    process( Master_Clk)
38    begin
39       if rising_edge( Master_Clk) then
40          if Enb='1' then
41             Q <= Datin;
42          end if;
43       end if;
44    end process;
45
46 end Behavioral;
```


```
4  -- Notes:
5  -- 1) This instantiation template has been automatically generated using types
6  -- std_logic and std_logic_vector for the ports of the instantiated module
7  -- 2) To use this template to instantiate this entity, cut-and-paste and then edit
8
9     COMPONENT Sampler_version1
10    PORT(
11       Master_Clk : IN std_logic;
12       Datin : IN std_logic_vector(7 downto 0);
13       Datout : OUT std_logic_vector(7 downto 0);
14       Clks : OUT std_logic
15       );
16    END COMPONENT;
17
18    Inst_Sampler_version1: Sampler_version1 PORT MAP(
19       Master_Clk => ,
20       Datin => ,
21       Datout => ,
22       Clks =>
23    );
```

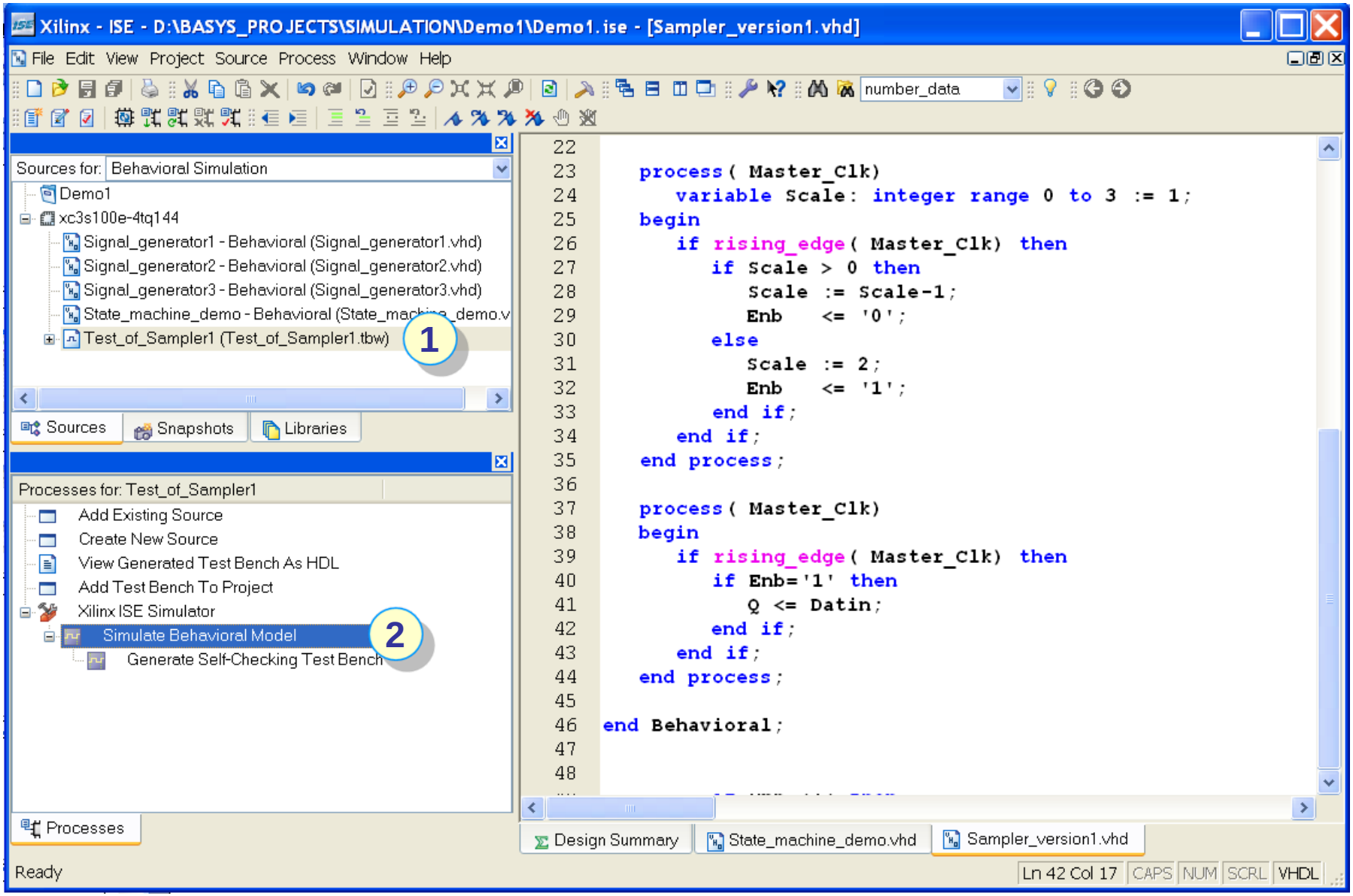
The TestBenchWaveForm – Tool (1)



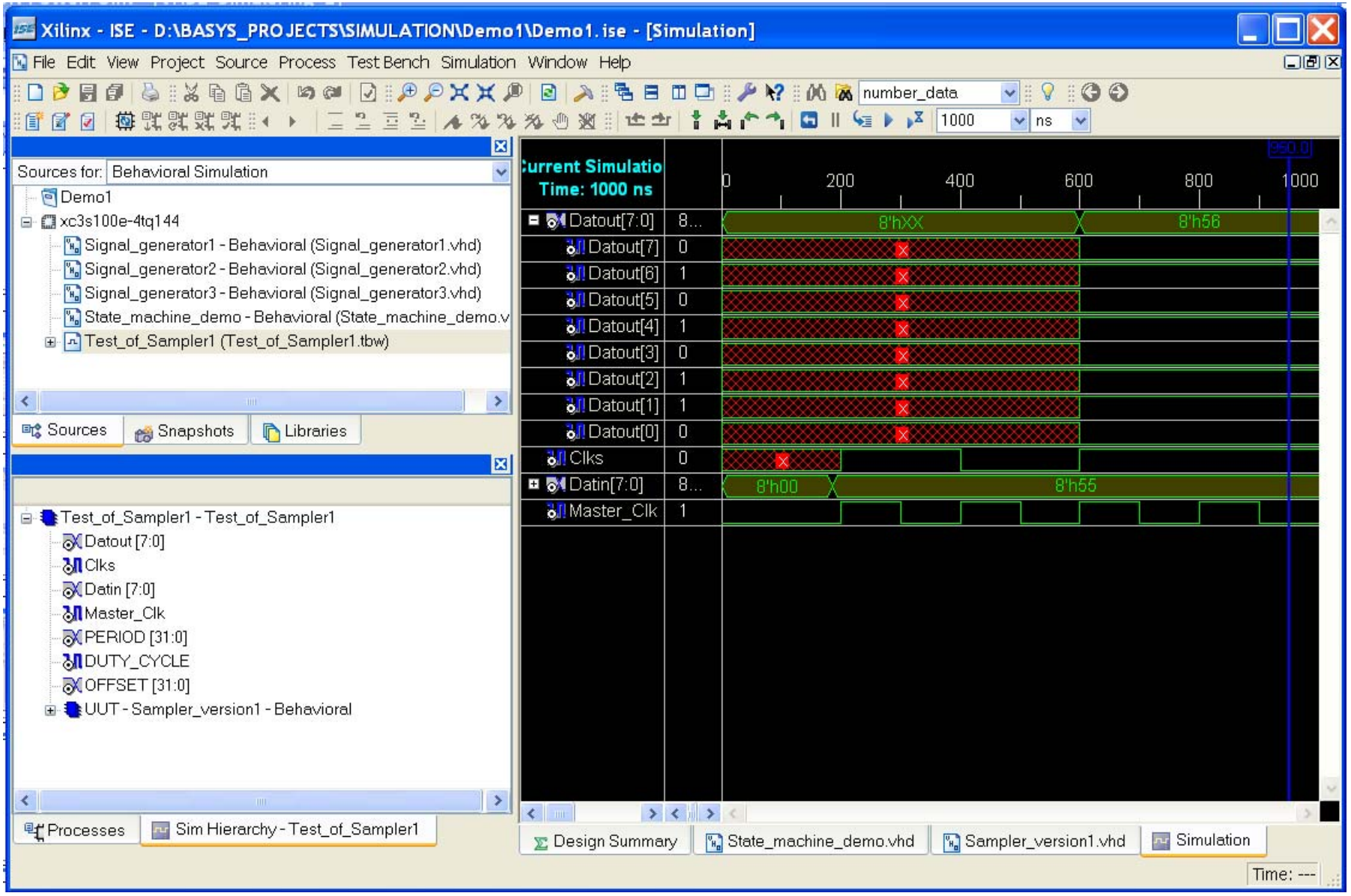
The TestBenchWaveForm – Tool (2)



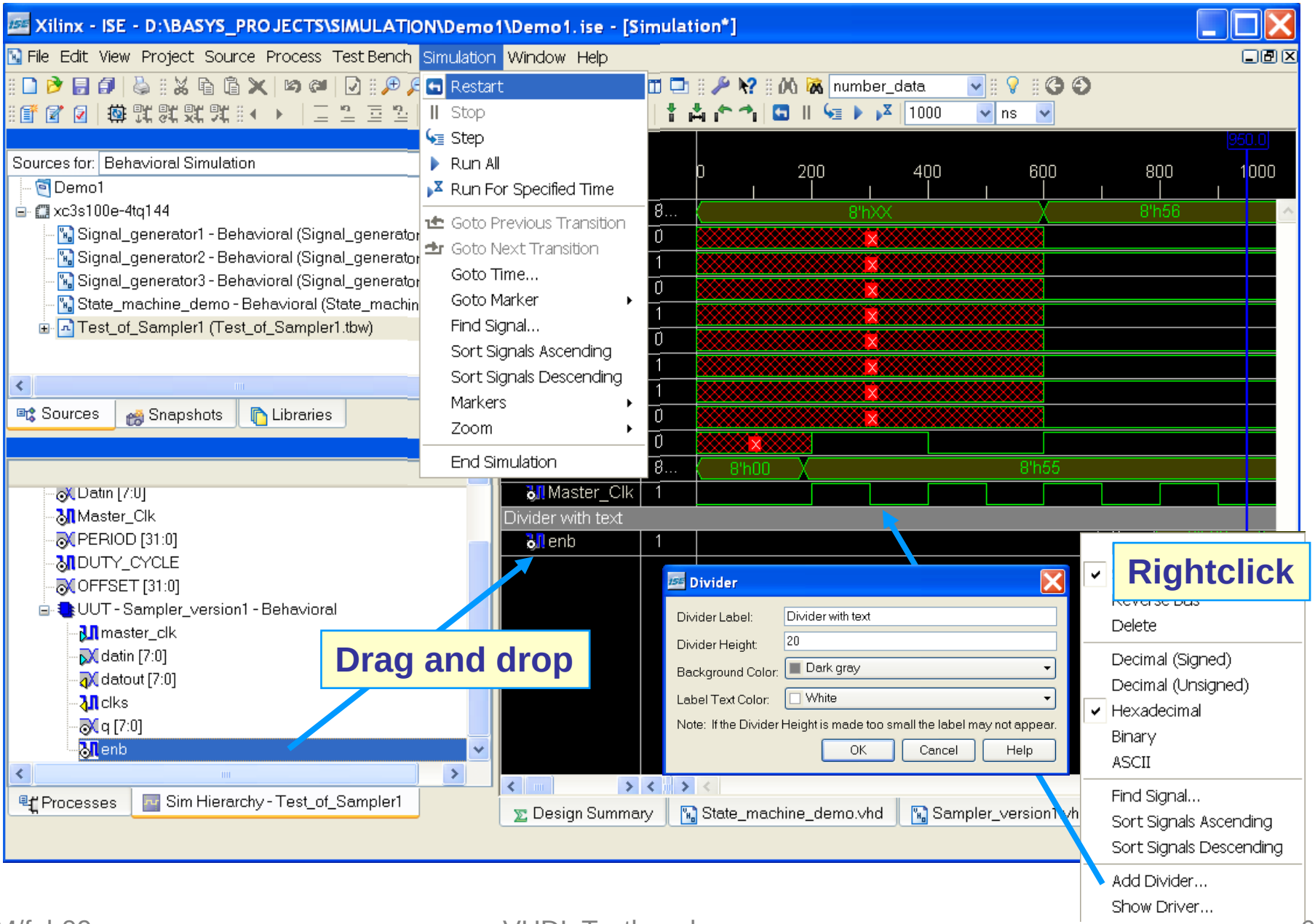
Behavioral Simulation (1)



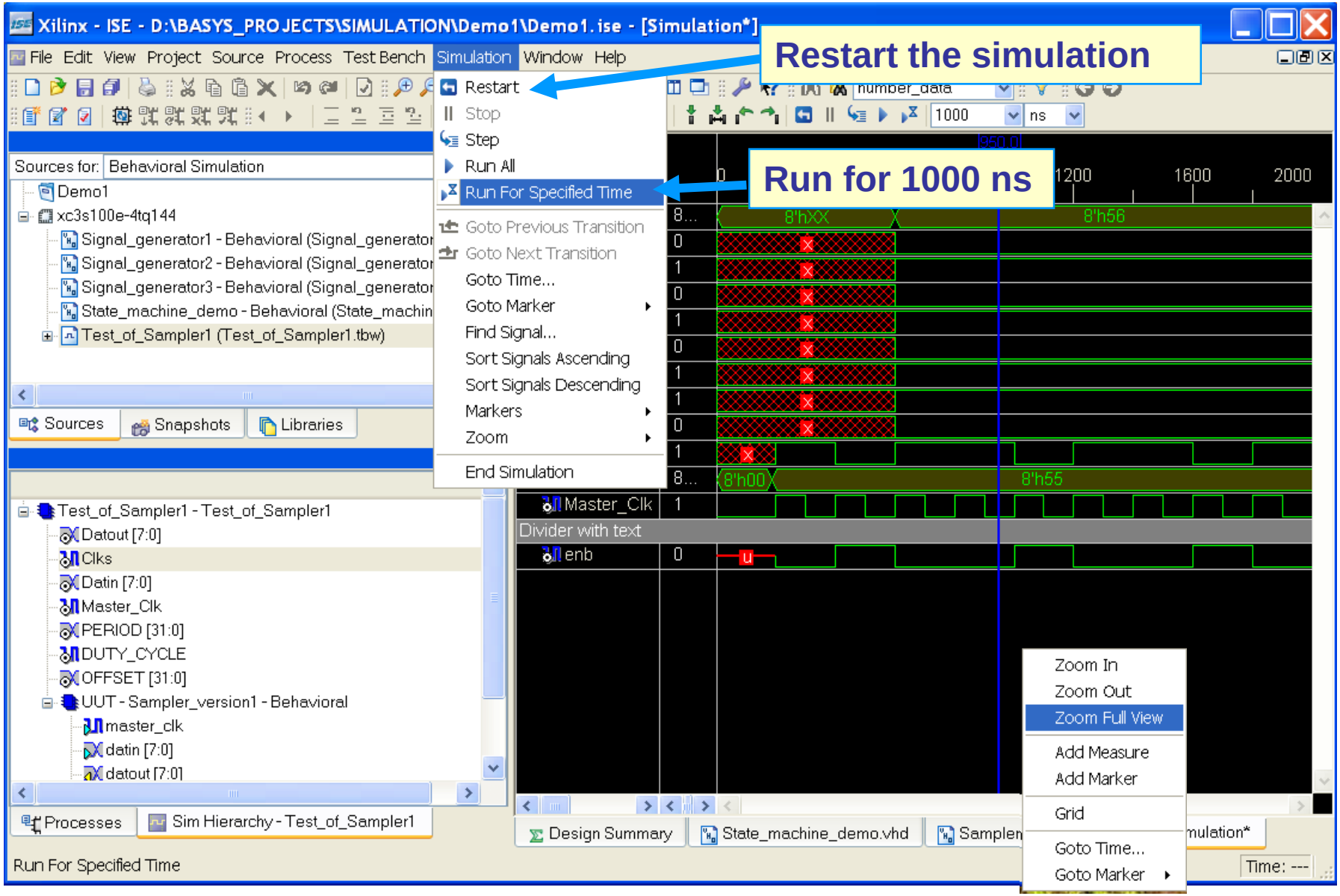
Behavioral Simulation (2)



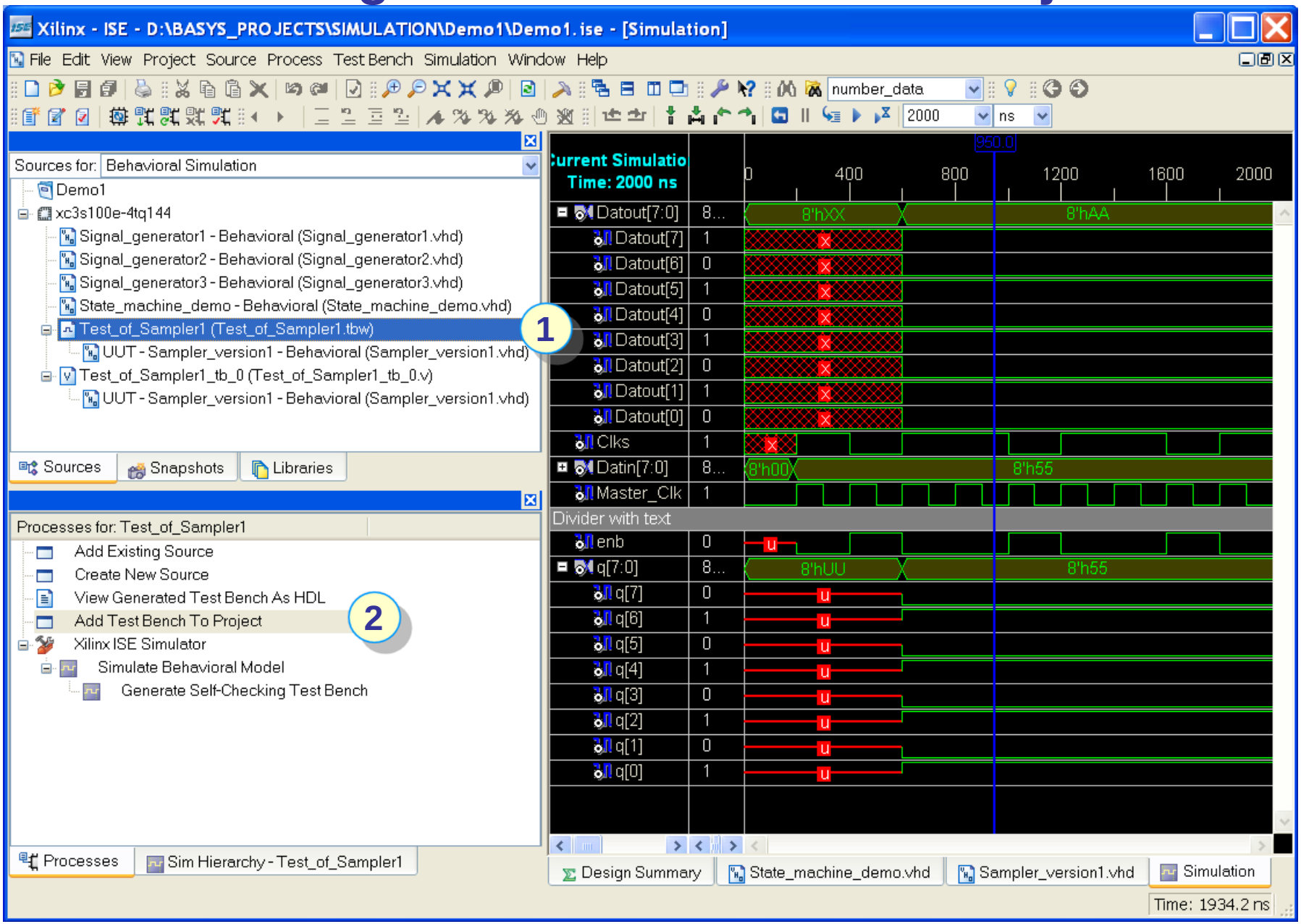
Behavioral Simulation (3)



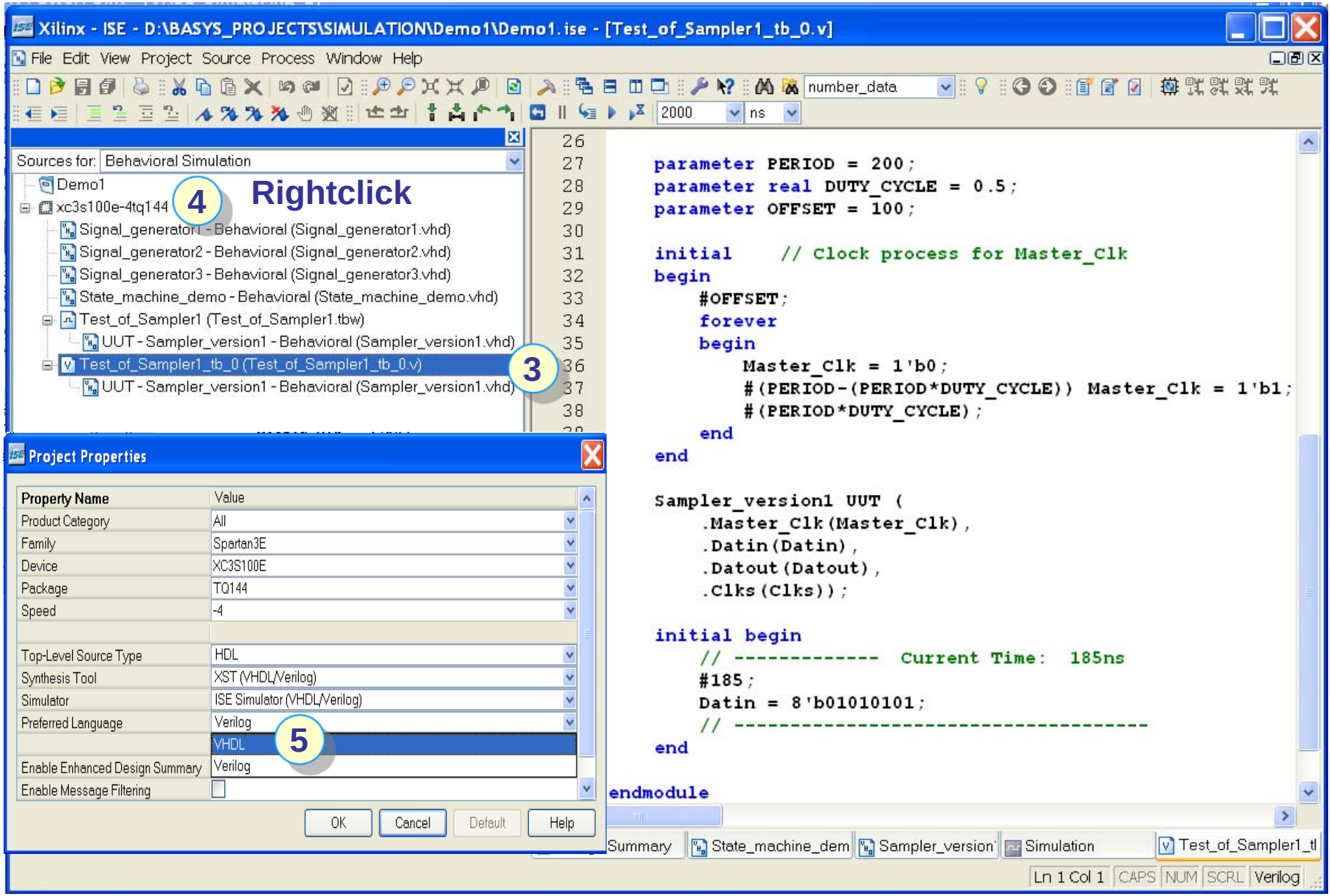
Behavioral Simulation (4)

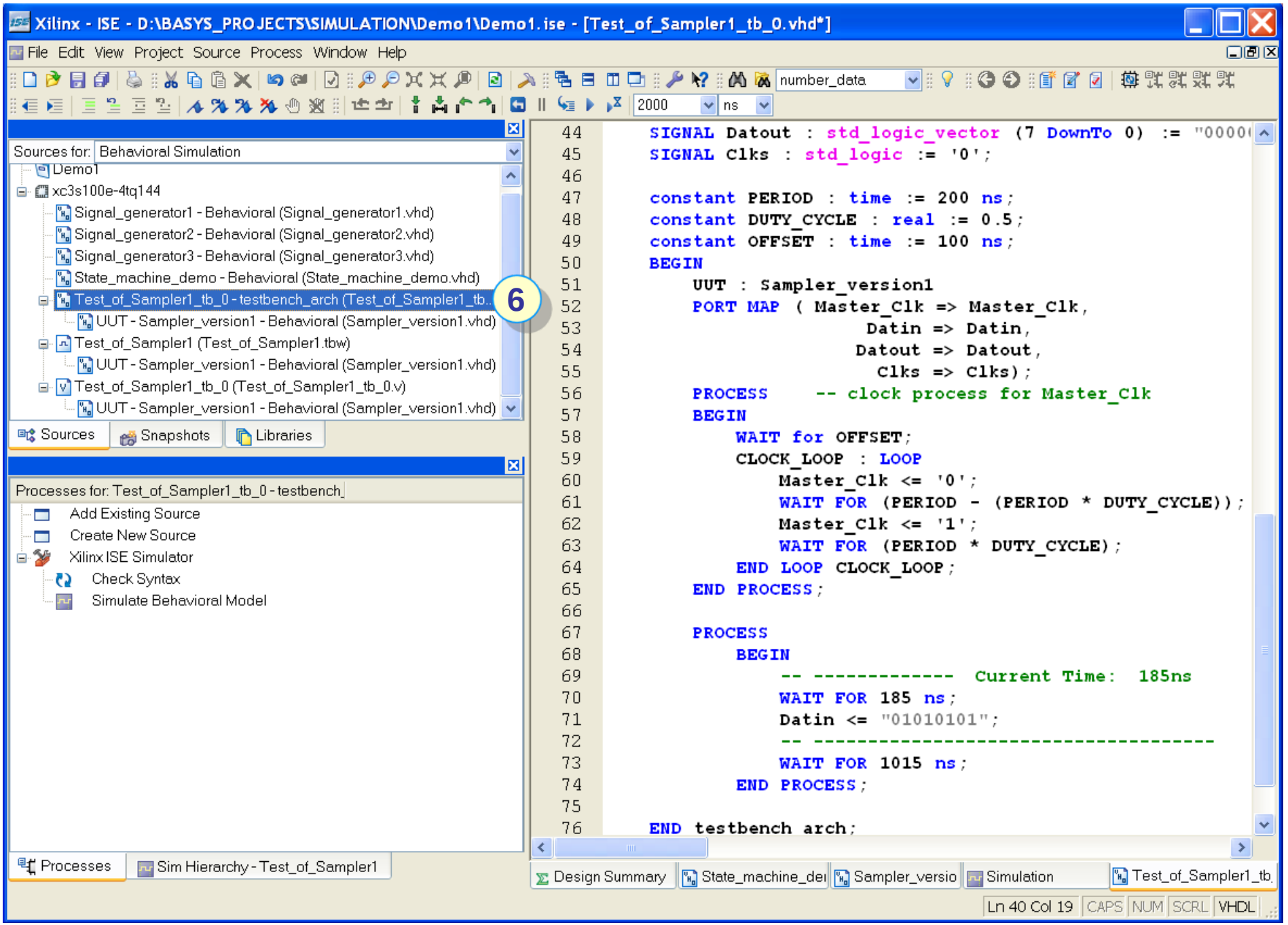


Adding the TestBench to the Project

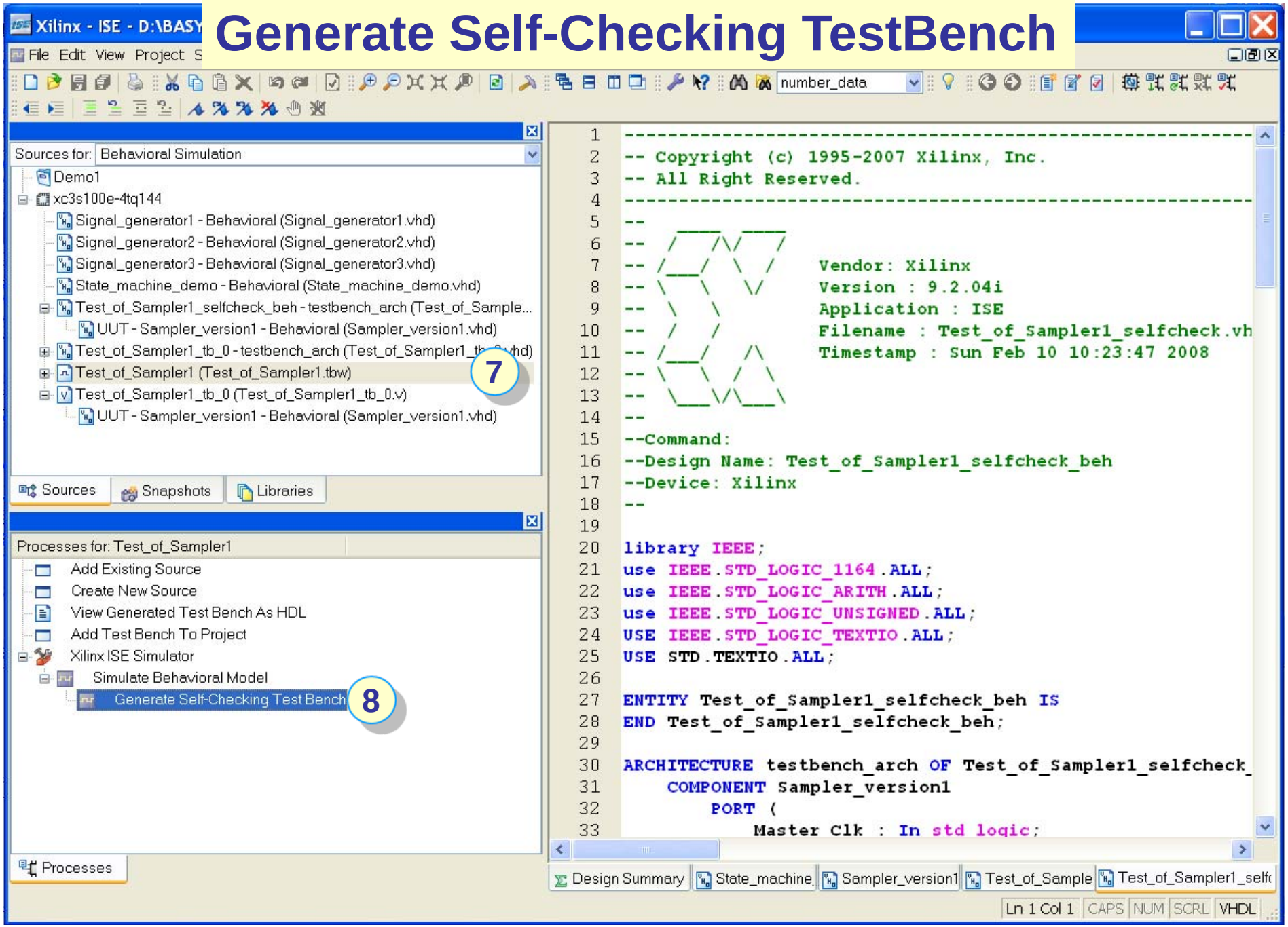


Changing Prefered Language to VHDL




Xilinx - ISE - D:\BASYS_PROJECTS\SIMULATION\Demo1\Demo1.ise - [Test_of_Sampler1_tb_0.vhd*]
File Edit View Project Source Process Window Help
number_data
2000
ns
Sources for: Behavioral Simulation
Demo1
xc3s100e-4tq144
Signal_generator1 - Behavioral (Signal_generator1.vhd)
Signal_generator2 - Behavioral (Signal_generator2.vhd)
Signal_generator3 - Behavioral (Signal_generator3.vhd)
State_machine_demo - Behavioral (State_machine_demo.vhd)
Test_of_Sampler1_tb_0 - testbench_arch (Test_of_Sampler1_tb..
UUT - Sampler_version1 - Behavioral (Sampler_version1.vhd)
Test_of_Sampler1 (Test_of_Sampler1.tbw)
UUT - Sampler_version1 - Behavioral (Sampler_version1.vhd)
Test_of_Sampler1_tb_0 (Test_of_Sampler1_tb_0.v)
UUT - Sampler_version1 - Behavioral (Sampler_version1.vhd)
6
Sources
Snapshots
Libraries
Processes for: Test_of_Sampler1_tb_0 - testbench_
Add Existing Source
Create New Source
Xilinx ISE Simulator
Check Syntax
Simulate Behavioral Model
Processes
Sim Hierarchy - Test_of_Sampler1
44 SIGNAL Datout : std_logic_vector (7 DownTo 0) := "0000(
45 SIGNAL Clks : std_logic := '0';
46
47 constant PERIOD : time := 200 ns;
48 constant DUTY_CYCLE : real := 0.5;
49 constant OFFSET : time := 100 ns;
50 BEGIN
51 UUT : Sampler_version1
52 PORT MAP (Master_Clk => Master_Clk,
53 Datin => Datin,
54 Datout => Datout,
55 Clks => Clks);
56 PROCESS -- clock process for Master_Clk
57 BEGIN
58 WAIT for OFFSET;
59 CLOCK_LOOP : LOOP
60 Master_Clk <= '0';
61 WAIT FOR (PERIOD - (PERIOD * DUTY_CYCLE));
62 Master_Clk <= '1';
63 WAIT FOR (PERIOD * DUTY_CYCLE);
64 END LOOP CLOCK_LOOP;
65 END PROCESS;
66
67 PROCESS
68 BEGIN
69 -- ------------ Current Time: 185ns
70 WAIT FOR 185 ns;
71 Datin <= "01010101";
72 -- -----------------------------------
73 WAIT FOR 1015 ns;
74 END PROCESS;
75
76 END testbench_arch;
Design Summary
State_machine_dei
Sampler_versio
Simulation
Test_of_Sampler1_tb
Ln 40 Col 19
CAPS
NUM
SCRL
VHDL

Generate Self-Checking TestBench



Selfchecking TestBench (1)

```
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.STD_LOGIC_ARITH.ALL;
23  use IEEE.STD_LOGIC_UNSIGNED.ALL;
24  USE IEEE.STD_LOGIC_TEXTIO.ALL;
25  USE STD.TEXTIO.ALL;
26
27  ENTITY Test_of_Sampler1_selfcheck_beh IS
28  END Test_of_Sampler1_selfcheck_beh;
29
30  ARCHITECTURE testbench_arch OF Test_of_Sampler1_selfcheck_beh IS
31      COMPONENT Sampler_version1
32          PORT (
33              Master_Clk : In std_logic;
34              Datin : In std_logic_vector (7 DownTo 0);
35              Datout : Out std_logic_vector (7 DownTo 0);
36              Clks : Out std_logic
37          );
38      END COMPONENT;
39
40      SIGNAL Master_Clk : std_logic := '0';
41      SIGNAL Datin : std_logic_vector (7 DownTo 0) := "00000000";
42      SIGNAL Datout : std_logic_vector (7 DownTo 0) := "UUUUUUUU";
43      SIGNAL Clks : std_logic := 'U';
44
45      SHARED VARIABLE TX_ERROR : INTEGER := 0;
46      SHARED VARIABLE TX_OUT : LINE;
47
48      constant PERIOD : time := 200 ns;
49      constant DUTY_CYCLE : real := 0.5;
50      constant OFFSET : time := 100 ns;
51
52      BEGIN
53          UUT : Sampler_version1
54          PORT MAP (
55              Master_Clk => Master_Clk,
56              Datin => Datin,
57              Datout => Datout,
58              Clks => Clks
59          );
```

Selfchecking TestBench (2)

```
60
61          PROCESS     -- clock process for Master_Clk
62          BEGIN
63              WAIT for OFFSET;
64              CLOCK_LOOP : LOOP
65                  Master_Clk <= '0';
66                  WAIT FOR (PERIOD - (PERIOD * DUTY_CYCLE));
67                  Master_Clk <= '1';
68                  WAIT FOR (PERIOD * DUTY_CYCLE);
69              END LOOP CLOCK_LOOP;
70          END PROCESS;
71
72          PROCESS
73              PROCEDURE CHECK_Clks(
74                  next_Clks : std_logic;
75                  TX_TIME : INTEGER
76              ) IS
77                  VARIABLE TX_STR : String(1 to 4096);
78                  VARIABLE TX_LOC : LINE;
79                  BEGIN
80                  IF (Clks /= next_Clks) THEN
81                      STD.TEXTIO.write(TX_LOC, string'("Error at time="));
82                      STD.TEXTIO.write(TX_LOC, TX_TIME);
83                      STD.TEXTIO.write(TX_LOC, string'("ns Clks="));
84                      IEEE.STD_LOGIC_TEXTIO.write(TX_LOC, Clks);
85                      STD.TEXTIO.write(TX_LOC, string'(", Expected = "));
86                      IEEE.STD_LOGIC_TEXTIO.write(TX_LOC, next_Clks);
87                      STD.TEXTIO.write(TX_LOC, string'(" "));
88                      TX_STR(TX_LOC.all'range) := TX_LOC.all;
89                      STD.TEXTIO.Deallocate(TX_LOC);
90                      ASSERT (FALSE) REPORT TX_STR SEVERITY ERROR;
91                      TX_ERROR := TX_ERROR + 1;
92                  END IF;
93              END;
```

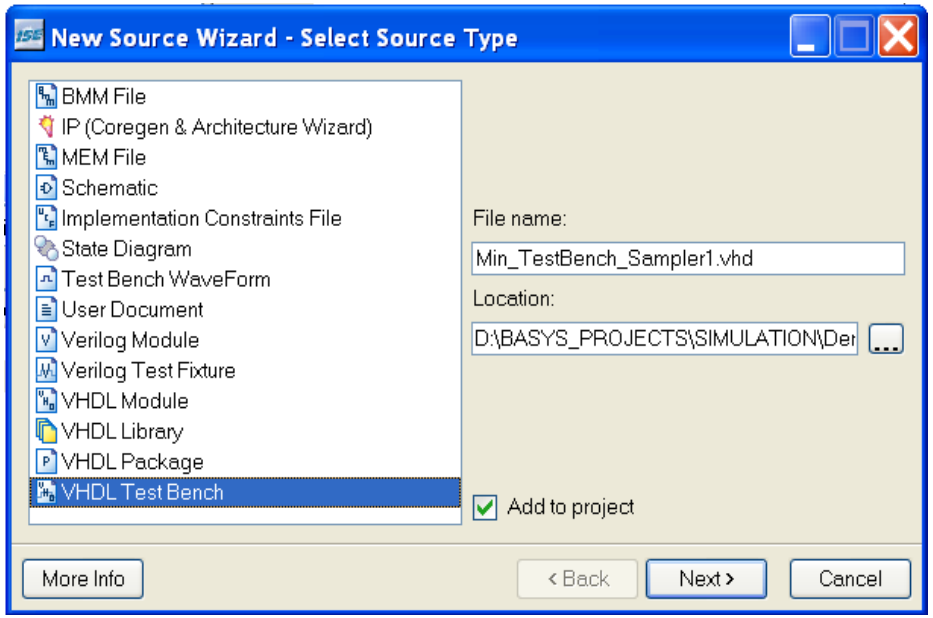
Selfchecking TestBench (3)

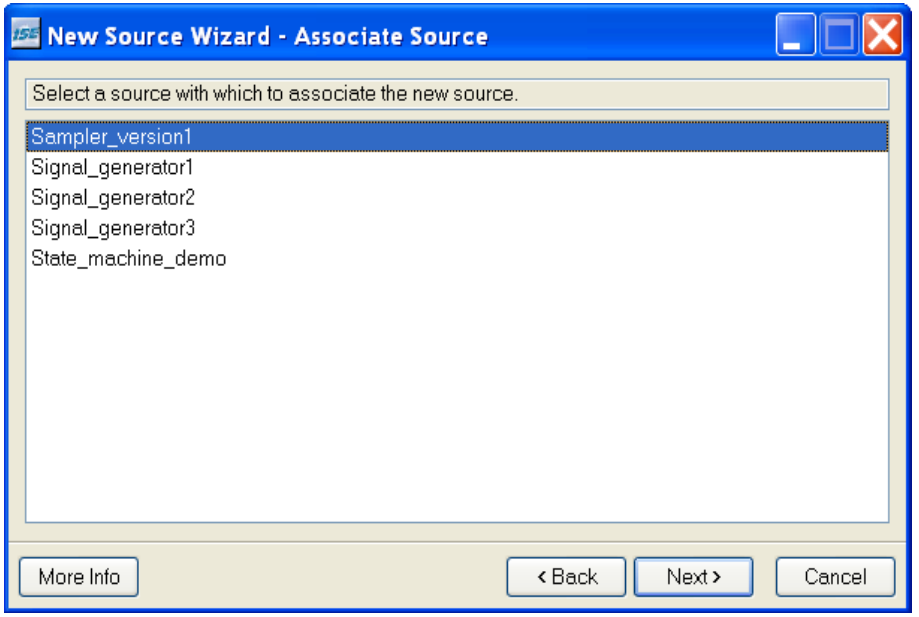
```
94          PROCEDURE CHECK_Datout(
95              next_Datout : std_logic_vector (7 DownTo 0);
96              TX_TIME : INTEGER
97          ) IS
98              VARIABLE TX_STR : String(1 to 4096);
99              VARIABLE TX_LOC : LINE;
100             BEGIN
101             IF (Datout /= next_Datout) THEN
102                 STD.TEXTIO.write(TX_LOC, string'("Error at time="));
103                 STD.TEXTIO.write(TX_LOC, TX_TIME);
104                 STD.TEXTIO.write(TX_LOC, string'("ns Datout="));
105                 IEEE.STD_LOGIC_TEXTIO.write(TX_LOC, Datout);
106                 STD.TEXTIO.write(TX_LOC, string'(", Expected = "));
107                 IEEE.STD_LOGIC_TEXTIO.write(TX_LOC, next_Datout);
108                 STD.TEXTIO.write(TX_LOC, string'(" "));
109                 TX_STR(TX_LOC.all'range) := TX_LOC.all;
110                 STD.TEXTIO.Deallocate(TX_LOC);
111                 ASSERT (FALSE) REPORT TX_STR SEVERITY ERROR;
112                 TX_ERROR := TX_ERROR + 1;
113             END IF;
114         END;
115         BEGIN
116             -- -------------  Current Time:  185ns
117             WAIT FOR 185 ns;
118             Datin <= "01010101";
119             -- ------------------------------------
120             -- -------------  Current Time:  215ns
121             WAIT FOR 30 ns;
122             CHECK_Clks('1', 215);
123             -- ------------------------------------
124             -- -------------  Current Time:  415ns
125             WAIT FOR 200 ns;
126             CHECK_Clks('0', 415);
127             -- ------------------------------------
128             -- -------------  Current Time:  615ns
129             WAIT FOR 200 ns;
130             CHECK_Clks('1', 615);
131             CHECK_Datout("10101010", 615);
132             -- ------------------------------------
133             WAIT FOR 585 ns;
```

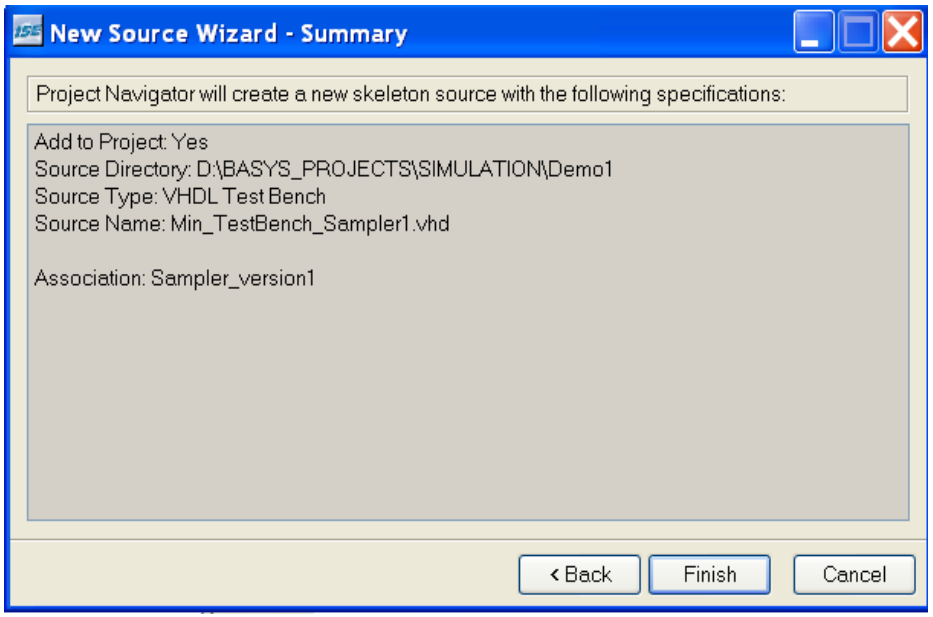
Selfchecking TestBench (4)

```
134
135                 IF (TX_ERROR = 0) THEN
136                     STD.TEXTIO.write(TX_OUT, string'("No errors or warnings"));
137                     ASSERT (FALSE) REPORT
138                       "Simulation successful (not a failure).  No problems detected."
139                       SEVERITY FAILURE;
140                 ELSE
141                     STD.TEXTIO.write(TX_OUT, TX_ERROR);
142                     STD.TEXTIO.write(TX_OUT,
143                         string'(" errors found in simulation"));
144                     ASSERT (FALSE) REPORT "Errors found during simulation"
145                          SEVERITY FAILURE;
146                 END IF;
147             END PROCESS;
148
149     END testbench_arch;
```

Creating your own TestBench (1)

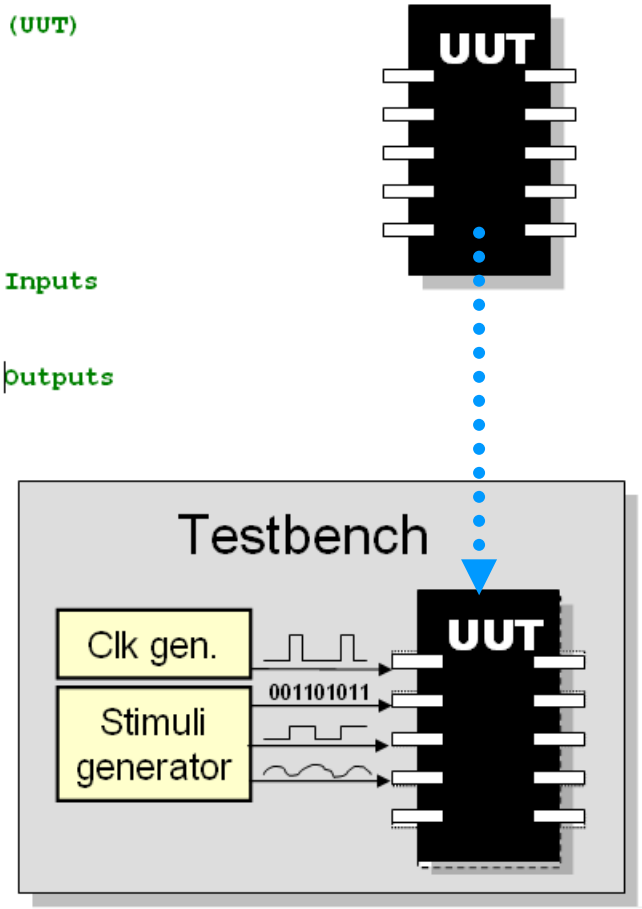


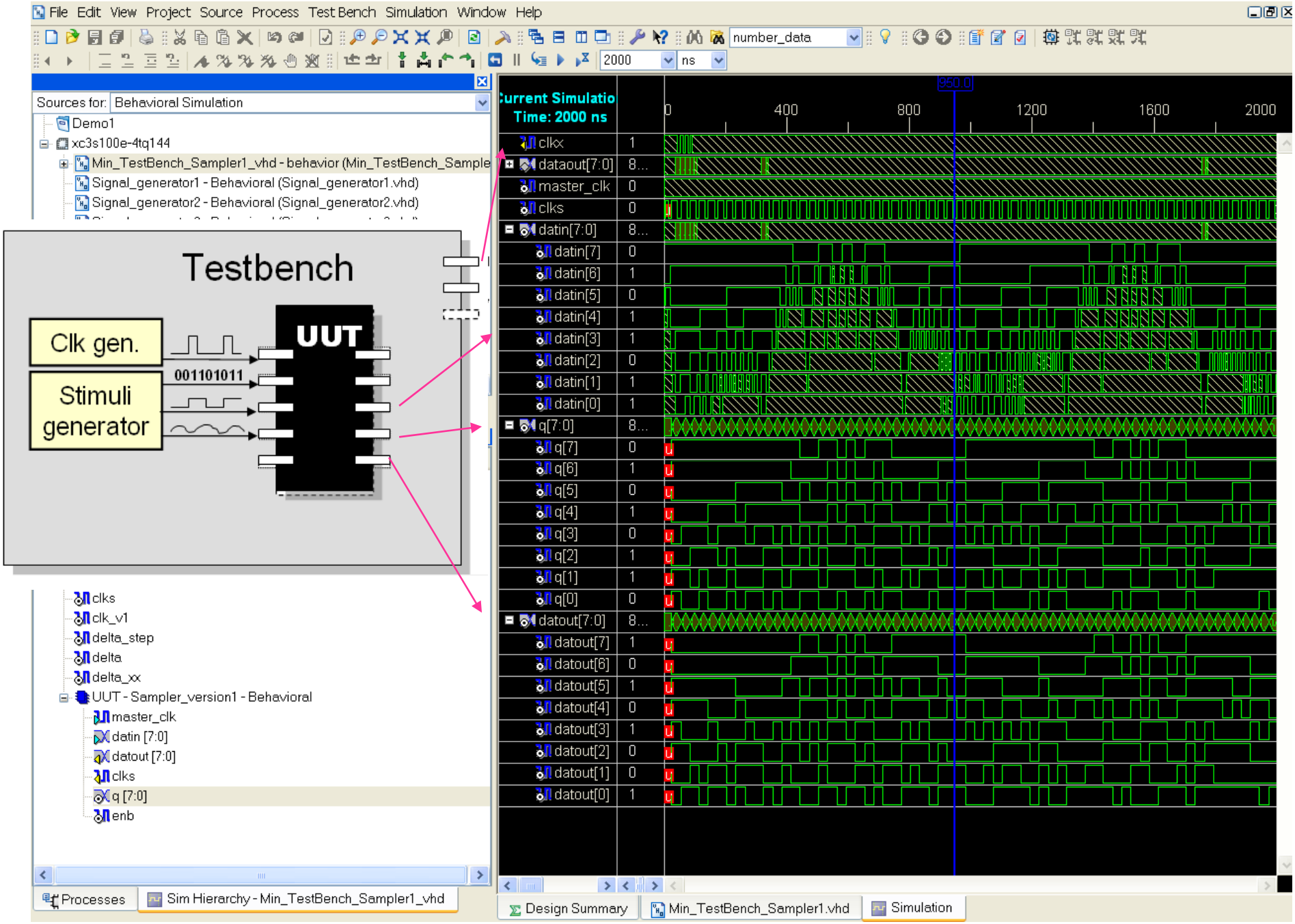




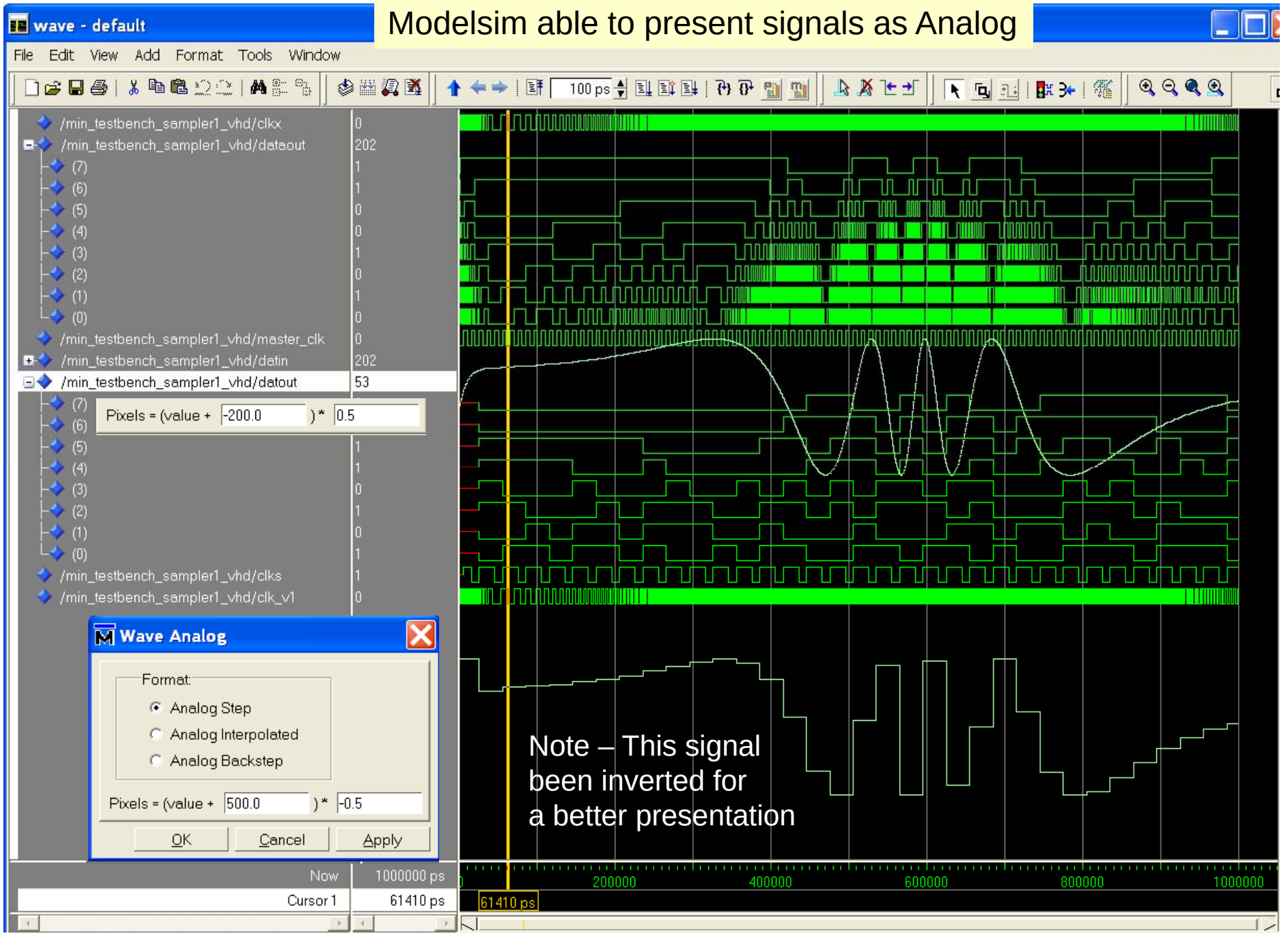
Creating your own TestBench (2)

```
-- This testbench has been automatically generated using types std_logic and
-- std_logic_vector for the ports of the unit under test.  Xilinx recommends
-- that these types always be used for the top-level I/O of a design in order
-- to guarantee that the testbench will bind correctly to the post-implementation
-- simulation model.
--------------------------------------------------------------------------------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;

ENTITY Min_TestBench_Sampler1_vhd IS
END Min_TestBench_Sampler1_vhd;

ARCHITECTURE behavior OF Min_TestBench_Sampler1_vhd IS
   ---------------------- Component Declaration for the Unit Under Test (UUT)
   COMPONENT Sampler_version1
   PORT(    Master_Clk : IN  std_logic;
            Datin :      IN  std_logic_vector(7 downto 0);
            Datout :     OUT std_logic_vector(7 downto 0);
            Clks :       OUT std_logic);
   END COMPONENT;

   ------------------------------------------------------------------ Inputs
   SIGNAL Master_Clk : std_logic := '0';
   SIGNAL Datin :      std_logic_vector(7 downto 0) := (others=>'0');
   ------------------------------------------------------------------ Outputs
   SIGNAL Datout :     std_logic_vector(7 downto 0);
   SIGNAL Clks :       std_logic;
BEGIN
   -- ------------------------------- Instantiate the Unit Under Test
   UUT: Sampler_version1 PORT MAP( Master_Clk => Master_Clk,
                                   Datin      => Datin,
                                   Datout     => Datout,
                                   Clks       => Clks );
   tb : PROCESS
   BEGIN
      -- Wait 100 ns for global reset to finish
      wait for 100 ns;
      -- Place stimulus here
      wait; -- will wait forever
   END PROCESS;
END;
```




File Edit View Project Source Process Test Bench Simulation Window Help
number_data
2000
ns
Sources for: Behavioral Simulation
Demo1
xc3s100e-4tq144
Min_TestBench_Sampler1_vhd - behavior (Min_TestBench_Sample
Signal_generator1 - Behavioral (Signal_generator1.vhd)
Signal_generator2 - Behavioral (Signal_generator2.vhd)
Testbench
Clk gen.
Stimuli generator
001101011
UUT
clks
clk_v1
delta_step
delta
delta_xx
UUT - Sampler_version1 - Behavioral
master_clk
datin [7:0]
datout [7:0]
clks
q [7:0]
enb
Processes
Sim Hierarchy - Min_TestBench_Sampler1_vhd
Current Simulation Time: 2000 ns
950.0
0
400
800
1200
1600
2000
clkx
dataout[7:0]
master_clk
clks
datin[7:0]
datin[7]
datin[6]
datin[5]
datin[4]
datin[3]
datin[2]
datin[1]
datin[0]
q[7:0]
q[7]
q[6]
q[5]
q[4]
q[3]
q[2]
q[1]
q[0]
datout[7:0]
datout[7]
datout[6]
datout[5]
datout[4]
datout[3]
datout[2]
datout[1]
datout[0]
Design Summary
Min_TestBench_Sampler1.vhd
Simulation

Modelsim able to present signals as Analog



Creating your own TestBench (3)

```
29  LIBRARY ieee;
30  USE ieee.std_logic_1164.ALL;
31  USE ieee.std_logic_unsigned.all;
32  USE ieee.numeric_std.ALL;
33  use IEEE.STD_LOGIC_ARITH.ALL;
34  use IEEE.MATH_REAL.ALL;              -- This contains the SIN( )
35
36  ENTITY Min_TestBench_Sampler1_vhd IS
37      Port ( CLKX:       out  STD_LOGIC;
38             Dataout :   inout  STD_LOGIC_VECTOR (7 downto 0));
39  END Min_TestBench_Sampler1_vhd;
40
41  ARCHITECTURE behavior OF Min_TestBench_Sampler1_vhd IS
42     ---------------------- Component Declaration for the Unit Under Test (UUT)
43     COMPONENT Sampler_version1
44     PORT(    Master_Clk : IN  std_logic;
45              Datin :      IN  std_logic_vector(7 downto 0);
46              Datout :     OUT std_logic_vector(7 downto 0);
47              Clks :       OUT std_logic);
48     END COMPONENT;
49
50     ------------------------------------------------------------------- Inputs
51     SIGNAL Master_Clk : std_logic := '0';
52     SIGNAL Datin :      std_logic_vector(7 downto 0) := (others=>'0');
53     ------------------------------------------------------------------ Outputs
54     SIGNAL Datout :     std_logic_vector(7 downto 0);
55     SIGNAL Clks :       std_logic;
56     ------------------------------------------------------------------------
57     signal Clk_v1:          STD_LOGIC := '0';
58     -- Please note - Shared variables can be used for interprocess data
59     -- exchange. Moreover can they be observed under a simulation as well
60     shared variable Delta_step: real := 100.0;
61     shared variable Delta:      real := 1000.0;
62     shared variable Delta_xx:   real := -1.0;
```

Creating your own TestBench (4)

```
63  BEGIN
64     -- ------------------------------- Instantiate the Unit Under Test (UUT)
65     UUT: Sampler_version1 PORT MAP( Master_Clk => Master_Clk,
66                                     Datin      => Datin,
67                                     Datout     => Datout,
68                                     Clks       => Clks );
69
70    Master_Clk <= not Master_Clk after 5 ns;  -- 100 MHz master clock
71    Datin      <= DataOut;
72    CLKX       <= Clk_v1;
73
74     ---------------------------------------------------------------------
75     -- This process creates a clk-signal with a variable frequency
76     -- not use if its useful in practice, but it demonstrates what can
77     -- be done.
78     -- The shared variable "Delta" will decrease with the value "Delta_step"
79     -- for each step will the "Delta_step" value change with "Delta_xx"
80     ---------------------------------------------------------------------
81     Clk_generator: process
82     begin
83        Clk_v1 <= not Clk_v1;   -- Toogle the Clk
84        Delta := 1000.0;        -- Ready for a new count down
85        while Delta>0.0 loop    -- while not done
86           wait for 10 ps;      -- adjust this if needed
87           Delta := Delta - Delta_step;  -- one step down
88        end loop;
89
90        if Delta_xx < 0.0 then
91           if Delta_step < 2.0 then
92              Delta_xx := 0.1;
93           end if;
94        else
95           if Delta_step > 198.0 then
96              Delta_xx := -0.1;
97           end if;
98        end if;
99        Delta_Step := Delta_step + Delta_xx;
100    end process Clk_generator;
```

Creating your own TestBench (5)

```
102    ------------------------------------------------------------------
103    -- This process driven by an external clock signal
104    -- the statement "wait until rising_edge( clk_v1)" do the trick
105    ------------------------------------------------------------------
106    Sinus_generator: process
107       constant Umax:      integer := 127;      -- Max amplitude
108       constant f:         real    := 2.0E6;   -- Frequency [Hz]
109       constant Tper:      real    := 1.0/f;    -- Period of fr.
110    -- If you can find a way to convert real to time please let me know
111       constant Delta:     real    := 1000.0E-12; -- delta time - sec
112       constant DeltaWait: time    := 1000 ps;     -- delta time - ps
113    ------------------------------------------------------------------
114       variable t:     Real := 0.0;   -- Actual time
115       variable angle: real := 0.0;   -- Actual angle in radians
116       variable Usin:  real := 0.0;   -- The sin value [real]
117       variable Usin_int: integer;    -- The sin value as integer
118    begin
119       wait until rising_edge( Clk_v1);
120
121       angle    := 2.0 * MATH_PI * t * f;     -- calculate angle
122       t        := t + Delta;                 -- next time
123       Usin     := real(Umax)*( SIN( angle)+1.0);  -- Usin calculation
124       Usin_int := integer(Usin);            -- convert real to integer
125       Dataout  <= conv_std_logic_vector( Usin_int, 8);  -- to vector
126    end process sinus_generator;
127
128    ------------------------------------ not in use ----------------
129    tb : PROCESS
130    BEGIN
131       -- Wait 100 ns for global reset to finish
132       wait for 100 ns;
133       -- Place stimulus here
134       wait; -- will wait forever
135    END PROCESS;
136 END;
```