

Моделирование цифровых устройств, проектируемых на основе ПЛИС фирмы Xilinx, средствами ISIM в САПР ISE Design Suite

Валерий ЗОТОВ
walerry@km.ru

Введение

В ранних версиях САПР серии Xilinx ISE (Integrated Synthesis Environment/Integrated Software Environment) в течение длительного периода времени единственным инструментом верификации разрабатываемых устройств был пакет моделирования ModelSim, выпускаемый корпорацией Model Technology (www.model.com), которая входит в состав компании Mentor Graphics (www.mentor.com). Этот пакет представляет собой самостоятельный программный продукт, который необходимо приобретать и устанавливать дополнительно. Все редакции САПР серии Xilinx ISE комплектовались только свободно распространяемой версией этого пакета — ModelSim Xilinx Edition (XE) Starter [1, 2], которая имеет существенные ограничения по объему исполняемого HDL-кода. В процессе эволюционного развития систем проектирования фирмы Xilinx появились собственные встроенные средства моделирования — ISE Simulator, первоначальные версии которых уступали по функциональным возможностям пакету ModelSim XE. В результате дальнейшего постоянного совершенствования этих средств к настоящему времени сформировался полнофункциональный инструмент HDL-моделирования — ISIM. Компания Xilinx предоставляет его в составе обновленной САПР ISE Design Suite.

В статье рассматриваются основные характеристики и режимы функционирования встроенных средств HDL-моделирования ISIM, а также их практическое использование для верификации разрабатываемых цифровых устройств, реализуемых на базе ПЛИС фирмы Xilinx.

Основные характеристики и режимы функционирования встроенных средств HDL-моделирования Xilinx ISIM

Последние версии интегрированных средств моделирования Xilinx ISIM поддерживают различные варианты стандартов языков VHDL и Verilog, в том числе

IEEE VHDL Std 1076-2000 и IEEE Verilog Std 1364-2001. При этом рассматриваемые средства предоставляют разработчику возможность не только раздельного, но и смешанного моделирования проектируемых устройств, исходные описания которых включают в себя модули VHDL- и Verilog-кода.

Встроенные средства моделирования Xilinx ISIM обладают полной совместимостью со спецификацией OVI 3.0 стандартного формата описания задержек SDF (Standard Delay Format), обеспечивающей возможность обратной аннотации временных параметров. В этих средствах предусмотрена поддержка инкрементного режима компиляции модулей исходного описания разрабатываемого устройства, что позволяет существенно сократить суммарное время моделирования. С помощью рассматриваемых средств можно осуществлять моделирование проектируемых устройств, включающих различные специализированные аппаратные ресурсы ПЛИС фирмы Xilinx, в том числе высокоскоростные последовательные приемопередатчики Multi-Gigabit Transceivers (MGT), процессорные блоки PPC и модули PCIe. Эти средства обеспечивают возможность эффективной отладки разрабатываемых описаний проектируемого устройства на языках VHDL и Verilog на основе пошагового режима, позволяющего проследить исполнение HDL-кода.

В последних версиях средств Xilinx ISIM предусмотрена поддержка моделирования разрабатываемых устройств на аппаратном уровне с применением отладочных плат, выполненных на базе соответствующих кристаллов программируемой логики с архитектурой FPGA (Field Programmable Gate Array) [3].

Для управления процессом моделирования при использовании средств Xilinx ISIM можно применять один из следующих механизмов:

- в интерактивном режиме — элементы графического пользовательского интерфейса (меню, диалоговые панели, кнопки);
- в режиме интерактивной командной строки — директивы языка Tcl (Tool command language);

- в неинтерактивном пакетном режиме — командный файл, содержащий заранее сформированную последовательность Tcl-директив.

Использование графического пользовательского интерфейса GUI (Graphical User Interface) обеспечивает гибкость и наглядность управления моделированием и представления результатов этого процесса, поэтому в дальнейшем будет рассмотрен только этот режим функционирования средств Xilinx ISIM. Запуск встроенных средств моделирования в указанном режиме осуществляется в среде управляющей оболочки САПР ISE Design Suite Project Navigator [1].

Этапы моделирования цифровых устройств, поддерживаемые встроенными средствами моделирования Xilinx ISIM

Средства Xilinx ISIM позволяют выполнять моделирование разрабатываемых устройств на различных стадиях процесса проектирования. При осуществлении основных этапов этого процесса САПР ISE Design Suite предоставляет возможность формирования различных видов моделей разрабатываемого устройства. Степень достоверности генерируемых моделей возрастает после выполнения каждого основного этапа проектирования. Наличие нескольких этапов моделирования в процессе проектирования обеспечивает возможность обнаружения вероятных ошибок на ранних стадиях этого процесса, тем самым сокращается суммарное время разработки устройства. Количество этапов моделирования, поддерживаемых средствами Xilinx ISIM, определяется типом архитектуры ПЛИС, выбранным для реализации проектируемого устройства. При использовании кристаллов программируемой логики с архитектурой FPGA моделирование можно осуществлять после выполнения следующих основных фаз процесса проектирования:

- создания модулей исходного HDL-описания разрабатываемого устройства;
- трансляции синтезированного описания (Translate);

- отображения логического описания на физические ресурсы ПЛИС (MAP);
- размещения и трассировки проекта в кристалле (Place and Route).

На рис. 1 представлено расположение этапов моделирования в маршруте проектирования цифровых устройств на базе ПЛИС семейств FPGA.

До выполнения этапа синтеза сформированного HDL-описания проекта генерируется поведенческая модель, которая позволяет выполнить его предварительную верификацию. В этой модели фактически не учитываются особенности архитектуры и временные характеристики кристалла программируемой логики, на базе которого предполагается реализация проектируемого устройства. Кроме того, в поведенческой модели не принимаются во внимание ограничения, накладываемые применяемыми средствами синтеза. В процессе поведенческого моделирования проекта могут использоваться библиотеки UNISIM, UniMacro и XilinxCoreLib, компоненты которых имеют единичные задержки. Часто в литературе этот процесс называют моделированием на уровне регистровых передач (Register Transfer Level, RTL). Поведенческое моделирование позволяет обнаружить только логические и синтаксические ошибки в описании разрабатываемого устройства.

После трансляции синтезированного описания можно выполнять этап функционального моделирования проектируемого устройства на уровне вентилях. На этой стадии проектирования отсутствует информация о реальных значениях задержек компонентов и линий распространения сигналов, поэтому моделирование проводится с единичными значениями задержек. Указанный этап моделирования позволяет проверить корректность результатов синтеза и последующей трансляции сформированного описания разрабатываемого устройства. Результаты, полученные на этом этапе, могут отличаться от результатов поведенческого моделирования из-за особенностей функционирования средств синтеза. Информация об ограничениях средств синтеза XST (Xilinx Synthesis Technology), входящих в состав САПР ISE Design Suite, представлена в [4–6].

При успешном завершении фазы отображения логического описания проекта на физические ресурсы ПЛИС формируется частичная временная модель разрабатываемого устройства. Она включает в себя информацию о задержках отдельных блоков проекта, но не содержит значений задержек, вносимых трассировочными ресурсами кристалла программируемой логики. На этом и последующем этапах моделирования применяется библиотека SIMPRIM, которая содержит описание элементов на уровне ресурсов кристалла и позволяет учитывать информацию о задержках распространения сигналов, которая содержится в соответствующих файлах, имеющих стандартный формат SDF.

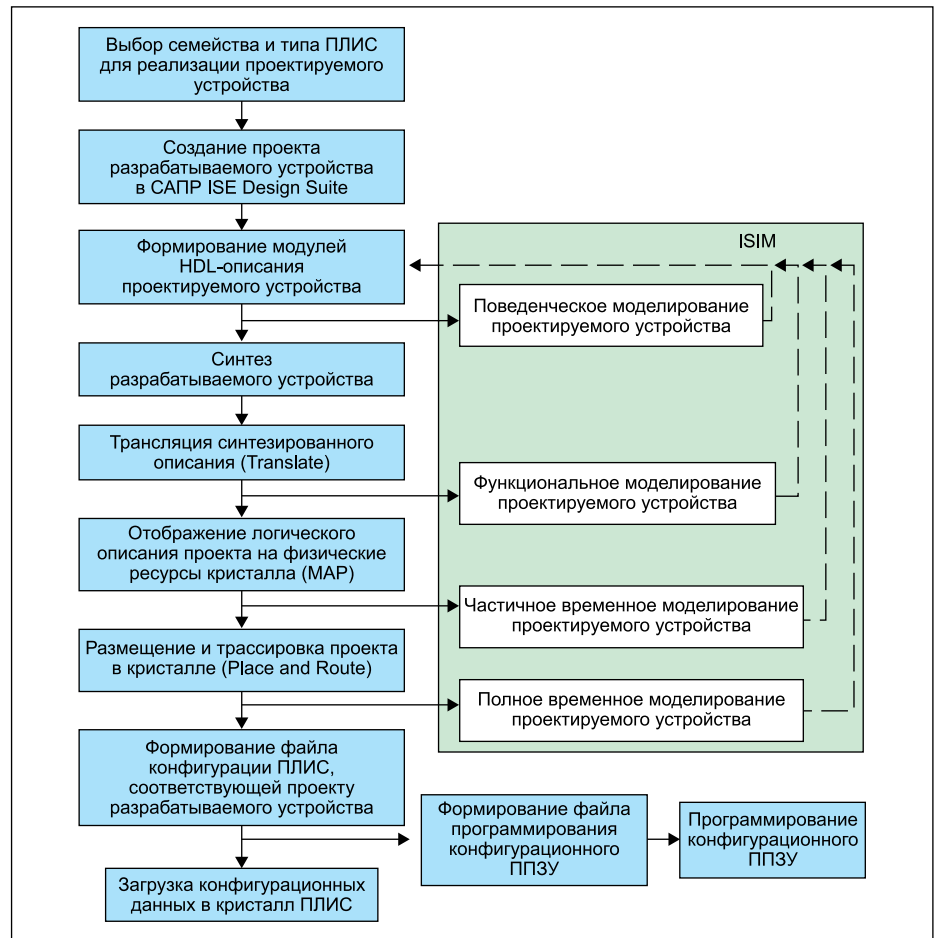


Рис. 1. Этапы моделирования в процессе проектирования цифровых устройств на базе ПЛИС с архитектурой FPGA

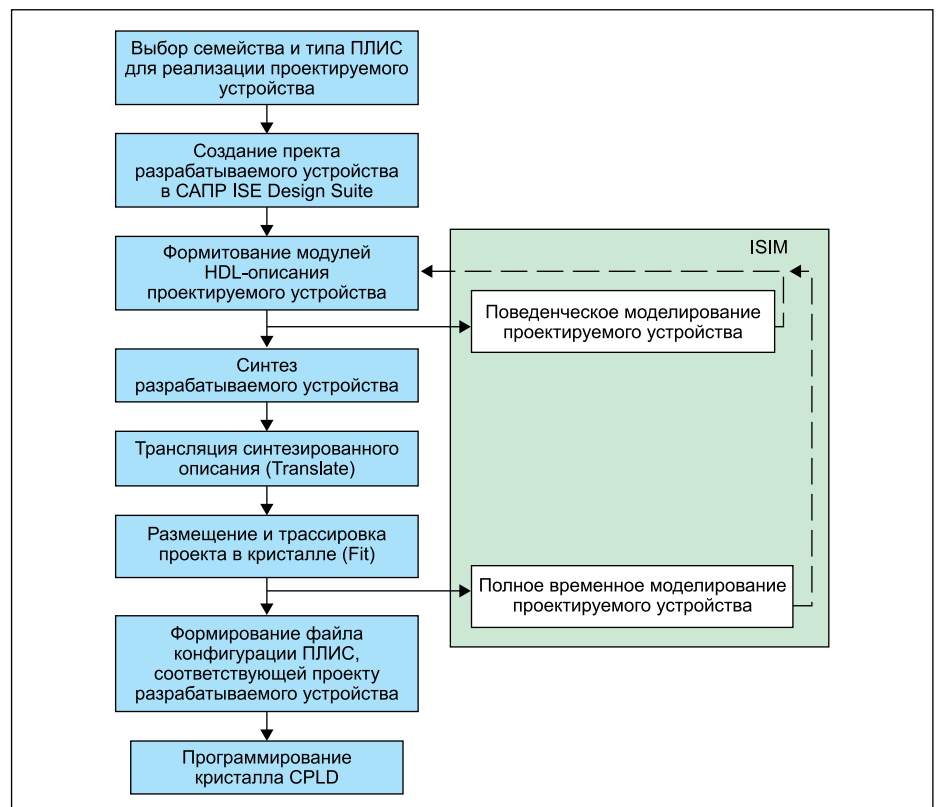


Рис. 2. Этапы моделирования в процессе проектирования цифровых устройств на базе ПЛИС с архитектурой CPLD

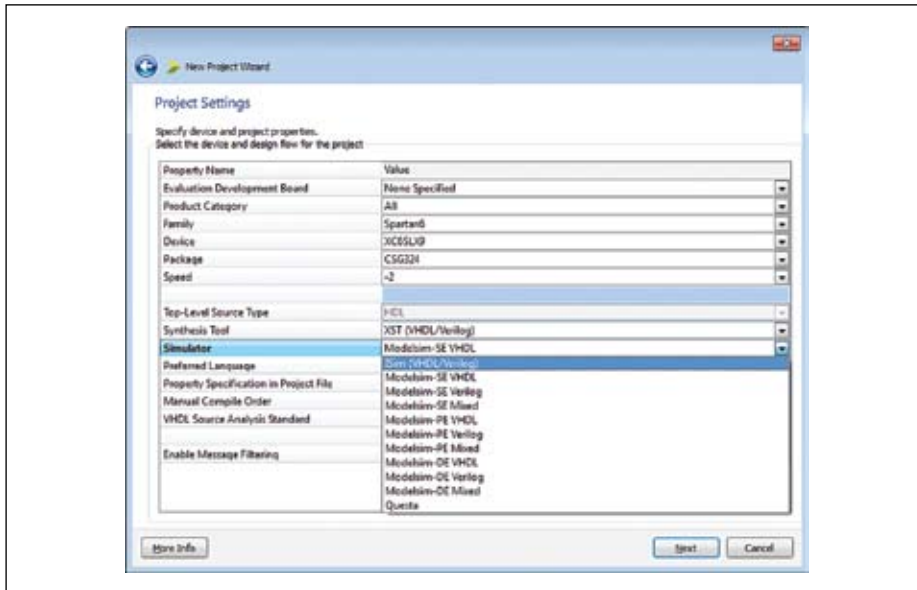


Рис. 3. Выбор встроенных средств моделирования Xilinx ISIM при создании нового проекта в САПР ISE Design Suite

Полная временная модель разрабатываемого устройства, включающая значения всех задержек распространения сигналов, вносимых различными ресурсами ПЛИС, генерируется только после успешного окончания фазы размещения и трассировки проекта в кристалле. Использование этой модели для верификации проектируемого устройства позволяет получить наиболее точные результаты, соответствующие временным характеристикам выбранного кристалла программируемой логики.

При проектировании цифровых устройств на базе ПЛИС с архитектурой CPLD (Complex Programmable Logic Device) средства Xilinx ISIM поддерживают только два из рассмотренных выше этапов верификации — этапы поведенческого и полного временного моделирования. Расположение этапов моделирования в процессе проектирования устройств, реализуемых на основе кристаллов программируемой логики семейств CPLD, показано на рис. 2.

Прежде чем приступить к изучению моделирования средствами Xilinx ISIM на этих этапах, рассмотрим процесс подготовки тестового модуля проекта, который необходим для их осуществления. Для выполнения различных этапов моделирования можно использовать один и тот же модуль описаний тестовых воздействий. Допускается также возможность применения нескольких тестовых файлов в проекте.

Формирование тестового модуля проекта

Чтобы использовать встроенные средства Xilinx ISIM в процессе разработки цифровых устройств, необходимо при создании нового проекта, процедура которого подробно рассмотрена в [7], выбрать указанный инструмент HDL-моделирования. Для

этого следует в диалоговой панели **Project Settings** «мастера» формирования нового проекта **New Project Wizard** воспользоваться полем выбора параметра **Simulator**. В выпадающем списке этого поля нужно выбрать строку **ISIM (VHDL/Verilog)**, как показано на рис. 3. После выбора средств моделирования Xilinx ISIM необходимо в поле выбора **Preferred Language** указать язык HDL, используемый для формирования моделей разрабатываемого устройства. (В дальнейшем рассматривается моделирование устройств, описание которых представлено на языке VHDL.) При этом следует с помощью поля выбора VHDL Source Analysis Standard определить вариант стандарта языка, в соответствии с которым будут формироваться модули исходного описания проектируемого устройства. Выпадающий список возможных значений этого поля выбора содержит два варианта — VHDL-93 и VHDL-200X. Значение VHDL-93 соответствует варианту стандарта IEEE VHDL Std 1076-1993, а VHDL-200X — варианту IEEE VHDL Std 1076-2000.

Затем с помощью интегрированного HDL-редактора САПР ISE Design Suite нужно сформировать модули исходного описания проектируемого устройства. Последовательность действий по созданию модулей HDL-описания разрабатываемых устройств представлена в [7]. В дальнейшем процесс подготовки тестовых модулей и выполнение этапов моделирования для наглядности будет продемонстрирован на примере устройства `revers_jcount`, включающего в себя два реверсивных четырехразрядных счетчика Джонсона, функционирующих с противоположными направлениями счета. Описание этого устройства на языке VHDL, сформированное с помощью «мастера» подготовки модулей исходного описания проекта **New Source Wizard**, выглядит следующим образом:

```

-----
-- Company:
-- Engineer:
-- Create Date: 12:17:35 01/05/2013
-- Design Name:
-- Module Name: revers_jcount - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
-- Dependencies:
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

--
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
--
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
--

entity revers_jcount is
    Port (
        CLK_INPUT : in STD_LOGIC;
        LEFT_CTRL : in STD_LOGIC;
        RIGHT_CTRL : in STD_LOGIC;
        STOP_CTRL : in STD_LOGIC;
        A_OUTPUT : out STD_LOGIC_VECTOR (3 downto 0);
        B_OUTPUT : out STD_LOGIC_VECTOR (3 downto 0)
    );
end revers_jcount;

--
architecture Behavioral of revers_jcount is
--
component jcount
    port (
        LEFT : in STD_LOGIC;
        RIGHT : in STD_LOGIC;
        STOP : in STD_LOGIC;
        CLK : in STD_LOGIC;
        Q : out STD_LOGIC_VECTOR (3 downto 0)
    );
end component;

--
begin
--
D1: jcount
    port map (
        LEFT => LEFT_CTRL,
        RIGHT => RIGHT_CTRL,
        STOP => STOP_CTRL,
        CLK => CLK_INPUT,
        Q => A_OUTPUT
    );
D2: jcount
    port map (
        LEFT => RIGHT_CTRL,
        RIGHT => LEFT_CTRL,
        STOP => STOP_CTRL,
        CLK => CLK_INPUT,
        Q => B_OUTPUT
    );
end Behavioral;

```

В приведенном описании используются два экземпляра компонента `jcount`, представляющего счетчик Джонсона, VHDL-описание которого имеет следующий вид:

```

library IEEE;
use IEEE.std_logic_1164.all;
--
entity jcount is
    port (
        LEFT : in STD_LOGIC;
        RIGHT : in STD_LOGIC;
        STOP : in STD_LOGIC;
        CLK : in STD_LOGIC;
        Q : out STD_LOGIC_VECTOR (3 downto 0)
    );
end jcount;
--
architecture jcount_arch of jcount is
    signal DIR : STD_LOGIC := '0';
    signal RUN : STD_LOGIC := '0';
    signal Q_int : STD_LOGIC_VECTOR (3 downto 0) := "0000";
begin
--

```

```

process (CLK, RIGHT, LEFT, STOP, RUN, DIR, Q_int)
begin
  if (CLK'event and CLK='1') then
    -- DIR register:
    if (RIGHT='0') then
      DIR <= '0';
    elsif (LEFT='0') then
      DIR <= '1';
    end if;
    -- RUN register:
    if (STOP='0') then
      RUN <= '0';
    elsif (LEFT='0' or RIGHT='0') then
      RUN <= '1';
    end if;
    -- Counter section:
    if (RUN='1') then
      if (DIR='1') then
        Q_int(3 downto 1) <= Q_int(2 downto 0);
        Q_int(0) <= not Q_int(3);
      else
        Q_int(2 downto 0) <= Q_int(3 downto 1);
        Q_int(3) <= not Q_int(0);
      end if;
    end if;
    Q <= Q_int;
  end process;
end jcount_arch;

```

После формирования модулей исходного описания разрабатываемого устройства нужно создать тестовый модуль проекта, который фактически содержит представленные модели испытательного стенда для этого устройства на языке описания аппаратуры HDL, используемом в процессе проектирования. Структура тестового модуля для языка VHDL включает в себя следующие разделы:

- ссылки на используемые библиотеки и пакеты;
- декларацию объекта описания (ENTITY);
- определение архитектуры этого объекта.

В состав раздела определения архитектуры объекта описания ENTITY входят следующие подразделы:

- декларация компонента, представляющего модуль описания верхнего уровня иерархии проектируемого устройства;
- декларация сигналов, используемых для подачи входных тестовых воздействий и контроля выходных реакций моделируемого устройства;
- оператор создания экземпляра компонента с подключением соответствующих сигналов;

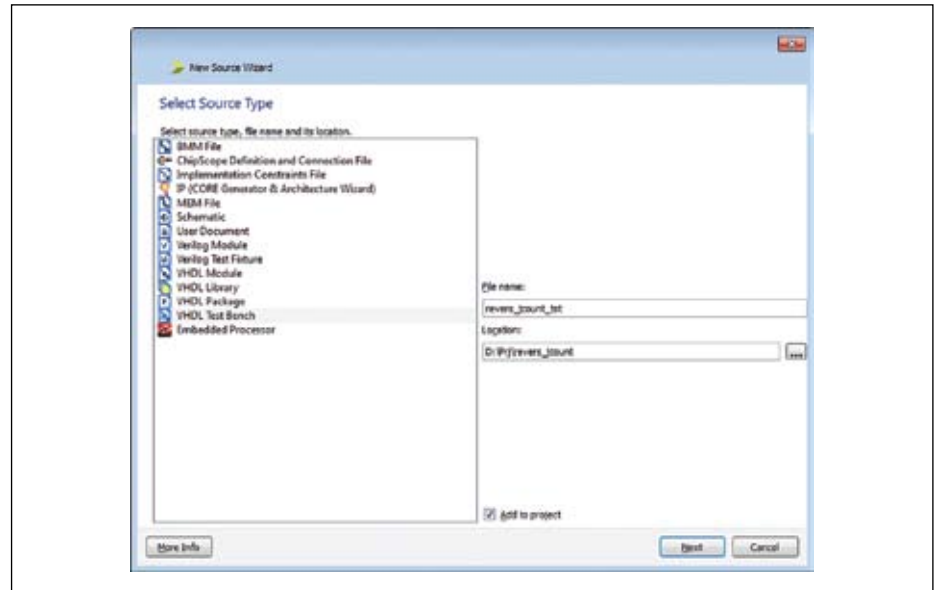



Рис. 4. Подготовка основы тестового модуля проекта с помощью «мастера» New Source Wizard

- выражения, описывающие поведение входных тестовых сигналов.

В данном случае под объектом описания ENTITY мы понимаем модель испытательного стенда в целом, поэтому он не имеет интерфейсных сигналов (портов). Архитектура этого объекта содержит единственный компонент, представляющий разрабатываемое устройство. «Мастер» подготовки модулей исходного описания проекта **New Source Wizard** в САПР ISE Design Suite позволяет на основе HDL-описания разрабатываемого устройства автоматически сформировать основу тестового файла, которая содержит все перечисленные выше разделы его структуры, за исключением секции, описывающей поведение входных сигналов. Для запуска этого «мастера» можно воспользоваться кнопкой  на оперативной панели либо командой **New Source** из раздела **Project** основного меню **Project Navigator** или контекстно-зависимого всплывающего меню, вызываемого щелчком правой кнопки мыши.

В открывшейся диалоговой панели **Select Source Type** в качестве типа нового модуля следует выбрать строку **VHDL Test Bench**, как показано на рис. 4.

Затем нужно задать имя создаваемого тестового файла в поле редактирования **File Name** и указать место его расположения на диске в поле редактирования **Location**, после чего нажатием клавиши **Next** перейти к следующей диалоговой панели с заголовком **Associate Source**. Следует обратить внимание на то, что задаваемое название тестового модуля определяет имя объекта, представляющего моделируемую систему. Диалоговая панель **Associate Source**, вид которой приведен на рис. 5, содержит список существующих модулей исходного описания проекта. В этом списке необходимо выделить название компонента проекта, для которого создается тестовый модуль, поместив курсор на соответствующую строку и щелкнув левой кнопкой мыши. Чтобы сформировать основу тестового модуля для разрабатываемого

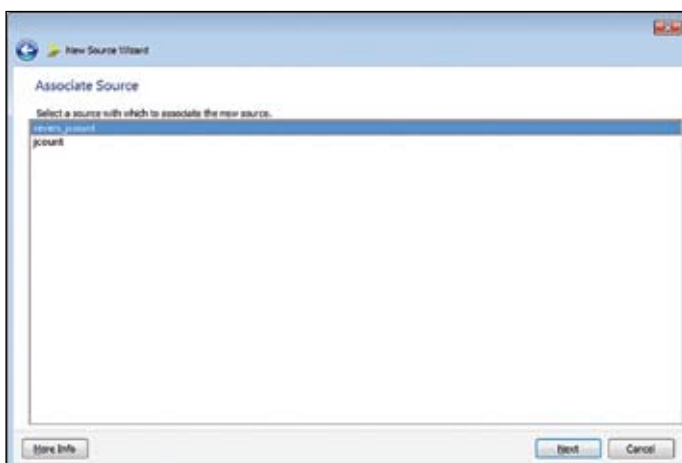


Рис. 5. Выбор компонента проекта, для которого создается основа тестового модуля

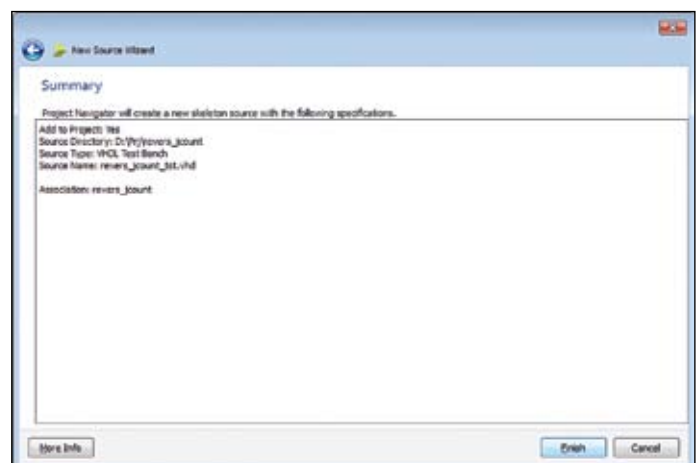


Рис. 6. Вид панели Summary «мастера» New Source Wizard

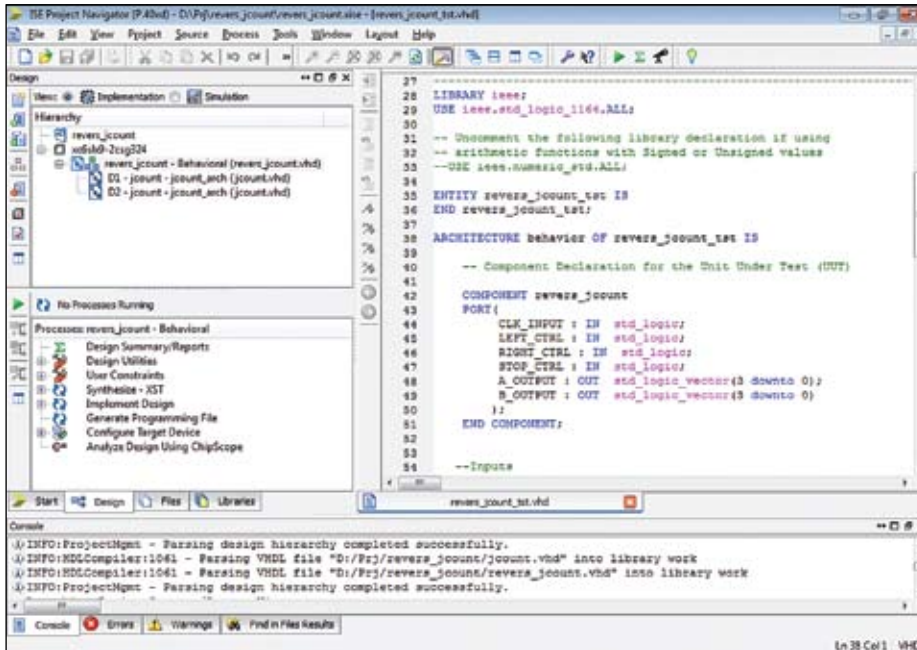



Рис. 7. Отображение сформированной основы тестового модуля в окне встроенного HDL-редактора

мого устройства в целом, нужно в представленном списке выделить строку с названием модуля исходного описания проекта верхнего уровня иерархии. Сделанный выбор подтверждается нажатием клавиши **Next**, расположенной в нижней части диалоговой панели **Associate Source**.

После этого открывается информационная панель **Summary**, отображающая значения всех параметров создаваемого тестового модуля, вид которой представлен на рис. 6. Если необходимо изменить значение какого-либо параметра, то кнопка **Back** позволяет вернуться к предыдущей диалоговой панели. Процесс автоматической генерации основы тестового модуля запускается нажатием кнопки **Finish**, находящейся в нижней части информационной панели **Summary**.

При успешном завершении этого процесса автоматически открывается окно встроенного HDL-редактора **Project Navigator**, в котором отображается код сформированной основы описания модели испытательной системы (рис. 7).

Далее необходимо дополнить сформированный модуль выражениями, описывающими изменение во времени входных сигналов. Для этого можно воспользоваться соответствующими шаблонами интегрированного HDL-редактора [8]. После завершения редактирования тестового файла следует обязательно его сохранить, выполнив команду **Save** из всплывающего меню **File** или нажав кнопку , расположенную на оперативной панели управления **Project Navigator**. В качестве примера далее представлен отредактированный текст завершеного тестового модуля для проекта устройства `revers_jcount`, который позволяет промоделировать основные режимы работы этого устройства:

```

-----
-- Company:
-- Engineer:
-- Create Date: 12:50:28 01/05/2013
-- Design Name:
-- Module Name: D:/Prj/revers_jcount/revers_jcount_tst.vhd
-- Project Name: revers_jcount
-- Target Device:
-- Tool versions:
-- Description:
-- VHDL Test Bench Created by ISE for module: revers_jcount
-- Dependencies:
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
-- Notes:
-- 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;
--
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;
--
ENTITY revers_jcount_tst IS
END revers_jcount_tst;
--
ARCHITECTURE behavior OF revers_jcount_tst IS
--
-- Component Declaration for the Unit Under Test (UUT)
--
COMPONENT revers_jcount
PORT(
    CLK_INPUT : IN std_logic;
    LEFT_CTRL : IN std_logic;
    RIGHT_CTRL : IN std_logic;
    STOP_CTRL : IN std_logic;
    A_OUTPUT : OUT std_logic_vector(3 downto 0);
    B_OUTPUT : OUT std_logic_vector(3 downto 0)
);
END COMPONENT;
--
--Inputs
signal CLK_INPUT : std_logic := '0';
signal LEFT_CTRL : std_logic := '0';
signal RIGHT_CTRL : std_logic := '0';
signal STOP_CTRL : std_logic := '0';
--Outputs
signal A_OUTPUT : std_logic_vector(3 downto 0);
signal B_OUTPUT : std_logic_vector(3 downto 0);
-- Clock period definitions
constant CLK_INPUT_period : time := 20 ns;

```

```

--
BEGIN
-- Instantiate the Unit Under Test (UUT)
uut: revers_jcount
PORT MAP (
    CLK_INPUT => CLK_INPUT,
    LEFT_CTRL => LEFT_CTRL,
    RIGHT_CTRL => RIGHT_CTRL,
    STOP_CTRL => STOP_CTRL,
    A_OUTPUT => A_OUTPUT,
    B_OUTPUT => B_OUTPUT
);
-- Clock process definitions
CLK_INPUT_process :process
begin
    CLK_INPUT <= '0';
wait for CLK_INPUT_period/2;
    CLK_INPUT <= '1';
wait for CLK_INPUT_period/2;
end process;
-- Stimulus process
stim_proc: process
begin
    LEFT_CTRL <= transport '1';
    RIGHT_CTRL <= transport '1';
    STOP_CTRL <= transport '1';
wait for CLK_INPUT_period*4;
    LEFT_CTRL <= transport '0';
wait for CLK_INPUT_period*2;
    LEFT_CTRL <= transport '1';
    RIGHT_CTRL <= transport '1';
wait for CLK_INPUT_period*14;
    STOP_CTRL <= transport '0';
wait for CLK_INPUT_period*2;
    STOP_CTRL <= transport '1';
wait for CLK_INPUT_period*6;
    RIGHT_CTRL <= transport '0';
wait for CLK_INPUT_period*2;
    RIGHT_CTRL <= transport '1';
wait for CLK_INPUT_period*2;
    wait;
end process;
--
END;

```

Перед выполнением каждого этапа моделирования необходимо установить требуемые значения параметров инициализации встроенных средств моделирования Xilinx ISIM и соответствующей модели. Для получения доступа к интерактивному списку этапов моделирования проектируемого устройства нужно воспользоваться группой кнопок с зависимой фиксацией **View**, которая располо-

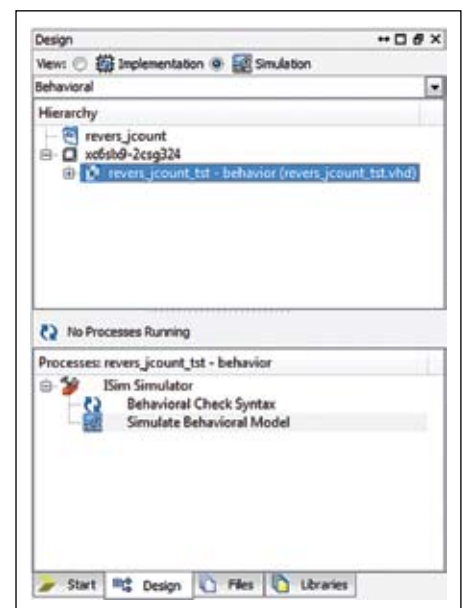


Рис. 8. Вид вкладки Design рабочей области Project Navigator при выполнении поведенческого моделирования

жена над окном исходных модулей **Project Navigator** (рис. 7). Чтобы выбрать режим отображения файлов исходных описаний, предназначенных для выполнения этапов моделирования, следует нажать кнопку **Simulation**. При этом вкладка **Design** рабочей области **Project Navigator** приобретает вид, показанный на рис. 8. В верхней части окна исходных модулей проекта появляется поле выбора, в котором отображается название этапа моделирования.

Установка значений параметров запуска средств Xilinx ISIM для поведенческого моделирования проектируемого устройства

Чтобы открыть список всех созданных тестовых модулей проекта, которые могут использоваться для поведенческой верификации разрабатываемого устройства, нужно в выпадающем списке поля выбора этапов моделирования указать вариант **Behavioral**. Затем в списке, открывшемся в окне исходных модулей проекта, необходимо выделить строку с названием используемого тестового файла, поместив на нее курсор и щелкнув левой кнопкой мыши. После этого в окне процессов появляется раздел с названием встроенных средств моделирования ISIM Simulator, который следует открыть, поместив курсор на знак «+», расположенный в этой строке, и щелкнуть левой кнопкой мыши.

Для установки требуемых значений параметров поведенческого моделирования нужно щелчком левой кнопки мыши выделить строку **Simulate Behavioral Model**, после чего нажать кнопку , расположенную на оперативной панели управления **Project Navigator**, или воспользоваться командой **Properties** контекстно-зависимого всплывающего меню, которое выводится при щелчке правой кнопкой мыши. В результате выполненных действий на экране монитора отображается диалоговая панель параметров инициализации средств Xilinx ISIM для выполнения этапа поведенческого моделирования проектируемого устройства, вид которой показан на рис. 9.

Все параметры представлены в виде таблицы, содержащей три колонки. В столбце с заголовком **Property Name** отображается идентификатор параметра, а в соответствующей строке колонки **Value** — его значение. Чтобы изменить значение параметра, следует активизировать соответствующее поле таблицы в столбце **Value**, поместив на него курсор и щелкнув левой кнопкой мыши, а затем воспользоваться кнопкой управления выпадающим списком значений. Если параметр может принимать одно из двух значений — True или False (которые отображаются с помощью индикатора состояния «включено» или «выключено»), то для выбора требуемого варианта достаточно щелкнуть левой кнопкой мыши на поле соответствующего индикатора состояния. Для некоторых параметров требуемое значение можно ввести с помощью клавиатуры непосредственно в поле редактирования, которое расположено в соответствующей ячейке колонки **Value**.

Значение параметра **Use Custom Simulation Command File** определяет необходимость выполнения пользовательского командного файла при запуске средств моделирования Xilinx ISIM. Этот файл содержит команды языка управления Tcl. Если индикатор состояния, предназначенный для установки значения параметра **Use Custom Simulation Command File**, находится в положении «включено», то при активизации системы моделирования будет выполнен командный файл, сформированный разработчиком. Для определения названия исполняемого пользовательского командного файла используется параметр **Custom Simulation Command File**. Идентификатор выполняемого командного файла может быть указан с помощью клавиатуры после активизации поля редактирования значения этого параметра или выбран из списка в стандартной диалоговой панели открытия файла, которая выводится при нажатии кнопки с пиктограммой в виде многоточия (...).

Параметр **Incremental Compilation** позволяет выбрать состав компилируемых файлов при выполнении повторной компиляции модулей исходного описания проектируемого устройства. Если индикатор состояния, определяющий значение этого параметра, установлен в положение «включено», то будут компилироваться только те мо-

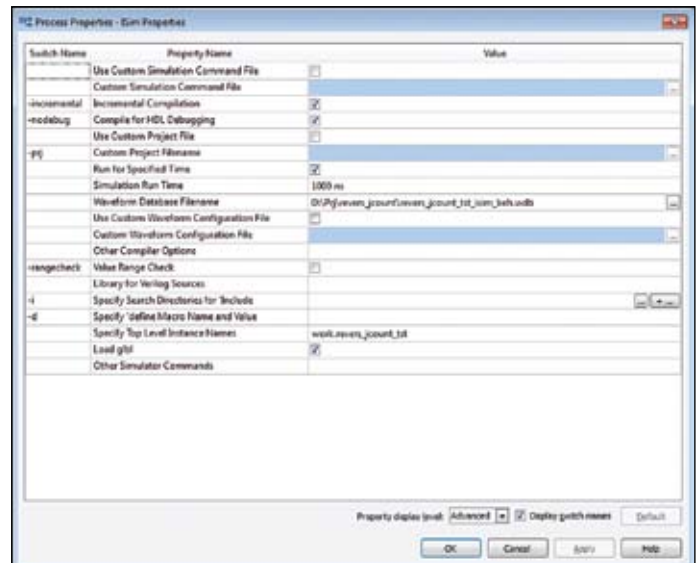


Рис. 9. Вид диалоговой панели параметров средств Xilinx ISIM для поведенческого моделирования

дули исходного описания, в которые вносились изменения после предыдущей компиляции. При выключенном состоянии индикатора **Incremental Compilation** производится компиляция всех исходных модулей HDL-описания, входящих в состав проекта. Таким образом, выбор значения «включено» для рассматриваемого параметра открывает возможность существенного сокращения времени, необходимого для компиляции модулей исходного описания при формировании модели разрабатываемого устройства.

С помощью параметра **Compile for HDL Debugging** определяется необходимость генерации отладочной информации при формировании выходных файлов. При включенном состоянии индикатора, задающего значение этого параметра, процесс моделирования сопровождается формированием отладочной информации, что приводит к увеличению суммарного времени моделирования.

Параметр **Use Custom Project File** предоставляет возможность использования файла проекта PRJ средств моделирования Xilinx ISIM для компиляции HDL-модулей. При установке индикатора состояния этого параметра в положение «включено» становится доступным поле редактирования параметра **Custom Project Filename**, в котором указывается название файла проекта.

Параметр **Run for Specified Time** используется для выбора длительности сеанса моделирования, осуществляемого автоматически при запуске средств Xilinx ISIM. Когда индикатор, определяющий значение этого параметра, находится в состоянии «включено», то длительность моделирования определяется пользователем в виде значения параметра **Simulation Run Time**. В противном случае процесс моделирования выполняется в течение интервала, заданного в модуле тестовых воздействий, или до принудительной остановки этого процесса. По умолчанию для параметра **Simulation Run Time** предлагается значение 1000 нс.

Значение параметра **Waveform Database Filename** определяет идентификатор файла, в который записываются временные диаграммы сигналов, сформированные в процессе моделирования. По умолчанию предлагается название файла временных диаграмм в виде:

идентификатор_тестового_файла_ism_
обозначение_этапа_моделирования.wdb.

В этом идентификаторе для поведенческого моделирования используется краткое условное обозначение beh, а для полного временного — rag. Для изменения предлагаемого названия можно воспользоваться теми же способами, что и при выборе значения параметра **Custom Simulation Command File**.

Параметр *Use Custom Waveform Configuration File* позволяет использовать при моделировании файл конфигурации окна временных диаграмм, идентификатор которого указывается в поле редактирования *Custom Waveform Configuration File*.

С помощью параметра *Other Compiler Options* можно указать дополнительные опции командной строки для управления процессом компиляции HDL-кода.

Параметр *Value Range Check* позволяет разрешить или запретить контроль достоверности назначаемых значений сигналов в течение выполнения процесса моделирования. При включенном состоянии соответствующего индикатора в ходе моделирования осуществляется проверка соответствия устанавливаемых значений сигналов допустимому диапазону, который определяется типами этих сигналов. Если индикатор *Value Range Check* установлен в состоянии «выключено», то контроль достоверности значений сигналов в процессе моделирования не производится.

С помощью параметра *Library for Verilog Sources* указываются дополнительные библиотеки, используемые в модулях исходного описания на языке Verilog.

Значение параметра *Specify Search Directories for Include* определяет каталоги, в которых будет осуществляться поиск файлов, включаемых в состав формируемых Verilog-модулей с помощью директивы *Include*.

Параметр *Specify define Macro Name and Value* позволяет указать макросы, используемые в модулях исходного описания на языке Verilog, и соответствующие значения для этих макросов.

Значение параметра *Load glbl* разрешает или запрещает включение модуля *glbl.v* в состав списка соединений формируемой модели в случае использования языка Verilog.

Параметр *Specify Top Level Instance Names* предоставляет возможность изменения на-

звания объекта верхнего уровня описания моделируемой системы. По умолчанию в качестве значения этого параметра предлагается идентификатор объекта описания, совпадающий с названием сформированного тестового модуля проекта.

С помощью параметра *Other Simulator Command* могут быть заданы дополнительные команды управления для средств моделирования Xilinx ISIM.

Установленные значения параметров процесса поведенческого моделирования вступают в силу только после нажатия клавиши **Apply** или **OK**, расположенной в нижней части диалоговой панели **ISIM Properties** (рис. 9). Запуск поведенческого моделирования проекта осуществляется двойным щелчком левой кнопкой мыши на строке **Simulate Behavioral Model** в окне процессов **Project Navigator** (рис. 8).

Определение значений параметров инициализации средств Xilinx ISIM для функционального моделирования проектируемого устройства

Для установки параметров функциональной верификации разрабатываемого устройства нужно в выпадающем списке поля выбора этапов моделирования указать вариант **Post-Translate**. При этом вкладка **Design** рабочей области **Project Navigator** преобразуется к виду, представленному на рис. 10.

После выделения в окне исходных модулей проекта строки с названием используемого тестового файла, а затем в окне процессов строки **Simulate Post-Translate Model Properties** контекстно-зависимого всплывающего меню открыть диалоговую панель параметров функционального моделирования проектируемого устройства, вид которой приведен на рис. 11. Эта диалоговая панель,

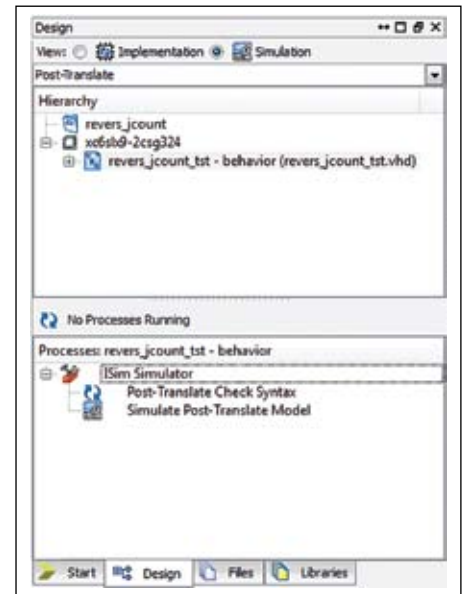


Рис. 10. Вид вкладки Design рабочей области Project Navigator при выполнении функционального моделирования

в отличие от этапа поведенческого моделирования, содержит две страницы — **ISIM Properties** и **Simulation Model Properties**.

Страница **ISIM Properties** включает в себя параметры инициализации встроенных средств моделирования, большинство из которых было представлено в предыдущем разделе. При этом следует учитывать, что одноименные параметры инициализации средств Xilinx ISIM могут принимать различные значения на каждом этапе моделирования. Значения таких параметров, заданные для одного этапа моделирования, не распространяются на другие этапы.

В дополнение к параметрам, назначение которых было рассмотрено выше, на странице **ISIM Properties** присутствует параметр **UUT Instance Name**. Значение этого параме-

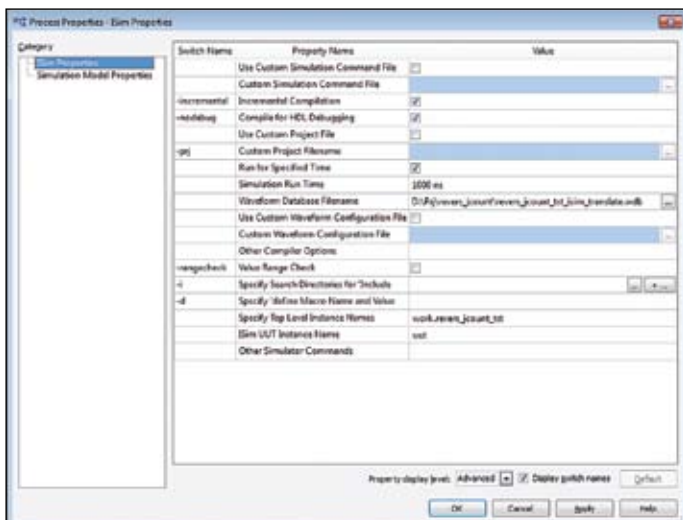


Рис. 11. Вид страницы ISIM Properties диалоговой панели параметров функционального моделирования

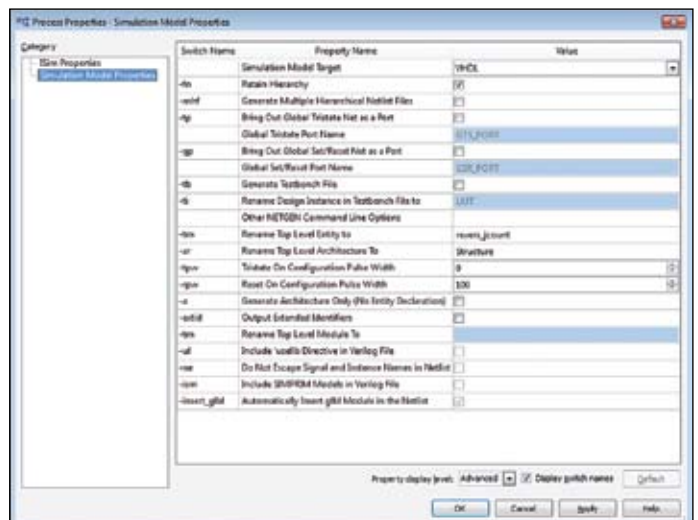


Рис. 12. Вид страницы Simulation Model Properties диалоговой панели параметров функционального моделирования

тра определяет идентификатор экземпляра компонента, представляющего объект верхнего уровня иерархии в тестовом модуле проекта. По умолчанию в качестве названия экземпляра компонента, описывающего испытательный стенд, используется идентификатор *UUT*. Если необходимо изменить идентификатор, предлагаемый по умолчанию, то следует активизировать поле редактирования параметра *UUT Instance Name* и ввести требуемое название с помощью клавиатуры.

На странице *Simulation Model Properties*, вид которой изображен на рис. 12, расположены параметры, предназначенные для управления процедурами формирования функциональной модели разрабатываемого устройства, создаваемой после выполнения этапов синтеза и трансляции проекта.

Значение параметра *Simulation Model Target* определяет язык HDL, используемый для представления описания формируемой модели. Выпадающий список возможных значений этого параметра зависит от выбранных средств синтеза. По умолчанию предлагается язык описания аппаратуры HDL, на котором создаются исходные модули проекта. При использовании средств синтеза XST выпадающий список допустимых значений параметра *Simulation Model Target* содержит два варианта: VHDL и Verilog. В этом случае по умолчанию предлагается значение VHDL.

Значение параметра *Retain Hierarchy* определяет способ представления объектов HDL-описаний в иерархической структуре проекта в генерируемой функциональной модели. Если индикатор, задающий значение этого параметра, установлен в состоянии «включено», то указанные объекты представляются в формируемой модели в виде отдельных иерархических модулей, соответствующих исходному описанию проекта. При выключенном состоянии индикатора *Retain Hierarchy* иерархическая структура объектов, определенная в исходном описании проектируемого устройства, в формируемой модели не сохраняется. По умолчанию этот индикатор состояния находится в положении «включено», при котором сохраняется иерархия объектов в создаваемой функциональной модели.

Параметр *Generate Multiple Hierarchical Netlist Files* позволяет создавать отдельные файлы списков соединений для каждого уровня иерархии проекта, если индикатор состояния предыдущего параметра установлен в положение «включено». Для этого следует индикатор *Generate Multiple Hierarchical Netlist Files* также установить в состояние «включено». По умолчанию указанный индикатор состояния находится в выключенном состоянии, при котором файлы списков соединений для каждого иерархического уровня проекта не формируются.

С помощью параметра *Bring Out Global Tristate Net as a Port* предоставляется возможность включения глобальной цепи установки пользовательских выводов ПЛИС

в высокоимпеданное состояние GTS (Global Tristate) в описание интерфейса моделируемого объекта. При установке индикатора состояния этого параметра в положение «включено» глобальный сигнал GTS преобразуется в формат порта объекта, который представлен в описании верхнего уровня иерархии формируемой модели. По умолчанию индикатор *Bring Out Global Tristate Net as a Port* находится в состоянии «выключено», которое соответствует обычному представлению глобальной цепи переключения выводов кристалла программируемой логики в состояние высокого импеданса.

Значение параметра *Global Tristate Port Name* определяет идентификатор порта, соответствующего глобальной цепи установки выводов ПЛИС в высокоимпеданное состояние. Эта опция становится доступной только при установке индикатора *Bring Out Global Tristate Net as a Port* в состояние «включено». По умолчанию в качестве названия порта установки выводов кристалла в состояние высокого импеданса предлагается идентификатор GTS_PORT.

Параметр *Bring Out Global Set/Reset Net as a Port* используется для включения глобальной цепи сброса/установки кристалла программируемой логики в описание интерфейса моделируемого объекта. Если индикатор состояния, определяющий значение этого параметра, находится в положении «включено», то глобальный сигнал сброса/установки триггерных ресурсов кристалла GSR преобразуется в формат порта объекта, который представлен в описании верхнего уровня иерархии модели. Выключенное состояние индикатора *Bring Out Global Set/Reset Net as a Port*, установленное по умолчанию, соответствует обычному представлению глобальной цепи сброса/установки ПЛИС. При этом соответствующий порт не включается в состав интерфейса объекта верхнего уровня иерархии проекта.

Значение параметра *Global Set/Reset Port Name* позволяет указать название порта, соответствующего глобальной цепи сброса/установки кристалла. По умолчанию название порта совпадает с идентификатором этой цепи — GSR_PORT. Новое значение этого параметра вводится с помощью клавиатуры после активизации соответствующего поля редактирования. Эта опция доступна только в том случае, если индикатор *Bring Out Global Set/Reset Net as a Port* установлен в состояние «включено».

Параметр *Generate Testbench File* управляет процессом автоматической генерации шаблона тестового файла. Создаваемый файл имеет расширение *tvhd* при использовании языка VHDL или *tv*, если применяется язык Verilog. Название автоматически генерируемого файла состоит из идентификатора объекта описания и имени модели (*<entity_name>_<sim_model_name>* для языка VHDL или *<module_name>_<sim_model_name>*

для языка Verilog). По умолчанию индикатор состояния, определяющий значение этого параметра, установлен в положение «выключено», которое запрещает автоматическое формирование шаблона тестового файла.

Параметр *Rename Design Instance in Testbench File to* позволяет переименовать экземпляр компонента, представляющего объект верхнего уровня иерархии в создаваемом шаблоне тестового модуля проекта. По умолчанию в качестве обозначения экземпляра компонента, описывающего испытательный стенд, используется идентификатор *UUT*. Изменить значение этого параметра можно только в том случае, если индикатор *Generate Testbench File* находится в состоянии «включено».

С помощью параметра *Other NETGEN Command Line Options* можно указать дополнительные опции командной строки для программы NETGEN.

Параметр *Rename Top Level Entity to* предоставляет возможность изменения названия для объекта верхнего уровня иерархии создаваемого VHDL-описания модели. По умолчанию если значение этого параметра не определено, то имя объекта верхнего уровня иерархии наследуется из модуля исходного описания проекта. Чтобы указать другой идентификатор, нужно активизировать поле редактирования значения рассматриваемого параметра и воспользоваться клавиатурой.

Параметр *Rename Top Level Architecture to* позволяет изменить название архитектурного тела объекта, соответствующего верхнему уровню иерархии создаваемого VHDL-описания модели. По умолчанию в качестве имени архитектуры объекта верхнего уровня иерархии предлагается идентификатор *STRUCTURE*. Для изменения этого названия следует активизировать поле редактирования и воспользоваться клавиатурой.

Значение параметра *Tristate On Configuration Pulse Width* задает длительность импульса (в наносекундах), необходимого для моделирования компонентов, управляющих инициализацией глобальных цепей установки выводов ПЛИС в состояние высокого импеданса. По умолчанию для этого параметра предлагается нулевое значение.

Значение параметра *Reset On Configuration Pulse Width* определяет длительность импульса, необходимого для моделирования компонентов, управляющих инициализацией глобальных цепей сброса/установки в начальный момент времени (Reset-On-Configuration, ROC). По умолчанию для этого параметра используется значение, равное 100 нс.

С помощью параметра *Generate Architecture Only (No Entry Declaration)* указывается состав формируемого описания функциональной модели. Если соответствующий индикатор состояния находится в положении «включено», то в составе генерируемого описания будет присутствовать только блок определения архитектуры соответствующего

объекта. В этом случае блок декларации этого объекта и его интерфейса не включается в формируемое описание. При выключенном состоянии индикатора **Generate Architecture Only (No Entry Declaration)**, установленном по умолчанию, генерируется полное описание соответствующего объекта.

Параметр **Output Extended Identifiers** предоставляет возможность записи идентификаторов в расширенном формате в формируемом VHDL-описании модели, если во входном файле эти идентификаторы представлены в указанной форме.

Параметр **Rename Top Level Module To** позволяет изменить название модуля верхнего уровня иерархии в описании модели, формируемом на языке Verilog. По умолчанию если значение этого параметра не определено, то имя модуля верхнего уровня иерархии модели совпадает с идентификатором модуля верхнего уровня иерархии из файла исходного описания проектируемого устройства.

Значение параметра **Include uselib Directive in Verilog File** разрешает или запрещает включение директивы **uselib** в состав формируемых файлов. Для встроенных средств моделирования рекомендуется использовать запрещающее значение для этого параметра, предлагаемое по умолчанию.

Параметр **Do Not Escape Signal and Instance Names in Netlist** определяет способ обработки идентификаторов сигналов и компонентов описания проектируемого устройства, в состав которых входят недопустимые символы. Когда индикатор состояния, определяющий значение этого параметра, установлен в положение «включено», производится автоматическое исправление названий сигналов и компонентов, содержащих некорректные символы. В этом случае каждый недопустимый символ в названии автоматически заменяется символом подчеркивания. Если

индикатор **Do Not Escape Signal and Instance Names in Netlist** находится в состоянии «выключено», предлагаемом по умолчанию, то некорректные идентификаторы сигналов и компонентов исключаются средствами генерации функциональной модели.

Значение параметра **Include SIMPRIM Models in Verilog File** разрешает или запрещает включение модулей библиотеки SIMPRIM в состав каждого формируемого файла списка соединений. По умолчанию индикатор **Include SIMPRIM Models in Verilog File** находится в состоянии «выключено», запрещающем использование модулей библиотеки SIMPRIM.

Параметр **Automatically Insert gbl Module in the Netlist** предоставляет возможность автоматического включения модуля **gbl.v** в состав списка соединений формируемой модели.

После установки требуемых значений параметров следует подтвердить их нажатием клавиши **Apply** или **OK**, расположенной в нижней части каждой страницы диалоговой панели (рис. 11, 12). Для выполнения этапа функционального моделирования проектируемого устройства следует дважды щелкнуть левой кнопкой мыши на строке **Simulate Post-Translate Model** в окне процессов **Project Navigator** (рис. 10).

Установка значений параметров запуска средств Xilinx ISIM для частичного временного моделирования разрабатываемого устройства

Для определения параметров частичной временной верификации проектируемого устройства необходимо в выпадающем списке поля выбора этапов моделирования выделить строку **Post-Map**. После этого вкладка **Design** рабочей области **Project Navigator** принимает вид, изображенный на рис. 13.

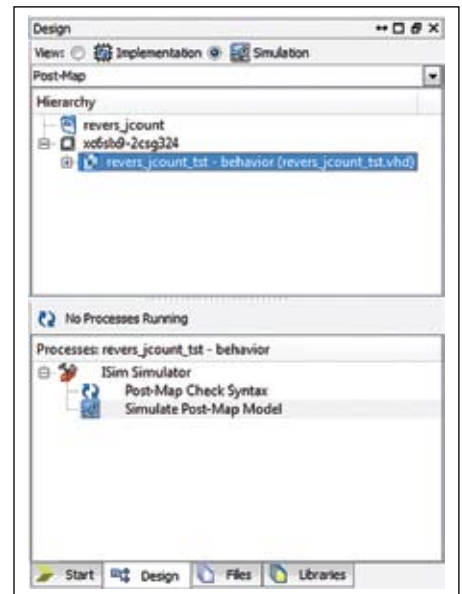


Рис. 13. Вид вкладки Design рабочей области Project Navigator при выполнении частичного временного моделирования

Затем нужно выполнить ту же последовательность операций, что и для предыдущих этапов моделирования, воспользовавшись строкой **Simulate Post-Map Model** в окне процессов, чтобы открыть диалоговую панель параметров частичного временного моделирования разрабатываемого устройства. Эта панель содержит те же две страницы — **ISIM Properties** и **Simulation Model Properties**, что и для этапа функционального моделирования, но в их состав входят дополнительные параметры.

На рис. 14 показан вид страницы **ISIM Properties** диалоговой панели параметров этапа частичного временного моделирования. На этой странице, кроме параметров инициализации средств моделирования

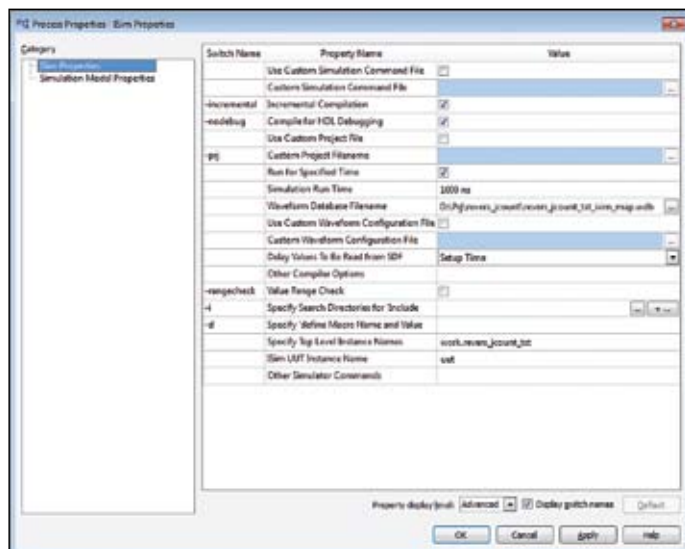


Рис. 14. Вид страницы ISIM Properties диалоговой панели параметров частичного временного моделирования

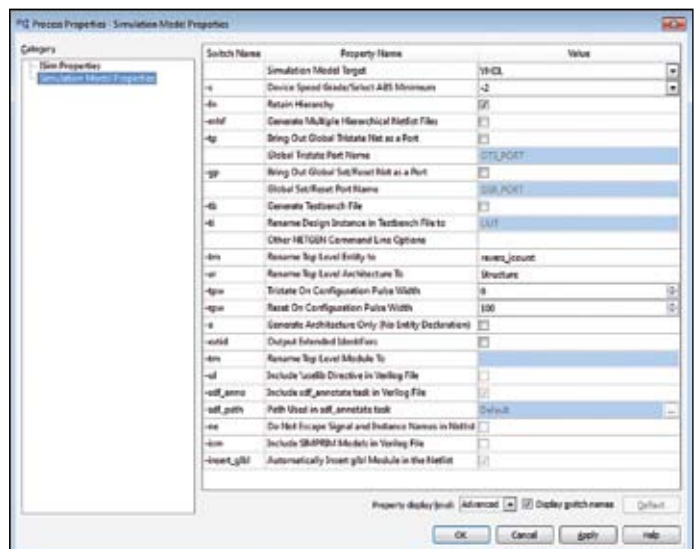


Рис. 15. Вид страницы Simulation Model Properties диалоговой панели параметров частичного временного моделирования

ния Xilinx ISIM, рассмотренных ранее, присутствует параметр *Delay Values to be Read from SDF*. Он предназначен для выбора одного из двух вариантов значений задержек распространения сигналов, представленных в файле SDF, при осуществлении частичного временного моделирования. В выпадающем списке возможных значений этого параметра представлены следующие варианты: *Setup Time* и *Hold Time*. При использовании значения *Setup Time*, установленного по умолчанию, проектируемое устройство моделируется с учетом максимальных величин задержек, указанных в соответствующем файле SDF. Значение *Hold Time* соответствует режиму моделирования при минимальных задержках распространения сигналов в разрабатываемом устройстве.

Вид страницы *Simulation Model Properties*, содержащей параметры частичной временной модели проектируемого устройства, генерируемой после выполнения этапа отображения логического описания на физические ресурсы ПЛИС (MAP), приведен на рис. 15.

Эта страница, кроме представленных выше параметров, включает в себя три новых параметра: *Device Speed Grade/Select ABS Minimum, Include sdf_annotate task in Verilog File* и *Path Used in sdf_annotate task*.

Значение параметра *Device Speed Grade/Select ABS Minimum* указывает класс быстродействия выбранного кристалла программируемой логики, который используется для определения временных характеристик формируемой модели. Выпадающий список допустимых значений этого параметра содержит условные обозначения вариантов класса быстродействия, с которыми выпускается ПЛИС, используемая для реализации разрабатываемого устройства. Вариант *Select ABS Minimum*, представленный в этом списке, соответствует минимальным значениям временных параметров, допустимым для выбранного типа кристалла программируемой логики. Моделирование с различными значениями параметра *Device Speed Grade/Select ABS Minimum* позволяет выбрать оптимальный класс быстродействия ПЛИС, необходимый для выполнения заданных временных ограничений при реализации проектируемого устройства.

Параметр *Include sdf_annotate task in Verilog File* определяет необходимость включения конструкции *\$sdf_annotate* в каждый формируемый файл списка соединений на языке Verilog. По умолчанию индикатор состояния, определяющий значение этого параметра, установлен в положение «выключено», при котором указанная конструкция не записывается в формируемые файлы.


С помощью параметра *Path Used in sdf_annotate task* разработчик может указать раздел, в котором содержатся требуемые файлы SDF. Этот параметр применяется только в тех случаях, когда необходимые файлы SDF расположены вне рабочего каталога проекта.

По умолчанию в качестве значения параметра *Path Used in sdf_annotate task* используется рабочий каталог проекта.

После установки требуемых значений параметров процесса частичного временного моделирования необходимо подтвердить их нажатием клавиши **Apply** или **OK**, которые расположены в нижней части диалоговой панели параметров (рис. 14, 15). Активизация процесса частичного временного моделирования разрабатываемого устройства осуществляется двойным щелчком левой кнопкой мыши на строке **Simulate Post-Map Model** в окне процессов (рис. 13).

Определение значений параметров инициализации средств Xilinx ISIM для полного временного моделирования проектируемого устройства

Чтобы установить требуемые значения параметров полной временной верификации разрабатываемого устройства, нужно в выпадающем списке поля выбора этапов моделирования указать вариант **Post-Route**. При этом вкладка **Design** рабочей области приобретает вид, представленный на рис. 16.

Для определения параметров полного временного моделирования необходимо в окне исходных модулей проекта выделить строку с названием используемого тестового файла, а в окне процессов — строку **Simulate Post-Place & Route Model**, после чего воспользоваться кнопкой  или командой **Properties** из контекстно-зависимого всплывающего меню. В результате выполненных действий на экран выводится диалоговая панель параметров полной временной верификации разрабатываемого устройства, которая, в отличие от предыдущих этапов моделирования, содержит три страницы —

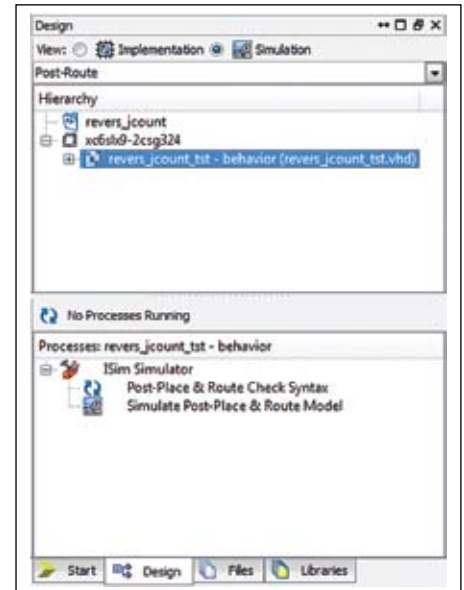


Рис. 16. Вид вкладки Design рабочей области Project Navigator при выполнении полного временного моделирования

ISIM Properties, Simulation Model Properties и *XPower Analyzer Properties*.

Страница *ISIM Properties*, вид которой приведен на рис. 17, кроме параметров, рассмотренных в предыдущих разделах, включает два дополнительных — *Generate SAIF File for Power Optimization/Estimation* и *SAIF File Name*.

Параметр *Generate SAIF File for Power Optimization/Estimation* предоставляет возможность генерации средствами моделирования Xilinx ISIM файла в формате Synopsys Activity Interchange Format (SAIF) для оценки и оптимизации потребляемой мощности. По умолчанию индикатор состояния, указывающий значение этого параметра, находит-

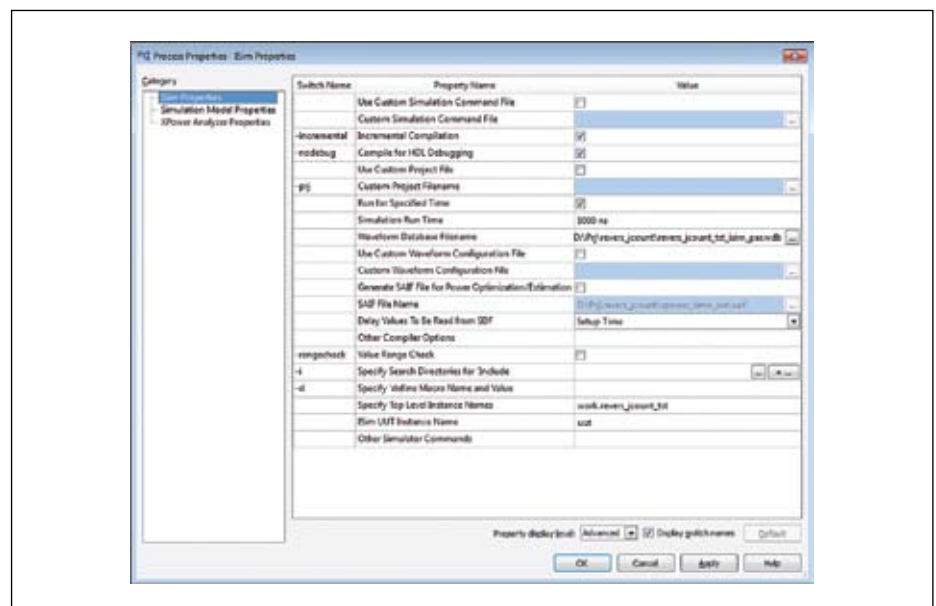


Рис. 17. Вид страницы ISIM Properties диалоговой панели параметров полного временного моделирования

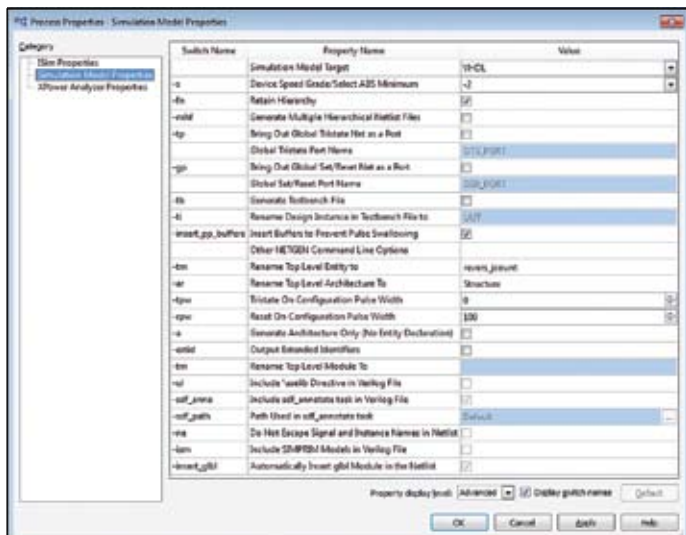


Рис. 18. Вид страницы Simulation Model Properties диалоговой панели параметров полного временного моделирования

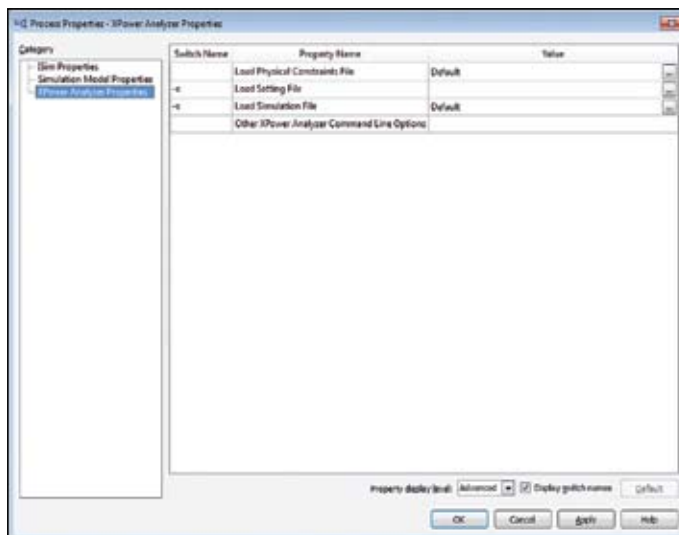


Рис. 19. Вид страницы XPower Analyzer Properties диалоговой панели параметров полного временного моделирования

ся в положении «выключено», при котором указанный файл не создается.

Значение параметра *SAIF File Name* определяет идентификатор файла SAIF, создаваемого средствами моделирования Xilinx ISIM, если индикатор *Generate SAIF File for Power Optimization/Estimation* установлен в состоянии «включено». По умолчанию для формируемого файла предлагается идентификатор *xpower_time_sim.saif*.

На странице *Simulation Model Properties*, содержащей параметры генерации полной временной модели разрабатываемого устройства, создаваемой после выполнения этапа размещения и трассировки проекта в кристалле программируемой логики (Post-Place and Route), по сравнению с частичным временным моделированием присутствует дополнительный параметр *Insert Buffers to Prevent Pulse Swallowing* (рис. 18). Он позволяет добавлять в формируемый список соединений модели разрабатываемого устройства буферные элементы, предотвращающие поглощение импульсов на входах компонентов, длительность которых меньше входной задержки. По умолчанию индикатор состояния параметра *Simulation Model Properties*

находится в положении «включено», разрешающем включение в состав списка соединений дополнительных буферных элементов.

Страница *XPower Analyzer Properties*, вид которой представлен на рис. 19, содержит параметры полной временной модели, управляющие формированием информационных данных для средств анализа потребляемой мощности XPower Analyzer.

Значение параметра *Load Physical Constraints File* указывает идентификатор файла физических ограничений Physical Constraints File (PCF), содержащего информацию о значениях температуры, напряжения питания и временных параметров, которая необходима для более достоверной оценки потребляемой мощности.

Параметр *Load Setting File* предоставляет возможность использования файла установок (с расширением *.хра*), который может формироваться средствами XPower Estimator или XPower Analyzer.

Значение параметра *Load Simulation File* определяет файл, формируемый средствами моделирования Xilinx ISIM, который будет использоваться для оценки потребляемой мощности.

С помощью параметра *Other XPower Analyzer Command Line Options* можно задать дополнительные опции командной строки для средств XPower Analyzer.

Процедура установки требуемых значений параметров полного временного моделирования завершается нажатием кнопки **Apply** или **OK**, которые находятся в нижней части диалоговой панели (рис. 17–19). Для запуска процесса полного временного моделирования проектируемого устройства достаточно дважды щелкнуть левой кнопкой мыши на строке **Simulate Post-Place & Route Model** в окне процессов (рис. 16).

Окончание следует

Литература

1. Зотов В. Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPack ISE. М.: Горячая линия – Телеком, 2003.
2. Зотов В. ModelSim — система HDL-моделирования цифровых устройств // Компоненты и технологии. 2002. № 6.
3. Кузелин М. О., Кнышев Д. А., Зотов В. Ю. Современные семейства ПЛИС фирмы Xilinx. Справочное пособие. М.: Горячая линия – Телеком, 2004.
4. XST User Guide for Virtex-4, Virtex-5, Spartan-3, and Newer CPLD Devices. Xilinx, 2010.
5. XST User Guide for Virtex-6, Spartan-6, and 7 Series Devices. Xilinx, 2012.
6. Synthesis and Simulation Design Guide. Xilinx, 2012.
7. Зотов В. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx. М.: Горячая линия – Телеком, 2006.
8. Зотов В. Разработка VHDL-описаний цифровых устройств, проектируемых на основе ПЛИС фирмы Xilinx, с использованием шаблонов САПР ISE Design Suite // Компоненты и технологии. 2010. № 2–12. 2011. № 1–12. 2012. № 1–12.