

Измеритель частоты цифровых сигналов, выполненный на основе микропроцессорного ядра семейства PicoBlaze и реализуемый на базе инструментального комплекта Spartan-3E Starter Kit фирмы Xilinx

Предлагаемая вниманию читателей статья продолжает серию публикаций, начатую в [1]. Она знакомит с проектами встраиваемых микропроцессорных систем различного назначения, выполненными на основе конфигурируемых микропроцессорных ядер фирмы Xilinx® семейств PicoBlaze™ [2–7] и MicroBlaze™ [2, 8–10]. В настоящей статье рассматривается проект измерителя частоты цифровых сигналов, в структуру которого входит 8-разрядное микропроцессорное ядро семейства PicoBlaze, предназначенное для применения в кристаллах ПЛИС серий Spartan-3, Spartan-3E, Virtex-II, Virtex-II-Pro и Virtex-4. Этот проект наглядно демонстрирует возможность использования конфигурируемых ядер указанного семейства при разработке измерительных систем. Разработчики могут использовать рассматриваемое устройство или его отдельные модули исходного описания в процессе проектирования собственных систем и комплексов. Для аппаратной реализации проекта измерителя частоты цифровых сигналов используется инструментальный модуль Xilinx Spartan-3E Starter Board™, входящий в состав стартового комплекта Spartan-3E Starter Kit, который был представлен в [11].

Валерий ЗОТОВ
walerry@km.ru

Структура архива проекта измерителя частоты цифровых сигналов, предназначенного для реализации на базе инструментального комплекта Spartan-3E Starter Kit

Все модули исходного описания проекта измерителя частоты цифровых сигналов, а также файлы, необходимые для его загрузки в кристалл, упакованы в архив *s3esk_frequency_counter.zip*, который можно свободно скопировать, открыв Web-страницу http://www.xilinx.com/products/boards/s3estarter/reference_designs.htm. В состав этого архива входят следующие файлы:

- модуль исходного описания генератора тестового сигнала с частотой 118 МГц на языке VHDL *ring_osc.vhd*;
- файл временных и топологических ограничений проекта измерителя частоты цифровых сигналов *frequency_counter.ucf*, соответствующий схеме включения ПЛИС XC3S500E в составе инструментального модуля Xilinx Spartan-3E Starter Board;
- исходный текст микропроцессорной программы и процедуры обработки прерывания на языке ассемблера *fc_ctrl.psm*;
- файл описания содержимого программной памяти измерителя частоты цифровых сигналов на языке VHDL *fc_ctrl.vhd*, полученный в результате трансляции исходного текста микропроцессорной программы и процедуры обработки прерывания на языке ассемблера;
- файл конфигурационной последовательности *frequency_counter.bit*, реализующей измеритель частоты цифровых сигналов, предназначенный для непосредственной загрузки в кристалл;
- командный файл *install_frequency_counter.bat*, используемый для непосредственной загрузки конфигурационной последовательности измерителя частоты цифровых сигналов в ПЛИС XC3S500E инструментального модуля Xilinx Spartan-3E Starter Board, выполняемой автоматически в пакетном режиме;
- файл документации *frequency_counter_v100.pdf*, содержащий описание проекта измерителя частоты цифровых сигналов, выполненное в формате PDF.

В архив проекта не включен файл, содержащий исходное VHDL-описание исполнительного модуля микропроцессорного ядра семейства PicoBlaze. Поэтому при внесении изменений в модули исходного описания измерителя частоты цифровых сигналов или создании новых проектов, выполняемых на основе рассматриваемого устройства, для выполнения процессов синтеза, размещения

и трассировки проекта в кристалле следует добавить в состав этого проекта файл *kcpsm3.vhd*. Этот файл входит в состав архива исходных модулей описания микропроцессорного ядра PicoBlaze *KCPSM3.zip*, которое предназначено для применения в проектах, реализуемых на основе ПЛИС семейств Spartan-3, Spartan-3E, Virtex-II, Virtex-II PRO и Virtex-4 [7]. Для получения данного архива необходимо вначале выполнить процедуру бесплатной регистрации на Web-сервере фирмы Xilinx (<http://www.xilinx.com/>), в ходе которой пользователю присваивается введенный им идентификационный код (UserID) и пароль (Password). Затем следует открыть Web-страницу http://www.xilinx.com/ipcenter/processor_central/picoblaze/index.htm и, указав параметры идентификации пользователя, скопировать файл архива, соответствующий указанной версии микропроцессорного ядра PicoBlaze.

Принцип работы и функциональная схема измерителя частоты цифровых сигналов, реализуемого на базе инструментального модуля Xilinx Spartan-3E Starter Board

Функционирование рассматриваемого измерителя частоты цифровых сигналов основано на вычислении количества периодов изменения исследуемого сигнала за определенный промежуток времени. В качестве ин-

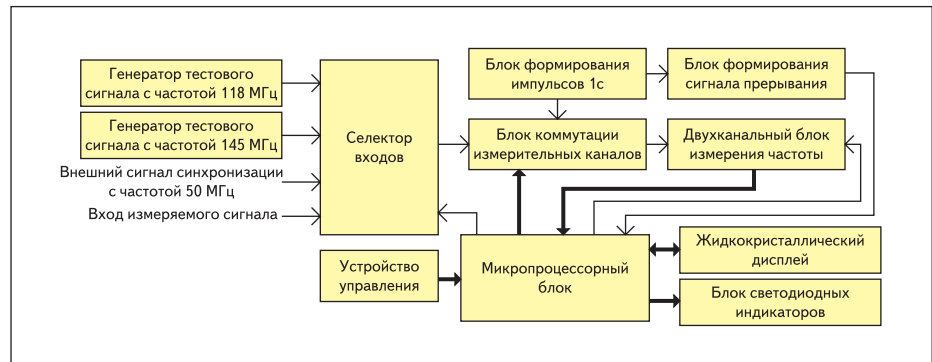


Рис. 1. Структурная схема измерителя частоты цифровых сигналов, реализуемого на базе инструментального модуля Xilinx Spartan-3E Starter Board

тервала измерения выбран промежуток времени, длительность которого составляет 1 с. Номинальный диапазон измерения частоты входного сигнала составляет от 1 Гц до 200 МГц. Верхняя граница этого диапазона определяется максимально допустимым значением частоты сигналов проекта, реализуемого на базе выбранной ПЛИС семейства Spartan-3E.

Структурная схема измерителя частоты цифровых сигналов представлена на рис. 1. В состав аппаратной платформы рассматриваемого устройства входят следующие блоки:

- селектор входов;
- блок формирования импульсов длительностью 1 с;
- блок коммутации измерительных каналов;
- двухканальный блок измерения частоты;
- блок формирования сигнала прерывания;
- генератор тестового сигнала с частотой 145 МГц;
- генератор тестового сигнала с частотой 118 МГц;
- устройство управления;
- микропроцессорный блок;
- жидкокристаллический дисплей;
- блок светодиодных индикаторов.

- блок формирования сигнала прерывания;
- генератор тестового сигнала с частотой 145 МГц;
- генератор тестового сигнала с частотой 118 МГц;
- устройство управления;
- микропроцессорный блок;
- жидкокристаллический дисплей;
- блок светодиодных индикаторов.

Селектор входов предназначен для выбора одного из четырех источников измеряемого сигнала: генератора тестового сигнала с частотой 145 МГц, генератора тестового сигнала с частотой 118 МГц, внешнего кварцевого генератора тактового сигнала с частотой 50 МГц или сигнала, поступающего через высокочастотный разъем типа SMA. Управление рабо-

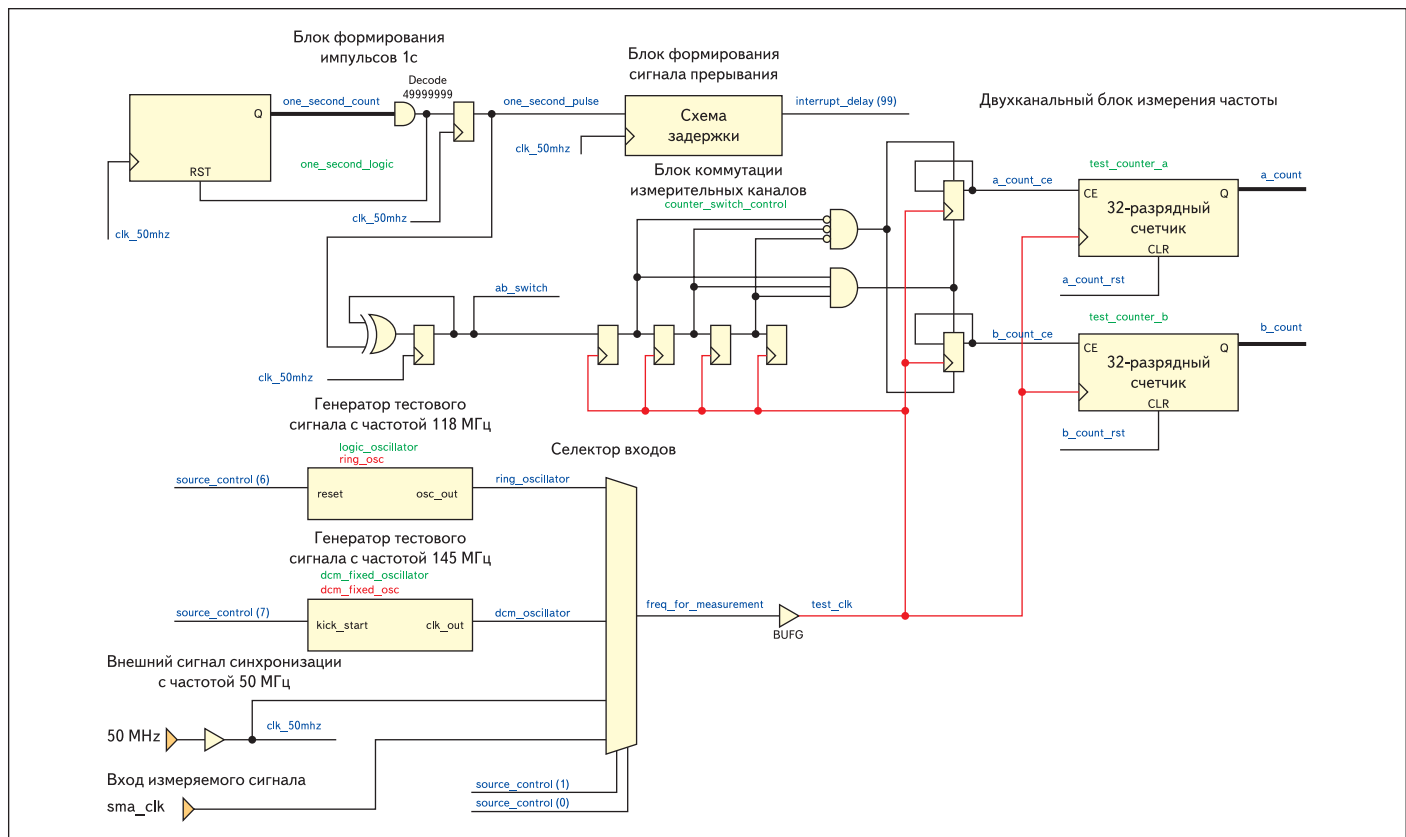


Рис. 2. Функциональная схема входных и измерительных узлов цифрового частотомера, реализуемого на базе инструментального модуля Xilinx Spartan-3E Starter Board

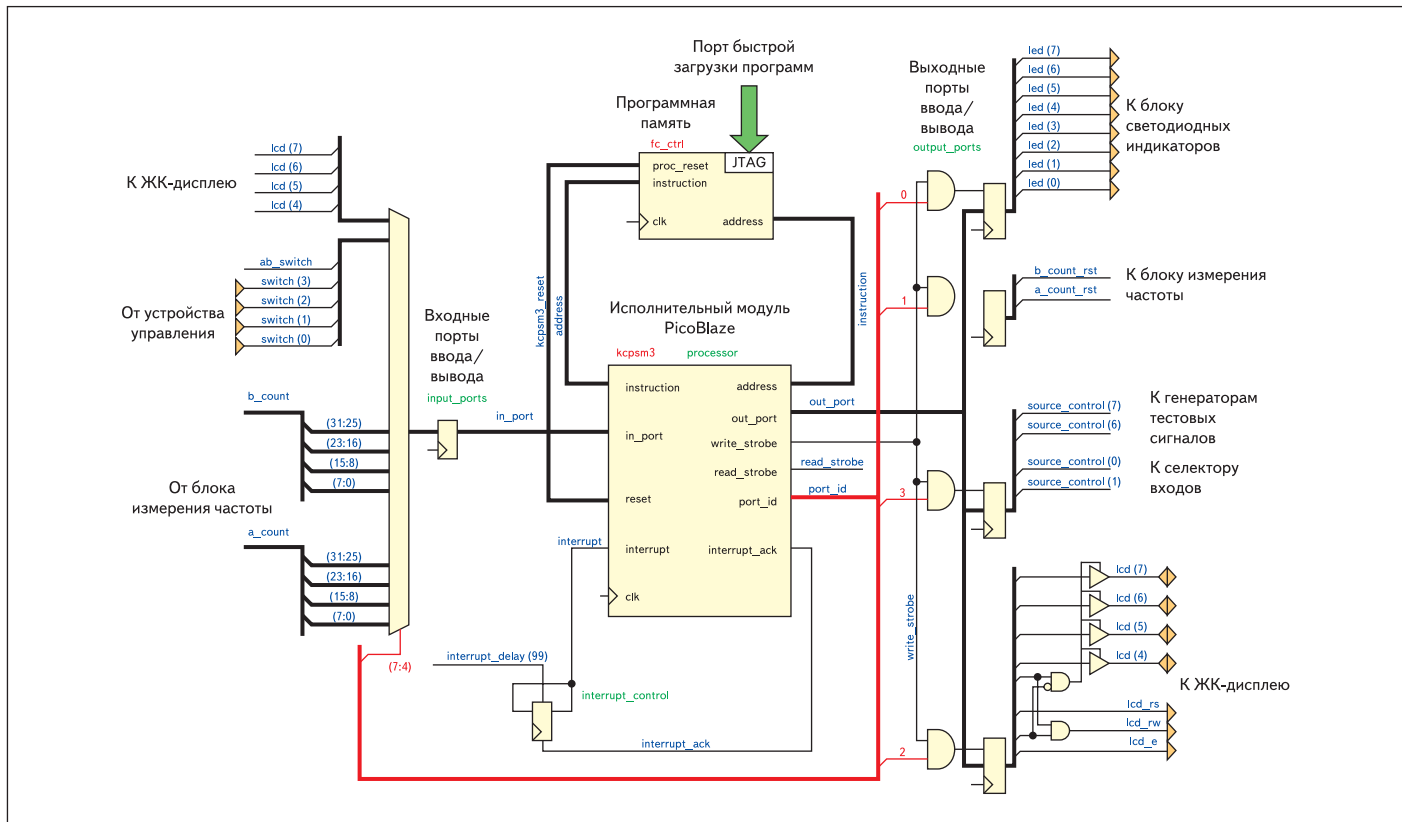


Рис. 3. Функциональная схема микропроцессорного блока измерителя частоты цифровых сигналов

той селектора входов осуществляется с помощью сигналов, которые формируются в микропроцессорном блоке на основании анализа данных, считываемых из устройства управления.

Более детальная функциональная схема, поясняющая структуру входных и измерительных узлов цифрового частотомера, реализуемого на базе инструментального модуля Xilinx Spartan-3E Starter Board, приведена на рис. 2. Блок формирования импульсов длительностью 1 с выполнен на основе 26-разрядного счетчика, на тактовый вход которого подается основной сигнал синхронизации с частотой 50 МГц. В качестве источника основного внешнего (по отношению к ПЛИС) сигнала синхронизации используется кварцевый генератор с той же частотой, установленный на отладочной плате Xilinx Spartan-3E Starter Board. Формируемые импульсы длительностью 1 с определяют интервал измерения частоты.

Блок измерения частоты включает в себя два идентичных, поочередно переключаемых канала. Такое решение обеспечивает непрерывность измерительного процесса. В то время когда один канал работает в режиме измерения частоты, в другом канале производится считывание результатов измерения, полученных на предыдущем интервале. Каждый канал блока измерения частоты представляет собой 32-разрядный счетчик. Управление режимами работы этих счетчиков осуществляется с помощью сиг-

налов, вырабатываемых блоком коммутации измерительных каналов (рис. 2). Разрядность счетчиков выбрана в соответствии с верхней границей диапазона измерения частоты входного сигнала.

Устройство управления предназначено для выбора одного из четырех входов измерителя частоты цифровых сигналов и переключения режимов его работы. Основными элементами этого устройства являются четыре ползунковых переключателя, которые входят в состав инструментального модуля Xilinx Spartan-3E Starter Board. Когда все переключатели находятся в выключенном состоянии или более одного переключателя установлено в положение «Включено», измеритель частоты функционирует в режиме ожидания. Выбор требуемого входа измерителя осуществляется установкой только одного (соответствующего) переключателя в положение «Включено». При этом автоматически включается режим измерения частоты сигнала, поступающего на выбранный вход.

Микропроцессорный блок измерителя частоты цифровых сигналов предназначен для выполнения следующих функций:

- декодирование сигналов, поступающих от устройства управления;
- вычисление значения частоты измеряемого сигнала;
- формирование совокупности информационных и управляющих сигналов, соответствующих интерфейсу жидкокристаллического дисплея, которые необходимы для

отображения текстовой информации и результатов измерения частоты;

- управление работой блока светодиодных индикаторов.

Функциональная схема микропроцессорного блока измерителя частоты цифровых сигналов показана на рис. 3. Основными элементами этой схемы являются:

- исполнительный модуль микропроцессорного ядра семейства PicoBlaze, предназначенного для реализации на базе ПЛИС серий Spartan-3, Spartan-3E, Virtex-II, Virtex-II-Pro и Virtex-4;
- память микропрограмм;
- входные и выходные порты ввода/вывода.

Блок формирования сигнала прерывания представляет собой схему задержки, выполненную на базе сдвиговых регистров SRL16E ПЛИС семейства Spartan-3E. На вход этой схемы задержки поступают импульсы длительностью 1 с.

Блок светодиодных индикаторов используется для визуального контроля текущего режима работы измерителя частоты цифровых сигналов. В его состав входят восемь индикаторов, установленных на плате инструментального модуля Xilinx Spartan-3E Starter Board. В режиме измерений происходит периодическое переключение состояния светодиодных индикаторов.

Жидкокристаллический дисплей предназначен для отображения информации о входных сигналах рассматриваемого частотомера и значений частоты измеряемого сигнала.

В режиме ожидания ЖК-дисплей содержит меню выбора входов. При этом в верхней строке дисплея отображаются условные обозначения входных сигналов измерителя частоты, а в нижней — номера соответствующих переключателей устройства управления. В режиме измерения в верхней строке выводится значение частоты сигнала, поступающего с выбранного входа, а в нижней — условное обозначение этого входа.

Генератор тестового сигнала с частотой 145 МГц выполнен на основе цифрового модуля управления синхронизацией Digital Clock Manager (DCM) ПЛИС семейства Spartan-3E. Для реализации генератора тестового сигнала с частотой 118 МГц применяется кольцевая схема, которая будет представлена при рассмотрении VHDL-описания этого блока.

Для расширения возможностей рассматриваемого измерителя частоты цифровых сигналов в состав проекта можно включить модуль универсального асинхронного приемопередатчика UART (Universal Asynchronous Receiver-Transmitter), который входит в архив исходных файлов описания микропроцессорного ядра PicoBlaze *KCPSM3.zip*. Этот модуль можно использовать для передачи результатов измерений через последовательный порт (СОМ-порт) в персональный компьютер.

В последующих разделах рассматриваются описания основных функциональных блоков измерителя частоты цифровых сигналов на языке VHDL, которые реализуются на основе ресурсов ПЛИС семейства Spartan-3E.

VHDL-описание основных блоков аппаратной части измерителя частоты цифровых сигналов

Описание аппаратной части измерителя частоты цифровых сигналов, выполненное на языке VHDL, имеет иерархическую структуру. В исходном модуле верхнего уровня иерархии проекта описывается объект *frequency_counter*, который представляет все устройство в целом. В состав описания этого объекта в качестве компонентов входят объекты нижнего уровня иерархии *dcm_fixed_osc* и *ring_osc*, представляющие генераторы тестовых сигналов. В настоящем разделе рассматривается описание объекта *frequency_counter*, структура которого соответствует типовому строению исходного модуля, выполненного с использованием языка VHDL [12–14]. Описания объектов *dcm_fixed_osc* и *ring_osc* приводятся в последующих разделах.

Первая часть VHDL-описания объекта *frequency_counter* содержит ссылки на используемые стандартные логические библиотеки и пакеты этих библиотек. Кроме того, в этой части есть ссылка на используемую библиотеку Unisim Library фирмы Xilinx и пакет *vcomponents* этой библиотеки, в которых определены примитивы, применяемые в качестве компонентов в описании объекта *frequency_counter*. Текст первой час-

ти VHDL-описания верхнего уровня иерархии проекта выглядит следующим образом:

```
-----
-- Library declarations
--
-- Standard IEEE libraries
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--
-- The Unisim Library is used to define Xilinx primitives. It is also used
during
-- simulation. The source can be viewed at XILINX%\vhdl\src
\unisim\unisim_VCOMP.vhd
library unisim;
use unisim.vcomponents.all;
-----
```

Во второй части объявляется объект *frequency_counter*, представляющий измеритель частоты цифровых сигналов, и описывается его внешний интерфейс. VHDL-код, содержащийся в этой части описания, имеет следующий вид:

```
-----
entity frequency_counter is
Port (
    led : out std_logic_vector(7 downto 0);
    sw : in std_logic_vector(3 downto 0);
    strataflash_oe : out std_logic;
    strataflash_ce : out std_logic;
    strataflash_we : out std_logic;
    lcd_d : inout std_logic_vector(7 downto 4);
    lcd_rs : out std_logic;
    lcd_rw : out std_logic;
    lcd_e : out std_logic;
    sma_clk : in std_logic;
    clk_50mhz : in std_logic
);
end frequency_counter;
-----
```

В представленной конструкции применяется следующая система условных обозначений входных, выходных и двунаправленных портов объекта *frequency_counter*. Векторы *led* и *sw* представляют соответственно выходы, предназначенные для сопряжения с блоком светодиодных индикаторов, и входы, к которым подключаются ползунковые переключатели устройства управления. Идентификаторы *strataflash_oe*, *strataflash_ce* и *strataflash_we* описывают выходы сигналов управления интерфейса параллельной Flash-памяти серии StrataFlash® Memory фирмы Intel®, которая расположена на плате инструментального модуля Xilinx Spartan-3E Starter Board. Идентификаторы *clk_50mhz* и *sma_clk* соответствуют входу внешнего тактового сигнала с частотой 50 МГц и входу измеряемого сигнала, сопряженному с высокочастотным разъемом типа SMA. Вектор *lcd_d* представляет двунаправленную шину данных интерфейса ЖК-дисплея, а идентификаторы *lcd_rs*, *lcd_rw*, *lcd_e* — сигналы управления этого интерфейса.

Третья часть VHDL-описания верхнего уровня иерархии проекта измерителя частоты содержит определение архитектуры объекта *frequency_counter*. В начале описания архитектурного тела представлены выражения декларации компонентов исполнительного модуля микропроцессорного ядра *kcpsm3*, программной памяти *fc_ctrl* и двух генерато-

ров тестовых сигналов *dcm_fixed_osc* и *ring_osc*. Содержание блока декларации компонентов, которые используются в составе VHDL-описания объекта *frequency_counter*, выглядит следующим образом:

```
-----
-- Start of test architecture
architecture Behavioral of frequency_counter is
-----
--
-- Декларация исполнительного модуля KCPSM3
--
component kcpsm3
Port (
    address : out std_logic_vector(9 downto 0);
    instruction : in std_logic_vector(17 downto 0);
    port_id : out std_logic_vector(7 downto 0);
    write_strobe : out std_logic;
    out_port : out std_logic_vector(7 downto 0);
    read_strobe : out std_logic;
    in_port : in std_logic_vector(7 downto 0);
    interrupt : in std_logic;
    interrupt_ack : out std_logic;
    reset : in std_logic;
    clk : in std_logic
);
end component;
--
-- Декларация модуля программной памяти
--
component fc_ctrl
Port (
    address : in std_logic_vector(9 downto 0);
    instruction : out std_logic_vector(17 downto 0);
    proc_reset : out std_logic; --JTAG Loader version
    clk : in std_logic
);
end component;
--
-- Декларация генератора тестового сигнала, реализуемого на основе цифрового
-- модуля управления синхронизацией Digital Clock Manager (DCM)
--
component dcm_fixed_osc
port(
    clk_out : out std_logic;
    kick_start : in std_logic
);
end component;
--
-- Декларация генератора тестового сигнала, построенного по кольцевой схеме
--
component ring_osc
port(
    osc_out : out std_logic;
    reset : in std_logic
);
end component;
-----
```

Следующий блок VHDL-кода содержит выражения декларации внутренних сигналов, которые используются в описании архитектуры измерителя частоты. В начале этого блока приведены объявления сигналов, которые относятся к селектору входов рассматриваемого измерителя частоты. Далее следует совокупность выражений, которые предназначены для декларации сигналов, управляющих процессом измерения частоты. Затем осуществляется декларация сигналов, используемых в схеме формирования импульсов длительностью 1 с. После этого приведена группа выражений, которые содержат объявления сигналов, используемых для сопряжения исполнительного модуля микропроцессорного ядра PicoBlaze с программной памятью и портами ввода/вывода. Заключительную часть блока декларации внутренних сигналов образует последовательность выражений, содержащих объявления сигналов, которые

описывают интерфейс управления ЖК-дисплея. Текст блока декларации внутренних сигналов в описании измерителя частоты выглядит следующим образом:

```

-----
-- Сигналы селектора входов измерителя частоты
--
signal source_control      : std_logic_vector(7 downto 0);
signal dcm_oscillator      : std_logic;
signal ring_oscillator     : std_logic;
signal freq_for_measurement : std_logic;
signal test_clk            : std_logic;
--
-- Сигналы управления процессом измерения частоты
--
signal ab_switch_delay     : std_logic_vector (3 downto 0);
signal a_count_ce         : std_logic;
signal a_count_rst        : std_logic;
signal a_count             : std_logic_vector (31 downto 0);
signal b_count_ce         : std_logic;
signal b_count_rst        : std_logic;
signal b_count             : std_logic_vector (31 downto 0);
--
-- Сигналы, используемые в схеме формирования импульсов длительностью 1 с
--
signal one_second_count    : integer range 0 to 49999999 :=0;
signal one_second_pulse   : std_logic;
signal ab_switch           : std_logic;
signal interrupt_delay     : std_logic_vector (99 downto 0);
--
-- Сигналы сопряжения исполнительного модуля KCPSM3
-- с программной памятью и портами ввода/вывода
--
signal address             : std_logic_vector(9 downto 0);
signal instruction         : std_logic_vector(17 downto 0);
signal port_id            : std_logic_vector(7 downto 0);
signal out_port           : std_logic_vector(7 downto 0);
signal in_port            : std_logic_vector(7 downto 0);
signal write_strobe       : std_logic;
signal read_strobe        : std_logic;
signal interrupt          : std_logic;
signal interrupt_ack      : std_logic;
signal reset              : std_logic;
--
-- Сигналы интерфейса управления ЖК-дисплеем
--
-- Tri-state output requires internal signals
-- 'lcd_drive' is used to differentiate between LCD and StrataFLASH
communications
-- which share the same data bits.
--
signal lcd_rw_control     : std_logic;
signal lcd_output_data    : std_logic_vector(7 downto 4);
signal lcd_drive          : std_logic;
-----

```

Вслед за блоком декларации внутренних сигналов, после ключевого слова `begin`, начинается собственно определение архитектуры рассматриваемого измерителя частоты цифровых сигналов. В самом начале этого описания представлена группа операторов, которые определяют состояния управляющих сигналов интерфейса параллельной Flash-памяти серии StrataFlash Memory фирмы Intel, установленной на плате инструментального модуля Xilinx Spartan-3E Starter Board. Данная Flash-память, не задействованная в измерителе частоты цифровых сигналов, и ЖК-дисплей, входящий в состав рассматриваемого устройства, используют общие проводники шины данных для сопряжения с ПЛИС XC3S500E. Во избежание конфликтов сигналов, поступающих на эти общие линии шины данных от различных источников, информационные входы/выходы микросхемы Flash-памяти, не используемой в данном проекте, переводятся в состояние «Выключено». Для этого на управляющие входы указанной микросхемы подаются соответствующие значения сигналов, которые переключают вхо-

ды/выходы данных Flash-памяти в «третье» состояние (состояние высокого импеданса):

```

-----
-- Start of circuit description
--
begin
-----
-- Блокировка входов и выходов неиспользуемых компонентов
--
--
-- Отключение информационных входов/выходов Flash-памяти
--
strataflash_oe <= '1';
strataflash_ce <= '1';
strataflash_we <= '1';
-----

```

Далее следует совокупность выражений, описывающих структуру входного узла измерителя частоты цифровых сигналов. В эту совокупность входят операторы, используемые для создания экземпляров компонентов генераторов тестового сигнала различного типа *dcm_fixed_osc* и *ring_osc*, а также глобального буферного элемента *BUFG*. Описание селектора входов выполнено с помощью параллельного оператора условного присваивания значения сигнала:

```

-----
-- Описание структуры входного узла измерителя частоты цифровых сигналов
--
-- Создание экземпляра генератора тестового сигнала, реализуемого на основе
-- цифрового модуля управления синхронизацией DCM
--
dcm_fixed_oscillator: dcm_fixed_osc
port map (
    clk_out => dcm_oscillator,
    kick_start => source_control(7)
);
--
-- Создание экземпляра генератора тестового сигнала,
-- выполненного по кольцевой схеме
--
logic_oscillator: ring_osc
port map (
    osc_out => ring_oscillator,
    reset => source_control(6)
);
--
-- Описание селектора входов
--
freq_for_measurement <= sma_clk when (source_control(1 downto 0) = 0)
else clk_50mhz when (source_control(1 downto 0) = 01)
else dcm_oscillator when (source_control(1 downto 0) = 10)
else ring_oscillator;
--
-- Создание экземпляра глобального входного буферного элемента
--
buffer_clk_in: BUFG
port map(
    O => test_clk,
    I => freq_for_measurement
);
-----

```

После выражений, определяющих структуру входного узла рассматриваемого устройства, представлена группа операторов, образующих поведенческое описание измерительного узла. В состав этого блока VHDL-кода входят три оператора процесса. С помощью процесса *counter_switch_control* описываются состояния управляющих сигналов, используемых для организации измерения частоты входных сигналов. Процессы *test_counter_a* и *test_counter_b* представляют собой поведенческое описание двух 32-разрядных счетчиков, используемых в соответствующих каналах измерительного узла:

```

-----
Поведенческое описание измерительного узла
-----
counter_switch_control: process(test_clk)
begin
    if test_clk'event and test_clk='1' then
        --
        -- Four stage register to ensure synchhronisation of control between
        clock domains
        ab_switch_delay <= ab_switch_delay(2 downto 0) & ab_switch;
        --
        -- Separate enables form switch between counters when control is consistently
        high or low.
        case ab_switch_delay(3 downto 1) is
            when «000» => a_count_ce <= '1';
                b_count_ce <= '0';
            when «111» => a_count_ce <= '0';
                b_count_ce <= '1';
            when others => a_count_ce <= a_count_ce;
                b_count_ce <= b_count_ce;
        end case;
    end if;
end process counter_switch_control;
--
-- 32-bit counters
test_counter_a: process(test_clk, a_count_rst)
begin
    if a_count_rst='1' then
        a_count <= X«00000000»;
    elsif test_clk'event and test_clk='1' then
        f_a_count_ce='1' then
            a_count <= a_count + 1;
        else
            a_count <= a_count;
        end if;
    end if;
end process test_counter_a;
--
test_counter_b: process(test_clk, b_count_rst)
begin
    if b_count_rst='1' then
        b_count <= X«00000000»;
    elsif test_clk'event and test_clk='1' then
        if b_count_ce='1' then
            b_count <= b_count + 1;
        else
            b_count <= b_count;
        end if;
    end if;
end process test_counter_b;
-----

```

В следующем блоке VHDL-кода измерителя частоты цифровых сигналов представлено описание схемы, формирующей импульсы длительностью 1 с. В этот же блок входят операторы, определяющие состояния сигналов управления прерываниями:

```

-----
Описание схемы, формирующей импульсы длительностью 1 с
и сигналы прерывания
-----
one_second_logic: process(clk_50mhz)
begin
    if clk_50mhz'event and clk_50mhz='1' then
        if one_second_count=49999999 then --divide by 50,000,000 is 1s
            one_second_count <= 0;
            one_second_pulse <= '1';
        else
            one_second_count <= one_second_count + 1;
            one_second_pulse <= '0';
        end if;
    --
    -- delay of 100 clock cycles before generating interrupt
    interrupt_delay <= interrupt_delay(98 downto 0) & one_second_pulse;
    --
    -- processor interrupt waits for an acknowledgement
    if interrupt_ack='1' then
        interrupt <= '0';
    elsif interrupt_delay(99)='1' then
        interrupt <= '1';
    else
        interrupt <= interrupt;
    end if;
    --
    -- counter selection switch toggles each second
    if one_second_pulse='1' then
        ab_switch <= not ab_switch;
    end if;
end process one_second_logic;
-----

```

Далее следуют операторы, с помощью которых создаются экземпляры компонентов исполнительного модуля микропроцессорно-ядра *kcpsm3* и программной памяти *fc_ctrl*:

```

-- Создание экземпляров исполнительного модуля KCPSM3 и про-
-- граммной памяти
--
processor: kcpsm3
port map(
  address => address,
  instruction => instruction,
  port_id => port_id,
  write_strobe => write_strobe,
  out_port => out_port,
  read_strobe => read_strobe,
  in_port => in_port,
  interrupt => interrupt,
  interrupt_ack => interrupt_ack,
  reset => reset,
  clk => clk_50mhz
);

--
program_rom: fc_ctrl
port map(
  address => address,
  instruction => instruction,
  proc_reset => reset, --JTAG Loader version
  clk => clk_50mhz
);
--reset <= '0'; --When using normal version

```

Для описания состояния входных и выходных портов микропроцессорного блока используется следующая последовательность операторов:

```

-- Входные порты микропроцессорного блока
--
-- The inputs are connected via a pipelined multiplexer
input_ports: process(clk_50mhz)
begin
  if clk_50mhz'event and clk_50mhz='1' then
    case port_id(7 downto 4) is
      -- read A-counter for addresses 00, 10, 20 and 30 hex
      when «0000» => in_port <= a_count(7 downto 0);
      when «0001» => in_port <= a_count(15 downto 8);
      when «0010» => in_port <= a_count(23 downto 16);
      when «0011» => in_port <= a_count(31 downto 24);
      --
      -- read B-counter for addresses 40, 50, 60 and 70 hex
      when «0100» => in_port <= b_count(7 downto 0);
      when «0101» => in_port <= b_count(15 downto 8);
      when «0110» => in_port <= b_count(23 downto 16);
      when «0111» => in_port <= b_count(31 downto 24);
      --
      -- read slide switches and counter circuit status at address 80 hex
      when «1000» => in_port <= «000» & ab_switch & sw;
      --
      -- read LCD data at address 90 hex
      when «1001» => in_port <= lcd_d & «0000»;
      --
      -- Don't care used to ensure minimum logic
      when others => in_port <= «XXXXXXXX»;
    end case;
  end if;
end process input_ports;

-- Выходные порты микропроцессорного блока
output_ports: process(clk_50mhz)
begin
  if clk_50mhz'event and clk_50mhz='1' then
    if write_strobe='1' then
      -- LED register at address 01 hex
      if port_id(0)= '1' then
        led <= out_port;
      end if;
      -- Counter reset controls at address 02 hex
      if port_id(1)= '1' then
        a_count_rst <= out_port(0);
        b_count_rst <= out_port(1);
      end if;
    end if;
  end if;
end process output_ports;

```

```

-- LCD data output and controls at address 04 hex.
if port_id(2)= '1' then
  lcd_output_data <= out_port(7 downto 4);
  lcd_drive <= out_port(3);
  lcd_rs <= out_port(2);
  lcd_rw_control <= out_port(1);
  lcd_e <= out_port(0);
end if;

-- Source selection and control at address 08 hex.
if port_id(3)= '1' then
  source_control <= out_port;
end if;
end if;
end if;
end process output_ports;

```

Завершают описание архитектуры рассматриваемого измерителя частоты цифровых сигналов следующие операторы, определяющие значения сигналов интерфейса управления ЖК-дисплеем:

```

-- Интерфейс управления жидкокристаллическим дисплеем
--
-- The 4-bit data port is bidirectional.
-- lcd_rw is '1' for read and '0' for write
-- lcd_drive is like a master enable signal which prevents either the
-- FPGA outputs or the LCD display driving the data lines.
--
-- Control of read and write signal
lcd_rw <= lcd_rw_control and lcd_drive;
--
-- use read/write control to enable output buffers.
lcd_d <= lcd_output_data when (lcd_rw_control='0' and lcd_drive='1')
else «ZZZZ»;
--
end Behavioral;

```

VHDL-описание генератора тестового сигнала с частотой 145 МГц, используемого в составе измерителя частоты цифровых сигналов

В составе VHDL-описания генератора тестового сигнала с частотой 145 МГц применяется компонент DCM_SPAR3_TEST, который представляет соответствующий вариант конфигурации цифрового модуля управления синхронизацией DCM в ПЛИС семейства Spartan-3E. Чтобы выполнить синтез, размещение и трассировку проекта, содержащего данный компонент, необходимо перед запуском средств проектирования серии Xilinx® ISE™ (Integrated Synthesis Environment/Integrated Software Environment) создать переменную окружения XIL_TEST_ARCS и присвоить ей единичное значение. Для этого в меню, которое открывается нажатием кнопки *Пуск (Start)* операционной системы Windows™, нужно выделить пункт *Мой компьютер (My computer)*, после чего щелкнуть на нем правой кнопкой мыши. Затем в появившемся на экране контекстно-зависимом всплывающем меню нужно выбрать команду *Свойства (Properties)*. В результате выполнения этой команды на экран выводится диалоговая панель с заголовком *Свойства системы (System Properties)*. В этой диалоговой панели следует открыть страницу *Дополнительно (Advanced)*, вид которой представлен на рис. 4.

Далее на указанной странице нужно нажать кнопку *Переменные среды (Environment variables)*,

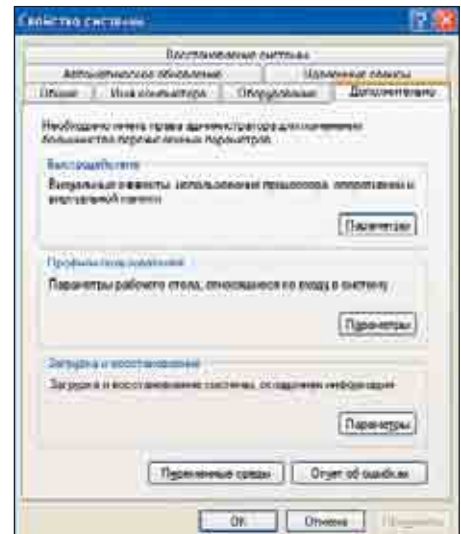


Рис. 4. Вид страницы Дополнительно (Advanced) диалоговой панели Свойства системы (System Properties)

после чего открывается одноименная диалоговая панель, показанная на рис. 5. Эта диалоговая панель предоставляет пользователю возможность создания, редактирования и удаления переменных окружения.



Рис. 5. Вид диалоговой панели Переменные среды (Environment variables), открывающей доступ к переменным окружения операционной системы Windows

Для создания новой переменной окружения следует во встроенной панели *Системные переменные (System variables)* (рис. 5) нажать кнопку *Создать (New)*. При этом на экране появится диалоговая панель с заголовком *Новая системная переменная (New system variable)*, вид которой изображен на рис. 6.

В соответствующих полях редактирования этой диалоговой панели необходимо указать идентификатор создаваемой переменной и ее значение. Затем нажатием кнопки *ОК* в диалоговой панели *Новая системная переменная (New system variable)* следует подтвердить ука-



Рис. 6. Вид диалоговой панели *Новая системная переменная (New system variable)*, предназначенной для создания новой переменной окружения операционной системы Windows

занные параметры создаваемой переменной. После этого идентификатор новой переменной и ее значение будут отражены в таблице, представленной во встроенной панели *Системные переменные (System variables)* (рис. 5). Процесс создания новой переменной завершается нажатием кнопки *ОК*, которая расположена в нижней части диалоговой панели *Переменные среды (Environment variables)*.

VHDL-описание генератора тестового сигнала, выполненного на основе цифрового модуля управления синхронизацией DCM, имеет следующий вид:

```

-----
-- Library declarations
--
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-----
-- Main Entity for dcm_fixed_osc
--
entity dcm_fixed_osc is
port(
    clk_out : out std_logic;
    kick_start : in std_logic
);
end dcm_fixed_osc;
-----
-- Start of Main Architecture for dcm_fixed_osc
--
architecture low_level_definition of dcm_fixed_osc is
-----
-- Definition of the special DCM test primitive
-----
component DCM_SPAR3_TEST
generic
    (CLKIN_PERIOD : real := 20.0); -- Avoids a MAP warning
but actual value not so important.
port(
    CLKIN : in std_ulogic;
    CTLMODE : in std_ulogic;
    CLKFX : out std_ulogic
);
end component;
-----
-- Circuit description
-----
begin
fixed_osc_dcm: DCM_SPAR3_TEST
port map (
    CLKIN => kick_start,
    CTLMODE => '1',
    CLKFX => clk_out
);
end low_level_definition;
-----

```

Приведенное описание соответствует типовой структуре модуля исходного описания проекта на языке VHDL. В его начале представлены ссылки на те же библиотеки и пакеты, что и в рассмотренном модуле исход-

ного описания верхнего уровня иерархии проекта измерителя частоты. Далее следует объявление объекта *dcm_fixed_osc*, который представляет рассматриваемый генератор. В состав интерфейса этого объекта входят два порта: входной и выходной. Для обозначения входа сигнала запуска используется идентификатор *kick_start*. Выход сигнала, формируемого рассматриваемым тестовым генератором, описывается в виде выходного порта *clk_out*. В блоке описания архитектуры объекта *dcm_fixed_osc* выполняется декларация компонента *DCM_SPAR3_TEST* и создается экземпляр этого компонента, который образует основу рассмотренного генератора.

При выполнении процессов синтеза, размещения и трассировки в кристалле проекта, содержащего компонент *DCM_SPAR3_TEST*, в САПР серии Xilinx ISE появляется несколько предупреждений, на которые, впрочем, не следует обращать внимание.

VHDL-описание генератора тестового сигнала с частотой 118 МГц, используемого в составе измерителя частоты цифровых сигналов

Функциональная схема генератора тестового сигнала с частотой 118 МГц показана на рис. 7. Для реализации этого генератора используются таблицы преобразования LUT, формирующие необходимую задержку и инвертирование сигнала. Кроме того, в состав схемы входит D-триггер, который применяется в качестве делителя частоты на два.

Описание кольцевого генератора тестового сигнала на языке VHDL выполнено в соответствии с функциональной схемой, представленной на рис. 5. Текст этого описания выглядит следующим образом:

```

-----
-- Library declarations
--
-- The Unisim Library is used to define Xilinx primitives. It is also used
during
-- simulation. The source can be viewed at XILINX%\vhdl\src\
unisims\unisim_VCOMP.vhd
--
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
library unisim;
use unisim.vcomponents.all;
-----
-- Main Entity for ring_osc
--
entity ring_osc is

```

```

port(
    osc_out : out std_logic;
    reset : in std_logic
);
end ring_osc;
-----
-- Start of Main Architecture for ring_osc
--
architecture low_level_definition of ring_osc is
-----
-- Signals used in RING_OSC
-----
signal ring_delay1 : std_logic;
signal ring_delay2 : std_logic;
signal ring_delay3 : std_logic;
signal ring_delay4 : std_logic;
signal ring_invert : std_logic;
signal toggle : std_logic;
signal clk_div2 : std_logic;
-----
-- Attributes to stop delay logic from being optimised.
--
attribute KEEP : string;
attribute KEEP of ring_delay1 : signal is «true»;
attribute KEEP of ring_delay2 : signal is «true»;
attribute KEEP of ring_delay3 : signal is «true»;
attribute KEEP of ring_delay4 : signal is «true»;
-----
-- Attributes to define LUT contents during implementation for primitives not
-- contained within generate loops. In each case the information is repeated
-- in the generic map for functional simulation
--
attribute INIT : string;
attribute INIT of div2_lut : label is «1»;
attribute INIT of delay1_lut : label is «4»;
attribute INIT of delay2_lut : label is «4»;
attribute INIT of delay3_lut : label is «4»;
attribute INIT of delay4_lut : label is «4»;
attribute INIT of invert_lut : label is «B»;
-----
-- Circuit description
-----
begin
--
--Output is the ring oscillator divided by 2 to provide a square wave.
osc_out <= clk_div2;
--
toggle_flop: FD
port map (
    D => toggle,
    Q => clk_div2,
    C => ring_invert
);
--
div2_lut: LUT2
--synthesis translate_off
generic map (INIT => X»1»)
--synthesis translate_on
port map(
    I0 => reset,
    I1 => clk_div2,
    O => toggle
);
--
--Ring oscillator is formed of 5 levels of logic of which one is an in-
verter.
--
delay1_lut: LUT2
--synthesis translate_off
generic map (INIT => X»4»)
--synthesis translate_on
port map(
    I0 => reset,
    I1 => ring_invert,
    O => ring_delay1
);
--

```

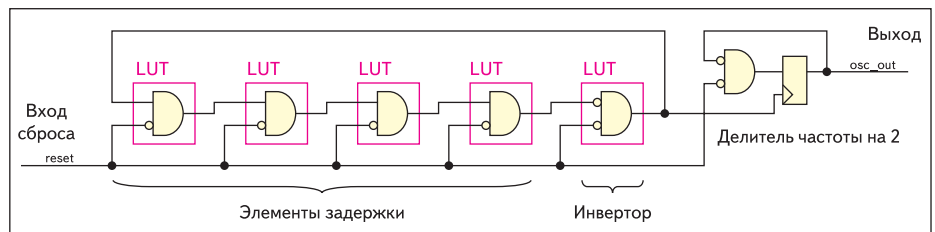


Рис. 7. Функциональная схема кольцевого генератора тестового сигнала с частотой 118 МГц

```

delay2_lut: LUT2
--synthesis translate_off
generic map (INIT => X«4»)
--synthesis translate_on
port map(
    I0 => reset,
    I1 => ring_delay1,
    O => ring_delay2
);
--
delay3_lut: LUT2
--synthesis translate_off
generic map (INIT => X«4»)
--synthesis translate_on
port map(
    I0 => reset,
    I1 => ring_delay2,
    O => ring_delay3
);
--
delay4_lut: LUT2
--synthesis translate_off
generic map (INIT => X«4»)
--synthesis translate_on
port map(
    I0 => reset,
    I1 => ring_delay3,
    O => ring_delay4
);
--
invert_lut: LUT2
--synthesis translate_off
generic map (INIT => X«B»)
--synthesis translate_on
port map(
    I0 => reset,
    I1 => ring_delay4,
    O => ring_invert
);
end low_level_definition;

```

В приведенном описании после ссылок на используемые стандартные логические библиотеки и библиотеку Unisim фирмы Xilinx объявляется объект *ring_osc*, который представляет рассматриваемый кольцевой генератор тестового сигнала. Интерфейс этого объекта включает входной порт сигнала сброса *reset* и выходной порт сигнала, формируемого генератором, *osc_out*. В начале блока описания архитектуры объекта *ring_osc* приведены выражения декларации внутренних сигналов, которые используются для сопряжения компонентов генератора. После этого следуют выражения, устанавливающие значения необходимых атрибутов. Определение архитектуры рассматриваемого генератора выполнено в виде структурного описания, компонентами которого являются элементы, соответствующие функциональной схеме (рис. 7). Это описание включает в себя операторы создания шести экземпляров компонентов таблиц преобразования и одного экземпляра D-триггера.

Временные и топологические ограничения проекта измерителя частоты цифровых сигналов

Временные ограничения в проекте измерителя частоты цифровых сигналов заданы для цепей основного тактового сигнала и входной цепи измерительного блока. Для цепи основного сигнала синхронизации *clk_50mhz* временные ограничения устанавливают максимальное значение периода этого сигнала, соответствующее частоте 50 МГц. Ограничения, накладываемые на время рас-

пространения сигналов по входной цепи измерительного блока *test_clk*, соответствуют максимальному значению частоты этих сигналов, равному 200 МГц. Выражения, определяющие временные ограничения для проекта измерителя частоты цифровых сигналов, выглядят следующим образом:

```

# Period constraint for 50MHz operation
#
NET «clk_50mhz» PERIOD = 20.0ns HIGH 50%;
#
# Period constraint for up to 200MHz operation
#
NET «test_clk» PERIOD = 5.0ns HIGH 50%;

```

Топологические ограничения рассматриваемого проекта измерителя частоты цифровых сигналов устанавливают соответствие интерфейсных портов (цепей) объекта *frequency_counter* и номеров выводов ПЛИС с учетом архитектуры инструментального модуля Xilinx Spartan-3E Starter Board версии С. Кроме того, в выражениях топологических ограничений указываются дополнительные параметры интерфейсных портов этого объекта, в частности стандарты ввода/вывода, в соответствии с которыми они должны быть сконфигурированы. Совокупность выражений, определяющих топологические ограничения для проекта измерителя частоты цифровых сигналов, имеет следующий вид:

```

# soldered 50MHz Clock.
#
NET «clk_50mhz» LOC = «C9» | IOSTANDARD = LVTTTL;
#
# SMA socket.
#
NET «sma_clk» LOC = «A10» | IOSTANDARD = LVTTTL;
#
# Simple LEDs
# Require only 3.5mA.
#
NET «led<0>» LOC = «F12» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
NET «led<1>» LOC = «E12» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
NET «led<2>» LOC = «E11» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
NET «led<3>» LOC = «F11» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
NET «led<4>» LOC = «C11» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
NET «led<5>» LOC = «D11» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
NET «led<6>» LOC = «E9» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
NET «led<7>» LOC = «F9» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 4;
#
# LCD display
# Very slow so can use lowest drive strength.
#
NET «lcd_rs» LOC = «L18» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;
NET «lcd_rw» LOC = «L17» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;
NET «lcd_e» LOC = «M18» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;
NET «lcd_d<4>» LOC = «R15» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;
NET «lcd_d<5>» LOC = «R16» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;
NET «lcd_d<6>» LOC = «P17» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;
NET «lcd_d<7>» LOC = «M15» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;
#
# Strata Flash (need to disable to use LCD display)
#
NET «strataflash_oe» LOC = «C18» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;

```

```

NET «strataflash_ce» LOC = «D16» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;
NET «strataflash_we» LOC = «D17» | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 2;
#
# Simple switches
# Pull UP resistors used to stop floating condition during switching.
#
NET «sw<0>» LOC = «L13» | IOSTANDARD = LVTTTL | PULLUP;
NET «sw<1>» LOC = «L14» | IOSTANDARD = LVTTTL | PULLUP;
NET «sw<2>» LOC = «H18» | IOSTANDARD = LVTTTL | PULLUP;
NET «sw<3>» LOC = «N17» | IOSTANDARD = LVTTTL | PULLUP;

```

В начале приведенной последовательности топологических ограничений представлены выражения, осуществляющие привязку цепей внешнего тактового сигнала и входного измеряемого сигнала к выводам ПЛИС, которые подключены на плате инструментального модуля к выходу кварцевого генератора и высокочастотному разъему типа SMA соответственно. Далее следуют выражения, определяющие номера выводов кристалла, сопряженных со светодиодными индикаторами, для выходных цепей сигналов индикации выполнения измерений. В следующем фрагменте сосредоточены ограничения, выполняющие привязку к выводам ПЛИС интерфейсных портов, предназначенных для подключения ЖК-дисплея. Затем приведены выражения, определяющие номера контактов кристалла, на которые должны быть выведены цепи сигналов управления для неиспользуемого модуля Flash-памяти. В заключительной части представлены выражения ограничений для портов управляющих сигналов, в которых указаны выводы кристалла, сопряженные с блоком ползунковых переключателей. ■

Окончание следует

Литература

1. Зотов В. Цифровой генератор сигнала с перестраиваемой частотой, реализуемый на базе инструментального комплекта Spartan-3E Starter Kit фирмы Xilinx // Компоненты и технологии. 2006. № 11–12.
2. Зотов В. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx. М.: Горячая линия — Телеком, 2006.
3. Зотов В. PicoBlaze — семейство восьмиразрядных микропроцессорных ядер, реализуемых на основе ПЛИС фирмы Xilinx // Компоненты и технологии. 2003. № 4.
4. Зотов В. Система команд микропроцессорного ядра PicoBlaze, реализуемого на основе ПЛИС семейства Spartan-II, Spartan-IIe, Virtex, Virtex-E // Компоненты и технологии. 2003. № 5.
5. Зотов В. Особенности микропроцессорного ядра PicoBlaze, предназначенного для применения в проектах, реализуемых на основе ПЛИС семейства Virtex-II // Компоненты и технологии. 2003. № 6.
6. Зотов В. Особенности микропроцессорного ядра PicoBlaze, предназначенного для применения в проектах, реализуемых на основе ПЛИС семейства CoolRunner-II // Компоненты и технологии. 2003. № 7.

7. Зотов В. Особенности микропроцессорного ядра PicoBlaze, предназначенного для применения в проектах, реализуемых на основе ПЛИС семейств Spartan-3, Virtex-II и Virtex-II PRO // Компоненты и технологии. 2005. № 5–6.
8. Зотов В. MicroBlaze — семейство тридцатидвухразрядных микропроцессорных ядер, реализуемых на основе ПЛИС фирмы Xilinx // Компоненты и технологии. 2003. № 9.
9. Зотов В. Система команд микропроцессорного ядра MicroBlaze // Компоненты и технологии. 2004. № 1–3.
10. Зотов В. Организация памяти микропроцессорного ядра MicroBlaze // Компоненты и технологии. 2004. № 5.
11. Зотов В. Новый инструментальный комплект Spartan-3E Starter Kit для практического освоения методов проектирования встраиваемых микропроцессорных систем на основе ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2006. № 10.
12. Бибило П. Н. Основы языка VHDL. М.: Солон-Р, 2000.
13. Бибило П. Н. Синтез логических схем с использованием языка VHDL. М.: Солон-Р, 2002.
14. Уэйкерли Дж. Ф. Проектирование цифровых устройств. Т. 1. М.: Постмаркет, 2002.