

Продолжение. Начало в № 2`2010

Разработка VHDL-описаний цифровых устройств, проектируемых на основе ПЛИС фирмы Xilinx, с использованием шаблонов САПР ISE Design Suite

Седьмая часть статьи завершает ознакомление с шаблонами VHDL-описаний компонентов, основанных на использовании библиотечных примитивов, которые предназначены для непосредственной реализации на базе соответствующих аппаратных ресурсов ПЛИС семейств CPLD [1]. Здесь рассматриваются образцы описаний основных вариантов конфигурации триггеров с динамическим и потенциальным управлением (защелок), которые представлены в составе макроячеек кристаллов программируемой логики с архитектурой CPLD. Кроме того, в этой же части приводится информация о шаблонах описания компонентов, реализуемых на основе специализированных ресурсов ПЛИС семейства Spartan-3 [1, 6]. В том числе представлены образцы VHDL-описаний аппаратных умножителей и элементов, предназначенных для коммутации и формирования тактовых сигналов в кристаллах программируемой логики с архитектурой FPGA указанного семейства.

Валерий ЗОТОВ
walerry@km.ru

В подразделе *Registers & Latches* содержатся образцы описаний экземпляров триггеров с динамическим и потенциальным управлением (защелок), входящих в состав архитектуры ПЛИС семейств CPLD. Данный подраздел включает в себя два каталога — *Double Data Rate (CR-II)* и *Single Data Rate* (КиТ. 2010. № 6. Рис. 39). В каталоге *Double Data Rate (CR-II)* представлены шаблоны описаний триггеров, тактируемых фронтом и спадом сигнала синхронизации, которые можно использовать в составе устройств, проектируемых на базе кристаллов программируемой логики семейства CoolRunner-II.

ПЛИС указанного семейства отличаются от других кристаллов с архитектурой CPLD, выпускаемых фирмой Xilinx, прежде всего возможностью конфигурирования триггерного элемента, входящего в состав макроячейки, не только в виде D- или T-триггера с динамическим управлением, с потенциальным управлением (защелки), но и триггера, тактируемого фронтом и спадом сигнала синхронизации [1]. При этом в каждой макроячейке ПЛИС семейства CoolRunner-II предусмотрена возможность использования сигнала разрешения синхронизации (clock enable), а также цепей сброса и установки триггерных элементов. Ресурсы кристаллов рассматриваемого семейства предоставляют возможность комплексного асинхронного

тактирования элементов проектируемого устройства с использованием сигналов, формируемых внутри функционального блока, и четырех глобальных сигналов синхронизации, поступающих с соответствующих выводов ПЛИС. Макроячейки кристаллов программируемой логики семейства CoolRunner-II позволяют реализовать режимы асинхронного сброса или асинхронной установки триггера, используя для этого локальные и глобальные управляющие сигналы. Для различных вариантов конфигурации триггерных элементов в кристаллах указанного семейства предлагаются соответствующие шаблоны VHDL-описаний.

D-FF/w Async Clear, Set and CE (FDDCPE) представляет собой образец описания D-триггера, управляемого фронтом и спадом тактового сигнала, со входами разрешения синхронизации, асинхронного сброса и асинхронной установки. Основой этого шаблона является оператор создания экземпляра библиотечного элемента *FDDCPE*:

```
-- FDDCPE : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (FDDCPE_inst) and/or the port declarations
-- code : after the "=" assignment maybe changed to properly
-- : connect this function to the design. All inputs must be
-- : connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
```

```
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- FDDCPE: Double Data Rate Register with Asynchronous Clear
and Set
-- and Clock Enable (Clear has priority). CoolRunner-II
-- Xilinx HDL Language Template, version 11.4
--
FDDCPE_inst : FDDCPE
port map (
    Q => Q, -- Data output
    C => C, -- Clock input
    CE => CE, -- Clock enable input
    CLR => CLR, -- Asynchronous clear input
    D => D, -- Data input
    PRE => PRE -- Asynchronous set input
);
-- End of FDDCPE_inst instantiation
```

В описании интерфейса библиотечного примитива *FDDCPE* применяется следующая система условных обозначений входных и выходных портов:

- Q — выход триггера;
- C — вход сигнала синхронизации;
- CE — вход разрешения тактового сигнала;
- CLR — вход асинхронного сброса;
- D — информационный вход D;
- PRE — вход асинхронной установки.

Запись входных данных в триггер, формируемый с помощью шаблона *D-FF/w Async Clear, Set and CE (FDDCPE)*, производится при высоком логическом уровне сигнала на входе разрешения CE по каждому переключению сигнала синхронизации. Для входов асинхронного сброса CLR и асинхронной установки PRE активным является высокий логический уровень сигнала. При этом вход асинхронного сброса имеет более высокий приоритет, чем вход асинхронной установки.

Условный графический образ D-триггера, управляемого фронтом и спадом тактового сигнала, со входами разрешения синхронизации, асинхронного сброса и асинхронной установки, описание которого создается с помощью шаблона *D-FF/w Async Clear, Set and CE (FDDCPE)*, приведен на рис. 47.

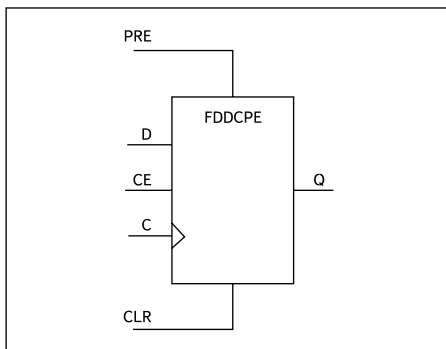


Рис. 47. Условный графический образ D-триггера, управляемого фронтом и спадом тактового сигнала, со входами разрешения синхронизации, асинхронного сброса и асинхронной установки

T-FF/w Async Clear, Set, Load and CE (FTDCPLE) содержит шаблон описания T-триггера с синхронной загрузкой информации, с входами разрешения синхронизации, асинхронного сброса и асинхронной установки, управляемого фронтом и спадом импульса синхронизации. Данный шаблон выполнен на основе использования библиотечного примитива *FTDCPLE*, который представляет соответствующий вариант конфигурирования триггера макроячейки в кристаллах программируемой логики семейства CoolRunner-II:

```
-- FTDCPLE : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (FTDCPLE_inst) and/or the port declarations
-- code : after the ">" assignment maybe changed to properly
-- : connect this function to the design. All inputs must be
-- : connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
```

```
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- FTDCPLE: Double Data Rate T Flip-Flop with Asynchronous
Clear, Set,
-- Load and Clock Enable (Clear has priority). CoolRunner-II
-- Xilinx HDL Language Template, version 11.4
--
FTDCPLE_inst : FTDCPLE
port map (
    Q => Q, -- Data output
    C => C, -- Clock input
    CE => CE, -- Clock enable input
    CLR => CLR, -- Asynchronous clear input
    D => D, -- Load data input
    L => L, -- Load enable input
    PRE => PRE, -- Asynchronous set input
    T => T -- T data input
);
-- End of FTDCPLE_inst instantiation
```

Система условных обозначений интерфейсных портов библиотечного примитива *FTDCPLE*, кроме наименований входов и выходов, представленных при рассмотрении шаблона *D-FF/w Async Clear, Set and CE (FDDCPE)*, включает в себя также следующие идентификаторы:

- L — вход разрешения записи информации с входа D;
- T — вход разрешения переключений.

В триггерах, формируемых с помощью шаблона *T-FF/w Async Clear, Set, Load and CE (FTDCPLE)*, наибольшим приоритетом обладает вход асинхронного сброса, а вход асинхронной установки имеет более высокий приоритет, чем вход разрешения записи информации с входа D. При наличии активного (высокого) уровня сигнала на входе разрешения записи информации L данные с входа D загружаются в триггер по фронту или спаду тактового сигнала. Если на входе разрешения записи информации L присутствует низкий логический уровень, а на входе разрешения тактового сигнала CE — высокий логический уровень, то при любом изменении сигнала синхронизации выход триггера переключается в противоположное состояние.

Условный графический образ T-триггера с синхронной загрузкой информации, с входами разрешения синхронизации, асинхронного сброса и асинхронной установки, управляемого фронтом и спадом тактового импульса, для подготовки описания которого используется шаблон *T-FF/w Async Clear, Set, Load and CE (FTDCPLE)*, изображен на рис. 48.

В каталоге *Single Data Rate* сосредоточены шаблоны описаний триггеров, тактируемых фронтом сигнала синхронизации, которые можно применять в составе устройств, разрабатываемых на базе ПЛИС с архитектурой CPLD различных семейств.

Posedge D-FF/w Async Clear, Set and CE (FDCPE) предоставляет образец описания D-триггера с асинхронным сбросом, асинхронной установкой и входом разрешения тактового сигнала:

```
-- FDCPE : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (FDCPE_inst) and/or the port declarations
```

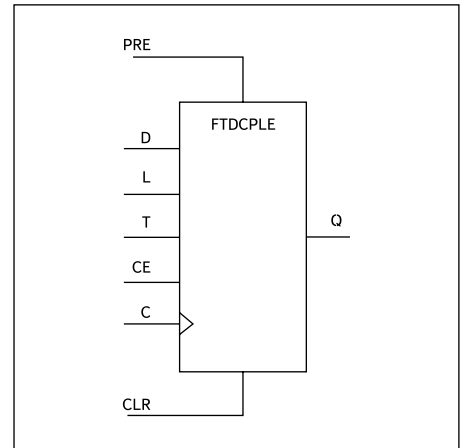


Рис. 48. Условный графический образ T-триггера с синхронной загрузкой информации, с входами разрешения синхронизации, асинхронного сброса и асинхронной установки, управляемого фронтом и спадом тактового импульса

```
-- code : after the ">" assignment maybe changed to properly
-- : connect this function to the design. All inputs must be
-- : connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear,
Set and
-- Clock Enable (posedge clk). All families.
-- Xilinx HDL Language Template, version 11.4
--
FDCPE_inst : FDCPE
generic map (
    INIT => '0' -- Initial value of register ('0' or '1')
)
port map (
    Q => Q, -- Data output
    C => C, -- Clock input
    CE => CE, -- Clock enable input
    CLR => CLR, -- Asynchronous clear input
    D => D, -- Data input
    PRE => PRE -- Asynchronous set input
);
-- End of FDCPE_inst instantiation
```

В приведенном шаблоне используется единственный параметр настройки INIT, значение которого определяет начальное состояние триггера, в которое он устанавливается при подаче напряжения питания ПЛИС. По умолчанию предлагается нулевое значение этого параметра, соответствующее сброшенному состоянию триггера в начальный момент времени функционирования проектируемого устройства, в составе которого он применяется.

Назначение идентификаторов входных и выходных портов библиотечного примитива *FDCPE* было рассмотрено выше при описании шаблона *D-FF/w Async Clear, Set and CE (FDDCPE)*. На рис. 49 показан услов-

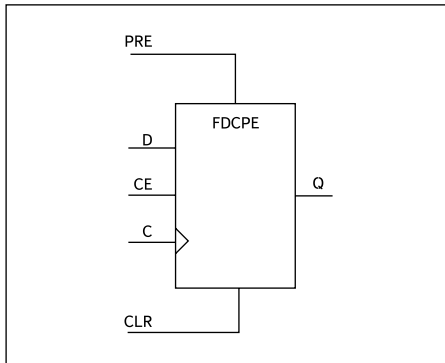


Рис. 49. Условный графический образ D-триггера, управляемого фронтом тактового сигнала, с входами разрешения синхронизации, асинхронного сброса и асинхронной установки

ный графический образ D-триггера с входами разрешения синхронизации, асинхронного сброса и асинхронной установки, описание которого формируется на основе шаблона *Posedge D-FF/w Async Clear, Set and CE (FDCPE)*.

Posedge D-FF/w Async Clear and CE (FDCE) включает в себя шаблон описания D-триггера с входом разрешения тактового сигнала и асинхронным сбросом. В качестве основы этого шаблона используется экземпляр библиотечного примитива *FDCE*:

```
-- FDCE : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (FDCE_inst) and/or the port declarations
-- code : after the "=" assignment maybe changed to properly
-- : connect this function to the design. All inputs must be
-- : connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body---->
--
-- FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
-- Clock Enable (posedge clk). All families.
-- Xilinx HDL Language Template, version 11.4
--
FDCE_inst : FDCE
generic map (
  INIT => '0' -- Initial value of register ('0' or '1')
)
port map (
  Q => Q, -- Data output
  C => C, -- Clock input
  CE => CE, -- Clock enable input
  CLR => CLR, -- Asynchronous clear input
  D => D -- Data input
);
-- End of FDCE_inst instantiation
```

Условный графический образ D-триггера, управляемого фронтом тактового сигнала, с входами разрешения синхронизации и асинхронного сброса, описание которого создается с помощью представленного шаблона, изображен на рис. 50.

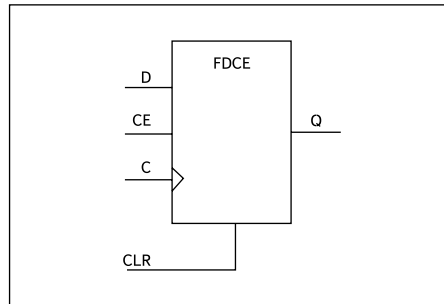


Рис. 50. Условный графический образ D-триггера, управляемого фронтом тактового сигнала, с входами разрешения синхронизации и асинхронного сброса

Шаблоны описаний экземпляров компонентов, выполненных на основе библиотечных примитивов, предназначенных для непосредственной реализации на базе соответствующих аппаратных ресурсов ПЛИС семейства Spartan-3

Раздел *Spartan-3* объединяет в себе образцы VHDL-описаний экземпляров компонентов, построенных на основе библиотечных примитивов, которые предназначены для непосредственной реализации на базе соответствующих аппаратных ресурсов кристаллов программируемой логики одноименного семейства. Подробная структура данного раздела представлена на рис. 51.

В состав раздела *Spartan-3* входят семь подразделов: *Arithmetic Functions, Clock Components, Config/BSCAN Components, I/O Components, RAM/ROM, Registers & Latches* и *Slice/CLB Primitives*. В подразделе *Arithmetic Functions* находятся шаблоны VHDL-описа-

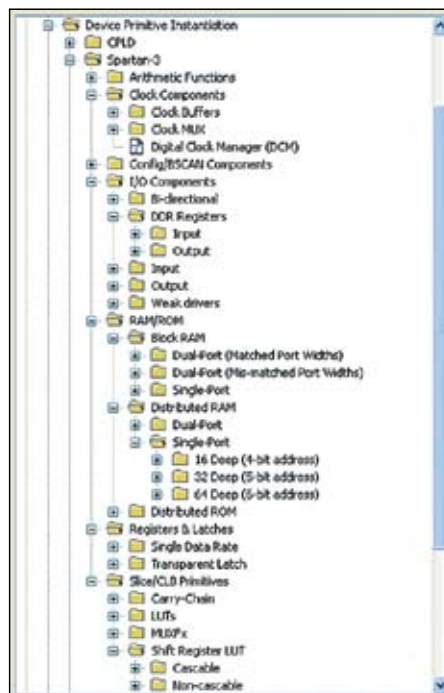


Рис. 51. Структура папки Spartan-3 раздела Device Primitive Instantiation шаблонов языка VHDL

ний компонентов, реализуемых на базе аппаратных умножителей, которые представлены в составе архитектуры ПЛИС семейства Spartan-3.

18X18 Asynchronous Multiplier (MULT18X18) содержит образец описания асинхронного умножителя двух 18-разрядных операндов, который предназначен для реализации на базе соответствующего аппаратного модуля ПЛИС семейства Spartan-3. Для подготовки VHDL-описания асинхронного аппаратного умножителя применяется библиотечный примитив *MULT18X18*.

```
-- MULT18X18 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (MULT18X18_inst) and/or the port
declarations
-- code : after the "=" assignment maybe changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body---->
--
-- MULT18X18: 18 x 18 signed asynchronous multiplier
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
MULT18X18_inst : MULT18X18
port map (
  P => P, -- 36-bit multiplier output
  A => A, -- 18-bit multiplier input
  B => B -- 18-bit multiplier input
);
-- End of MULT18X18_inst instantiation
```

В представленном шаблоне используется следующая система условных обозначений входных и выходных портов:

- P — 36-разрядная выходная шина данных, на которой отображается значение произведения входных операндов;
- A — 18-разрядная входная шина данных, совокупность сигналов которой образует значение первого сомножителя;
- B — 18-разрядная входная шина данных, совокупность сигналов которой образует значение второго сомножителя.

На рис. 52 показан условный графический образ асинхронного аппаратного умножителя, формируемого с помощью шаблона *18X18 Asynchronous Multiplier (MULT18X18)*.

18X18 Synchronous Multiplier (MULT18X18S) является шаблоном описания синхронного умножителя двух 18-разрядных операндов, предназначенного для реализации на базе соответствующих аппаратных ресурсов кристаллов программируемой логики семейства Spartan-3. Этот шаблон выполнен в виде конструкции, построенной на базе библиотечного примитива *MULT18X18S*:

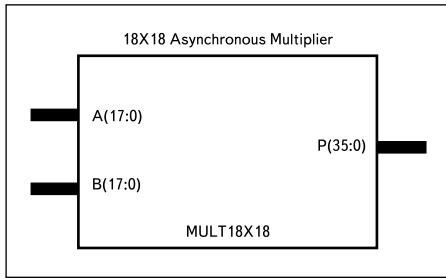


Рис. 52. Условный графический образ асинхронного аппаратного умножителя, реализуемого на базе ПЛИС семейства Spartan-3

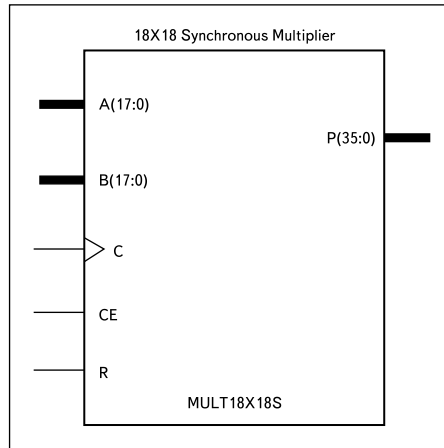


Рис. 53. Условный графический образ синхронного аппаратного умножителя, реализуемого на базе ПЛИС семейства Spartan-3

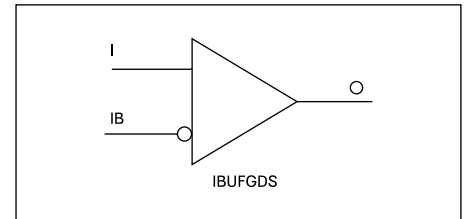


Рис. 54. Условный графический образ глобального буферного элемента с дифференциальными входами, реализуемого на базе ПЛИС семейства Spartan-3

```
-- MULT18X18S: In order to incorporate this function into the design,
-- VHDL: the following instance declaration needs to be placed
-- instance: in the architecture body of the design code. The
-- declaration: instance name (MULT18X18S_inst) and/or the port
declarations
-- code: after the "=" assignment maybe changed to properly
--: reference and connect this function to the design.
--: All inputs and outputs must be connected.
--
-- Library: In addition to adding the instance declaration, a use
-- declaration: statement for the UNISIM.vcomponents library needs to be
-- for: added before the entity declaration. This library
-- Xilinx: contains the component declarations for all Xilinx
-- primitives: primitives and points to the models that will be used
--: for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
--<-----Cut code below this line and paste into the architecture body----->
--
-- MULT18X18S: 18 x 18 signed synchronous multiplier
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
MULT18X18S_inst : MULT18X18S
port map (
    P => P, -- 36-bit multiplier output
    A => A, -- 18-bit multiplier input
    B => B, -- 18-bit multiplier input
    C => C, -- Clock input
    CE => CE, -- Clock enable input
    R => R -- Synchronous reset input
);
-- End of MULT18X18S_inst instantiation
```

Система условных обозначений интерфейсных портов, используемая в шаблоне *18X18 Synchronous Multiplier (MULT18X18S)*, кроме наименований входных и выходных шин, представленных в предыдущем шаблоне, включает в себя также следующие идентификаторы:

- C — вход сигнала синхронизации;
- CE — вход разрешения тактового сигнала;
- R — вход синхронного сброса.

Условный графический образ синхронного аппаратного умножителя, для подготовки описания которого используется шаблон *18X18 Synchronous Multiplier (MULT18X18S)*, приведен на рис. 53.

Подраздел *Clock Components* содержит два каталога — *Clock Buffers* и *Clock MUX* (рис. 51). В каталоге *Clock Buffers* представлены образцы VHDL-описаний глобальных буферных элементов, используемых для подключения сигналов синхронизации. Каталог *Clock MUX* включает в себя шаблоны описаний глобальных мультиплексоров, предназначенных для коммутации тактовых сигналов. Кроме того,

в подразделе *Clock Components* находится образец описания цифрового модуля управления синхронизацией DCM (Digital Clock Manager), входящего в состав архитектуры ПЛИС семейства Spartan-3 [1, 19, 20].

Differential Input Global Clock Buffer (IBUFGDS) предоставляет шаблон описания входного глобального буферного элемента с дифференциальными входами, который применяется для подключения внешних сигналов синхронизации, представленных в соответствии с указанными стандартами ввода/вывода. Основой этого шаблона является оператор создания экземпляра библиотечно-го элемента *IBUFGDS*:

```
-- IBUFGDS: In order to incorporate this function into the design,
-- VHDL: the following instance declaration needs to be placed
-- instance: in the architecture body of the design code. The
-- declaration: instance name (IBUFGDS_inst) and/or the port declarations
-- code: after the "=" assignment maybe changed to properly
--: reference and connect this function to the design.
--: All inputs and outputs must be connected.
--
-- Library: In addition to adding the instance declaration, a use
-- declaration: statement for the UNISIM.vcomponents library needs to be
-- for: added before the entity declaration. This library
-- Xilinx: contains the component declarations for all Xilinx
-- primitives: primitives and points to the models that will be used
--: for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
--<-----Cut code below this line and paste into the architecture
body----->
--
-- IBUFGDS: Differential Global Clock Buffer (sourced by an external
pin)
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
IBUFGDS_inst : IBUFGDS
generic map (
    IOSTANDARD => "DEFAULT"
)
port map (
    O => O, -- Clock buffer output
    I => I, -- Diff_p clock buffer input
    IB => IB -- Diff_n clock buffer input
);
-- End of IBUFGDS_inst instantiation
```

В библиотечном примитиве *IBUFGDS* предусмотрен параметр настройки

IOSTANDARD, значение которого определяет используемый дифференциальный стандарт ввода/вывода.

В шаблоне описания входного глобального буферного элемента с дифференциальными входами *Differential Input Global Clock Buffer (IBUFGDS)* используются следующие идентификаторы входных и выходных портов:

- O — выход буферного элемента;
- I — прямой вход дифференциальной пары сигналов;
- IB — инверсный вход дифференциальной пары сигналов.

На рис. 54 показан условный графический образ глобального входного буферного элемента с дифференциальными входами, описание которого формируется с помощью библиотечного примитива *IBUFGDS* для последующего использования в проектах цифровых устройств, разрабатываемых на базе ПЛИС семейства Spartan-3.

Global Clock Buffer/w Enable (BUFGCE) включает в себя конструкцию, предназначенную для подготовки описания глобального буферного элемента с входом разрешения тактового сигнала, который используется в составе проектируемых устройств, выполняемых на базе кристаллов программируемой логики семейства Spartan-3:

```
-- BUFGCE: In order to incorporate this function into the design,
-- VHDL: the following instance declaration needs to be placed
-- instance: in the architecture body of the design code. The
-- declaration: instance name (BUFGCE_inst) and/or the port declarations
-- code: after the "=" assignment maybe changed to properly
--: reference and connect this function to the design.
--: All inputs and outputs must be connected.
--
-- Library: In addition to adding the instance declaration, a use
-- declaration: statement for the UNISIM.vcomponents library needs to be
-- for: added before the entity declaration. This library
-- Xilinx: contains the component declarations for all Xilinx
-- primitives: primitives and points to the models that will be used
--: for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
--<-----Cut code below this line and paste into the architecture body----->
--
-- BUFGCE: Global Clock Buffer with Clock Enable (active high)
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
BUFGCE_inst : BUFGCE
port map (
    O => O, -- Clock buffer output
    CE => CE, -- Clock enable input
    I => I -- Clock buffer input
);
-- End of BUFGCE_inst instantiation
```

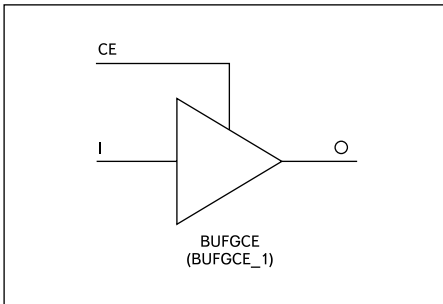


Рис. 55. Условный графический образ глобального буферного элемента с входом разрешения тактового сигнала, реализуемого на базе ПЛИС семейства Spartan-3

Представленная конструкция выполнена на основе библиотечного примитива *BUFGCE*. В систему условных обозначений интерфейсных портов этого примитива, кроме идентификаторов, рассмотренных при описании предыдущего шаблона, входит также входной порт *CE*, предназначенный для подключения сигнала разрешения синхронизации. Активным уровнем данного сигнала в шаблоне *Global Clock Buffer/w Enable (BUFGCE)* является высокий логический уровень напряжения (уровень логической единицы).

Условный графический образ глобального буферного элемента с входом разрешения тактового сигнала, описание которого выполняется на основе библиотечного примитива *BUFGCE*, показан на рис. 55.

Global Clock Buffer (BUFG) содержит образец варианта использования примитива *BUFG* для подготовки описания глобального буферного элемента, предназначенного для применения в составе цифровых устройств, разрабатываемых на базе ПЛИС семейства Spartan-3.

```
-- BUFG : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (BUFG_inst) and/or the port declarations
-- code : after the "=" assignment may be changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- BUFG: Global Clock Buffer (source by an internal signal)
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
BUFG_inst : BUFG
port map (
    O => O, -- Clock buffer output
    I => I -- Clock buffer input
);
-- End of BUFG_inst instantiation
```

Условный графический образ глобального буферного элемента, для подготовки описания которого используется шаблон *Global Clock Buffer (BUFG)*, показан на рис. 43 (КиТ. 2010. № 6).

Global Clock Buffer/w Enable (BUFGCE_1) предоставляет шаблон описания глобального буферного элемента с инверсным входом разрешения тактового сигнала, который предназначен для использования в составе проектируемых устройств, реализуемых на базе кристаллов программируемой логики семейства Spartan-3. В качестве основы данной конструкции применяется библиотечный примитив *BUFGCE_1*. В отличие от примитива *BUFGCE*, рассмотренного выше, в глобальном буферном элементе, создаваемом с помощью шаблона *Global Clock Buffer/w Enable (BUFGCE_1)*, активным уровнем для входа разрешения тактового сигнала является низкий логический уровень (уровень логического нуля).

```
-- BUFGCE_1 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (BUFGCE_1_inst) and/or the port
declarations
-- code : after the "=" assignment may be changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- BUFGCE_1: Global Clock Buffer with Clock Enable (active low)
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
BUFGCE_1_inst : BUFGCE_1
port map (
    O => O, -- Clock buffer output
    CE => CE, -- Clock enable input
    I => I -- Clock buffer input
);
-- End of BUFGCE_1_inst instantiation
```

Условный графический образ глобального буферного элемента с инверсным входом разрешения тактового сигнала, формируемого с помощью представленного шаблона, изображен на рис. 55.

Single-ended Input Global Clock Buffer (IBUFG) представляет собой образец описания входного глобального буферного элемента для использования в проектах цифровых устройств, разрабатываемых на основе ПЛИС семейства Spartan-3.

```
-- IBUFG : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (IBUFG_inst) and/or the port declarations
-- code : after the "=" assignment may be changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
```

```
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- IBUFG: Global Clock Buffer (sourced by an external pin)
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
IBUFG_inst : IBUFG
generic map (
    IOSTANDARD => "DEFAULT"
)
port map (
    O => O, -- Clock buffer output
    I => I -- Clock buffer input (connect directly to top-level port)
);
-- End of IBUFG_inst instantiation
```

В представленном шаблоне для формирования экземпляра входного глобального буферного элемента используется библиотечный примитив *IBUFG*. В данном примитиве для определения используемого стандарта ввода/вывода предусмотрен настраиваемый параметр *IOSTANDARD*.

Условный графический образ входного глобального буферного элемента, создаваемого с помощью шаблона *Single-ended Input Global Clock Buffer (IBUFG)*, отличается от символа, изображенного на рис. 43, только названием.

Global Clock Buffer MUX (BUFGMUX) является шаблоном описания глобального буферного элемента с мультиплексируемыми входами, позволяющего осуществлять выбор одного из двух входных тактовых сигналов в кристаллах программируемой логики семейства Spartan-3. Основой этого шаблона является оператор создания экземпляра библиотечного компонента *BUFGMUX*:

```
-- BUFGMUX : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (BUFGMUX_inst) and/or the port
declarations
-- code : after the "=" assignment may be changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <-----Cut code below this line and paste into the architecture
body----->
--
-- BUFGMUX: Global Clock Buffer 2-to-1 MUX
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
BUFGMUX_inst : BUFGMUX
port map (
    O => O, -- Clock MUX output
    I0 => I0, -- Clock0 input
    I1 => I1, -- Clock1 input
    S => S -- Clock select input
);
-- End of BUFGMUX_inst instantiation
```

Система условных обозначений входных и выходных портов, используемая в шаблоне **Global Clock Buffer MUX (BUFGMUX)**, включает в себя следующие идентификаторы:

- O — выход глобального буферного элемента;
- I0 — вход первого тактового сигнала;
- I1 — вход второго тактового сигнала;
- S — вход выбора глобального сигнала синхронизации.

При низком логическом уровне напряжения (логическом нуле) на входе выбора глобального сигнала синхронизации S на выход глобального буферного элемента поступает тактовый сигнал с входа I0. Когда на входе выбора глобального сигнала синхронизации S присутствует высокий логический уровень напряжения (логическая единица), на выход глобального буферного элемента транслируется тактовый сигнал с входа I1. В момент переключения сигнала на входе выбора S выход глобального буферного элемента, формируемого на основе шаблона **Global Clock Buffer MUX (BUFGMUX)**, переводится в состояние низкого логического уровня.

На рис. 56 приведен условный графический образ глобального буферного элемента с мультиплексируемыми входами, для создания описания которого используется шаблон **Global Clock Buffer MUX (BUFGMUX)**.

Global Clock Buffer MUX (BUFGMUX_1) содержит образец описания глобального буферного элемента с мультиплексируемыми входами, предоставляющего возможность выбора одного из двух входных тактовых сигналов в ПЛИС семейства Spartan-3. Этот шаблон выполнен на основе библиотечного примитива **BUFGMUX_1**:

```
-- BUFGMUX_1 : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (BUFGMUX_1_inst) and/or the port
-- declarations
-- code : after the "=" assignment may be changed to properly
-- : reference and connect this function to the design.
-- : All inputs and outputs must be connected.
--
-- Library : In addition to adding the instance declaration, a use
```

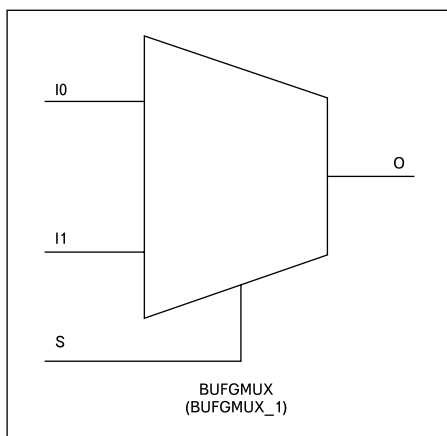


Рис. 56. Условный графический образ глобального буферного элемента с мультиплексируемыми входами, реализуемого на базе ПЛИС семейства Spartan-3

```
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- <---Cut code below this line and paste into the architecture body---->
--
-- BUFGMUX_1: Global Clock Buffer 2-to-1 MUX (inverted select)
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
BUFGMUX_1_inst : BUFGMUX_1
port map (
O => O, -- Clock MUX output
I0 => I0, -- Clock0 input
I1 => I1, -- Clock1 input
S => S -- Clock select input
);
-- End of BUFGMUX_1_inst instantiation
```

В отличие от глобального буферного элемента с мультиплексируемыми входами, формируемого на базе шаблона **Global Clock Buffer MUX (BUFGMUX_1)**, рассмотренного выше, выход глобального буфера, создаваемого с помощью конструкции **Global Clock Buffer MUX (BUFGMUX_1)**, в момент переключения сигнала на входе выбора S устанавливается в состояние высокого логического уровня.

Digital Clock Manager (DCM) включает в себя шаблон описания цифрового модуля управления синхронизацией, представленного в кристаллах программируемой логики семейства Spartan-3. Обобщенная структура модулей DCM, входящих в состав архитектуры ПЛИС указанного семейства, изображена на рис. 57. Основными элементами этой структуры являются следующие блоки:

- схема автоподстройки задержки Delay-Locked Loop (DLL);
- цифровой синтезатор частот Digital Frequency Synthesizer (DFS);
- схема фазового сдвига Phase Shifter (PS);

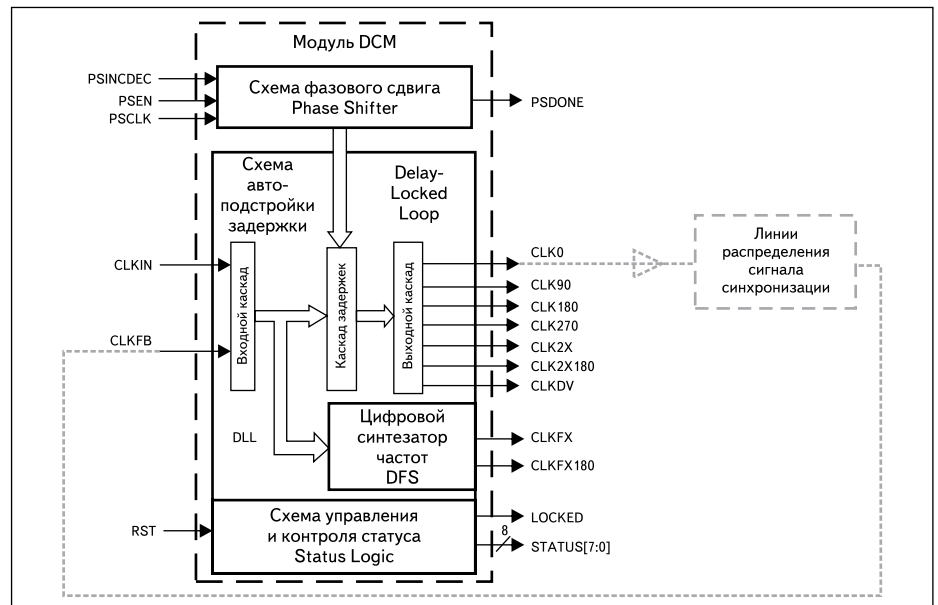


Рис. 57. Обобщенная структура модулей DCM, представленных в составе архитектуры ПЛИС семейства Spartan-3

- схема управления и контроля статуса Status Logic (SL).

Основная функция схемы автоподстройки задержки DLL — это устранение расхождения фронтов тактовых сигналов, возникающих при распространении этих сигналов внутри кристаллов или на печатных платах. Кроме того, данная схема формирует следующие выходные сигналы синхронизации:

- с той же частотой, что и входной тактовый сигнал, и различными значениями фазового сдвига (с нулевым сдвигом, на четверть, половину и три четверти периода);
- с удвоенным значением частоты входного тактового сигнала;
- с удвоенным значением частоты входного тактового сигнала и сдвигом по фазе на половину периода;
- с частотой входного тактового сигнала, поделенной на коэффициент, заданный разработчиком из набора допустимых значений.

Цифровой синтезатор частот DFS предназначен для генерации тактового сигнала с частотой, определяемой разработчиком в пределах допустимого диапазона. Значение частоты выходного сигнала синтезатора вычисляется путем умножения частоты входного тактового сигнала на дробный коэффициент, числитель и знаменатель которого определяются пользователем с помощью соответствующих параметров.

Схема фазового сдвига PS позволяет осуществить с большой точностью сдвиг всех сигналов синхронизации, формируемых модулем DCM, относительно входного тактового сигнала на произвольное значение фазы (постоянное или изменяемое в процессе функционирования блока синхронизации).

Схема управления и контроля статуса SL предназначена для выполнения функций асинхронного сброса и отображения ин-

формации о состоянии цифрового модуля управления синхронизацией в виде сигналов на выходах STATUS и LOCKED.

Основу шаблона *Digital Clock Manager (DCM)* образует оператор создания экземпляра библиотечного компонента DCM:

```
-- DCM : In order to incorporate this function into the design,
-- VHDL : the following instance declaration needs to be placed
-- instance : in the architecture body of the design code. The
-- declaration : instance name (DCM_inst) and/or the port declarations
-- code : after the "=" assignment may be changed to properly
-- : connect this function to the design. All inputs
-- : must be connected.
--
-- Library : In addition to adding the instance declaration, a use
-- declaration : statement for the UNISIM.vcomponents library needs to be
-- for : added before the entity declaration. This library
-- Xilinx : contains the component declarations for all Xilinx
-- primitives : primitives and points to the models that will be used
-- : for simulation.
--
-- Copy the following two statements and paste them before the
-- Entity declaration, unless they already exist.
--
Library UNISIM;
use UNISIM.vcomponents.all;
--
-- -----Cut code below this line and paste into the architecture body-----
--
-- DCM: Digital Clock Manager Circuit
-- Spartan-3
-- Xilinx HDL Language Template, version 11.4
--
DCM_inst : DCM
generic map (
    CLKDV_DIVIDE => 20, -- Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5
    5.0,5.5,6.0,6.5
    -- 7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
    CLKFX_DIVIDE => 1, -- Can be any integer from 1 to 32
    CLKFX_MULTIPLY => 4, -- Can be any integer from 1 to 32
    CLKIN_DIVIDE_BY_2 => FALSE, -- TRUE/FALSE to enable
    CLKIN divide by two feature
    CLKIN_PERIOD => 0.0, -- Specify period of input clock
    CLKOUT_PHASE_SHIFT => "NONE", -- Specify phase shift
    of NONE, FIXED or VARIABLE
    CLK_FEEDBACK => "1X", -- Specify clock feedback of NONE,
    1X or 2X
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", --
    SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
    -- an integer from 0 to 15
    DFS_FREQUENCY_MODE => "LOW", -- HIGH or LOW
    frequency mode for frequency synthesis
    DLL_FREQUENCY_MODE => "LOW", -- HIGH or LOW
    frequency mode for DLL
    DUTY_CYCLE_CORRECTION => TRUE, -- Duty cycle
    correction, TRUE or FALSE
    FACTORY_JF => X"C080", -- FACTORY JF Values
    PHASE_SHIFT => 0, -- Amount of fixed phase shift from
    -255 to 255
    SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST", see
    "Synthesis and Simulation
    -- Design Guide" for details
    STARTUP_WAIT => FALSE) -- Delay configuration DONE
until DCM_LOCK, TRUE/FALSE
port map (
    CLK0 => CLK0, -- 0 degree DCM CLK output
    CLK180 => CLK180, -- 180 degree DCM CLK output
    CLK270 => CLK270, -- 270 degree DCM CLK output
    CLK2X => CLK2X, -- 2X DCM CLK output
    CLK2X180 => CLK2X180, -- 2X, 180 degree DCM CLK out
    CLK90 => CLK90, -- 90 degree DCM CLK output
    CLKDV => CLKDV, -- Divided DCM CLK out (CLKDV_
    DIVIDE)
    CLKFX => CLKFX, -- DCM CLK synthesis out (M/D)
    CLKFX180 => CLKFX180, -- 180 degree CLK synthesis out
    LOCKED => LOCKED, -- DCM LOCK status output
    PSDONE => PSDONE, -- Dynamic phase adjust done output
    STATUS => STATUS, -- 8-bit DCM status bits output
    CLKFB => CLKFB, -- DCM clock feedback
    CLKIN => CLKIN, -- Clock input (from IBUFG, BUFG
    or DCM)
    PSCLK => PSCLK, -- Dynamic phase adjust clock input
    PSEN => PSEN, -- Dynamic phase adjust enable input
    PSINCDEC => PSINCDEC, -- Dynamic phase adjust
    increment/decrement
    RST => RST -- DCM asynchronous reset input
);
-- End of DCM_inst instantiation
```

Для установки требуемых режимов работы всех функциональных блоков создаваемого экземпляра модуля DCM в представленном

шаблоне предусмотрена следующая совокупность параметров настройки:

- CLKDV_DIVIDE — определяет значение коэффициента деления частоты на выходе CLKDV схемы автоподстройки задержки DLL (по умолчанию предлагается значение 2.0, соответствующее двукратному делению частоты входного сигнала).
- CLKFX_DIVIDE — устанавливает значение знаменателя дробного коэффициента, используемого для вычисления частоты выходного сигнала, формируемого цифровым синтезатором частот DFS.
- CLKFX_MULTIPLY — задает значение числителя коэффициента умножения частоты входного тактового сигнала, определяющего характеристики сигнала синхронизации, генерируемого на выходе цифрового синтезатора частот.
- CLKIN_DIVIDE_BY_2 — предоставляет возможность включения режима двукратного деления частоты входного тактового сигнала в модуле DCM.
- CLKIN_PERIOD — определяет значение периода входного сигнала синхронизации.
- CLKOUT_PHASE_SHIFT — указывает необходимость выполнения и тип фазового сдвига формируемых сигналов синхронизации (по умолчанию предлагается значение NONE, при котором генерация выходных тактовых сигналов осуществляется без фазового сдвига).
- CLK_FEEDBACK — задает тип сигнала, подаваемого на вход обратной связи модуля DCM.
- DESKEW_ADJUST — устанавливает длительность выравнивающей задержки выходных тактовых сигналов, формируемых модулем DCM, относительно сигнала синхронизации, поступающего на входной контакт ПЛИС.
- DFS_FREQUENCY_MODE — указывает режим работы цифрового синтезатора частот DFS.
- DLL_FREQUENCY_MODE — задает режим работы схемы автоподстройки задержки DLL.
- DUTY_CYCLE_CORRECTION — определяет необходимость коррекции скважности тактовых сигналов, формируемых на выходах CLK0, CLK90, CLK180 и CLK270.
- FACTORY_JF — задает значение джиттера выходного сигнала, формируемого цифровым синтезатором частот.
- PHASE_SHIFT — устанавливает величину фиксированного фазового сдвига.
- SIM_MODE — позволяет выбрать режим моделирования формируемого экземпляра модуля DCM (по умолчанию используется значение "SAFE", соответствующее безопасному режиму моделирования).
- STARTUP_WAIT — управляет включением паузы в процессе конфигурирования ПЛИС до момента нормального захвата следящей системы схемы автоподстройки задержек DLL.

В описании интерфейса библиотечного примитива DCM используется следующая система условных обозначений входных и выходных портов:

- CLK0 — выход сигнала с частотой входного сигнала синхронизации и нулевым сдвигом фазы;
 - CLK90 — выход сигнала с частотой входного сигнала синхронизации и сдвигом фазы на четверть периода;
 - CLK180 — выход сигнала с частотой входного сигнала синхронизации и сдвигом фазы на половину периода;
 - CLK270 — выход сигнала с частотой входного сигнала синхронизации и сдвигом фазы на три четверти периода;
 - CLK2X — выход сигнала с удвоенной частотой входного сигнала синхронизации и нулевым сдвигом фазы;
 - CLK2X180 — выход сигнала с удвоенной частотой входного сигнала синхронизации и сдвигом фазы на половину периода;
 - CLKDV — выход тактового сигнала, синтезируемого схемой автоподстройки задержки DLL с частотой, определяемой коэффициентом деления частоты исходного сигнала синхронизации;
 - CLKFX — выход сигнала, вырабатываемого цифровым синтезатором частот DFS, с нулевым сдвигом фазы и частотой, определяемой разработчиком;
 - CLKFX180 — выход сигнала, формируемого цифровым синтезатором частот, со сдвигом фазы на половину периода и частотой, определяемой разработчиком;
 - LOCKED — выход сигнала, информирующего о текущем соотношении фаз входного и выходного сигналов схемы автоподстройки задержек DLL создаваемого экземпляра модуля DCM;
 - PSDONE — выход сигнала завершения операции фазового сдвига;
 - STATUS — выходная шина состояния создаваемого экземпляра цифрового модуля управления синхронизацией;
 - CLKFB — вход сигнала обратной связи;
 - CLKIN — вход исходного тактового сигнала;
 - PSCLK — вход тактового сигнала схемы фазового сдвига PS;
 - PSEN — вход разрешения выполнения операции изменения фазового сдвига;
 - PSINCDEC — вход выбора режима изменения фазового сдвига (инкрементный или декрементный);
 - RST — вход асинхронного сброса.
- После включения текста шаблона *Digital Clock Manager (DCM)* в состав создаваемого модуля VHDL-описания нужно, прежде всего, заменить *DCM_inst* идентификатором соответствующего экземпляра цифрового модуля управления синхронизацией. Затем необходимо указать значение периода входного сигнала синхронизации, используя параметр настройки CLKIN_PERIOD. Далее следует с помощью настраиваемого параметра CLK_

FEEDBACK выбрать тип сигнала, подаваемого на вход обратной связи модуля DCM. По умолчанию для этого параметра установлено значение 1X, при котором на вход обратной связи должен поступать сигнал, формируемый на выходе CLK0 с той же частотой, что и исходный тактовый сигнал. Если на вход обратной связи должен подаваться сигнал с двукратным значением частоты, генерируемый на выходе CLK2X, то данному параметру необходимо присвоить значение 2X. В том случае, когда цепь обратной связи в цифровом модуле управления синхронизацией не используется, нужно для параметра CLK_FEEDBACK указать значение None.

Для выбора требуемого режима функционирования схемы DLL нужно задать соответствующее значение параметра DLL_FREQUENCY_MODE. По умолчанию данному параметру присвоено значение "LOW", соответствующее низкочастотному режиму работы схемы автоподстройки задержки. В этом режиме на всех выходах схемы DLL осуществляется формирование сигналов, период которых соответствует диапазону низких и средних частот. Чтобы установить высокочастотный режим работы схемы DLL [19, 20], следует для параметра DLL_FREQUENCY_MODE указать значение "HIGH". В этом случае разрешается генерация тактовых сигналов с максимально допустимыми значениями частоты на выходах CLK0, CLK180 и CLKDV. Остальные выходы схемы автоподстройки задержки в высокочастотном режиме недоступны.

Если в разрабатываемом устройстве предполагается использование сигнала, формируемого делителем частоты, входящим в состав схемы автоподстройки задержки DLL, то с помощью параметра CLKDV_DIVIDE нужно задать требуемое значение коэффициента деления частоты. Указанный параметр для модулей DCM кристаллов программируемой логики семейства Spartan-3 может принимать одно из следующих значений [19, 20]: 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 9.0, 10.0, 12.0, 13.0, 14.0, 15.0 и 16.0.

В случае необходимости формирования тактовых сигналов с фазовым сдвигом по отношению к исходному сигналу синхронизации нужно задать соответствующее значение для параметра CLKOUT_PHASE_SHIFT, при котором будет задействована схема фазового сдвига PS. Чтобы установить режим фиксированного фазового сдвига для этой схемы, следует указанному параметру присвоить значение FIXED. Величина и направление фазового сдвига при этом определяется с помощью параметра PHASE_SHIFT, значение которого может выбираться в диапазоне от -255 до 255. При этом фазовый сдвиг T_{PS} , вносимый схемой PS, вычисляется по следующей формуле.

$$T_{PS} = \left(\frac{PHASE_SHIFT}{256} \right) \times CLKIN_PERIOD. \quad (1)$$

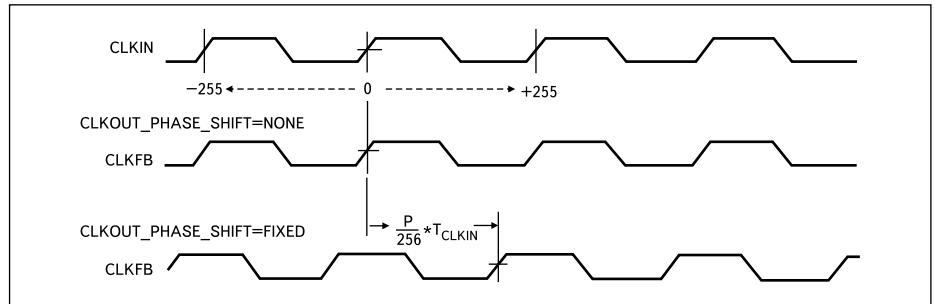


Рис. 58. Временные диаграммы, соответствующие фиксированному режиму работы схемы фазового сдвига PS модуля DCM в ПЛИС семейства Spartan-3

Положительное значение параметра PHASE_SHIFT соответствует запаздыванию фронта сигнала на входе обратной связи CLKFB относительно фронта исходного сигнала синхронизации, поступающего на вход CLKIN. Если параметру PHASE_SHIFT присваивается отрицательное значение, то фронт сигнала на входе обратной связи опережает фронт исходного тактового сигнала. На рис. 58 приведены временные диаграммы, поясняющие формирование тактовых сигналов в модуле DCM ПЛИС семейства Spartan-3 без фазового сдвига и с фиксированным фазовым сдвигом.

Когда необходимо использовать регулируемый режим работы схемы фазового сдвига PS цифрового модуля управления синхронизацией, следует указать для параметра CLKOUT_PHASE_SHIFT значение VARIABLE. В этом режиме для управления изменением фазового сдвига формируемых сигналов задействуются входы PSEN, PSClk, PSINCDEC. В других режимах (без фазового сдвига или с фиксированным сдвигом) эти входы не используются и должны быть подключены к общей шине («земле»). Временные диаграммы, демонстрирующие регулируемый режим работы схемы фазового сдвига PS модуля DCM в ПЛИС семейства Spartan-3, представлены на рис. 59.

Для выбора скважности тактовых сигналов, формируемых на выходах CLK0, CLK90, CLK180 и CLK270 схемы автоподстройки задержки цифрового модуля управления синхронизацией, следует установить соответ-

ствующее значение параметра DUTY_CYCLE_CORRECTION. Если данному параметру присваивается значение FALSE, то скважность указанных сигналов будет совпадать со скважностью исходного сигнала синхронизации, подаваемого на вход CLKIN. На рис. 60 показаны временные диаграммы тактовых сигналов, формируемых на выходах CLK0, CLK90, CLK180 и CLK270 без изменения скважности.

При использовании значения TRUE, предлагаемого по умолчанию для параметра DUTY_CYCLE_CORRECTION, производится коррекция скважности вырабатываемых сигналов синхронизации, в результате которой все указанные выходные тактовые сигналы будут иметь скважность 50%. Временные диаграммы, приведенные на рис. 61, поясняют процесс формирования сигналов на выходах CLK0, CLK90, CLK180 и CLK270 с коррекцией скважности.

Следует обратить внимание на то, что параметр DUTY_CYCLE_CORRECTION не оказывает влияния на скважность сигналов, вырабатываемых схемой автоподстройки задержки DLL на выходах CLK2X, CLK2XH и CLKDV. Эти сигналы всегда имеют скважность 50%, как показано на рис. 62.

Чтобы включить режим двукратного деления частоты входного тактового сигнала, нужно присвоить параметру CLKIN_DIVIDE_BY_2 значение TRUE. Необходимость применения такого режима может возникнуть, например, когда частота входного сигнала синхронизации превышает предельно допустимое значение для схемы автоподстройки задержек DLL.

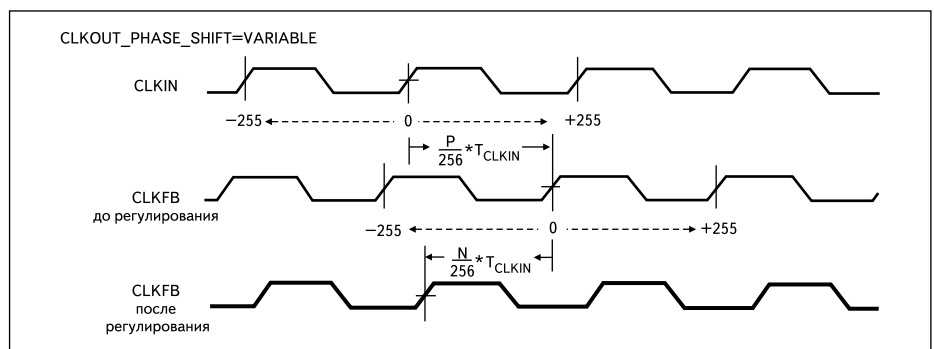


Рис. 59. Временные диаграммы, соответствующие регулируемому режиму работы схемы фазового сдвига PS модуля DCM в ПЛИС семейства Spartan-3

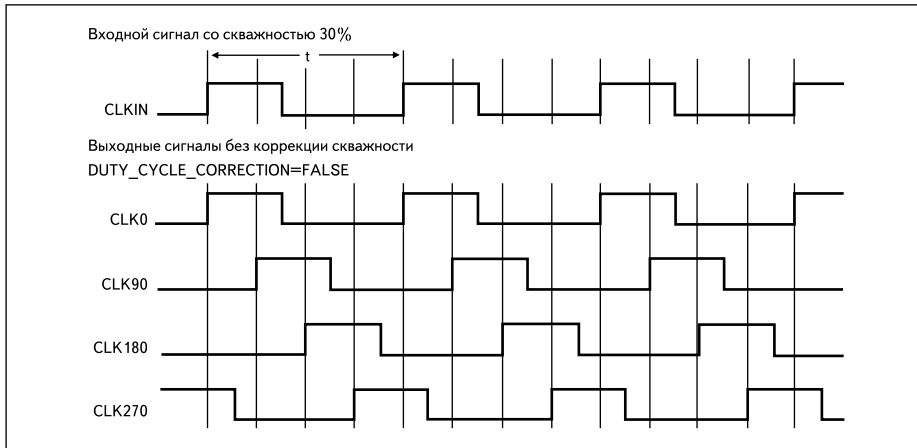


Рис. 60. Временные диаграммы тактовых сигналов, формируемых на выходах CLK0, CLK90, CLK180 и CLK270 без коррекции скважности

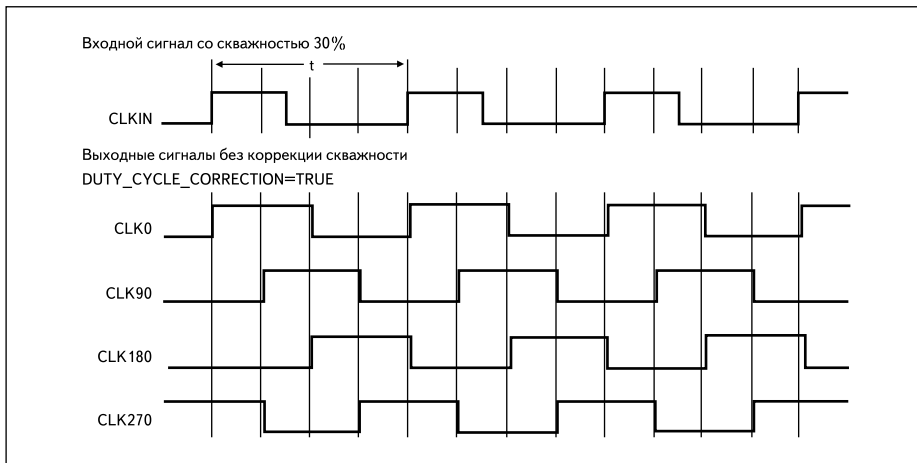


Рис. 61. Временные диаграммы тактовых сигналов, формируемых на выходах CLK0, CLK90, CLK180 и CLK270 с коррекцией скважности

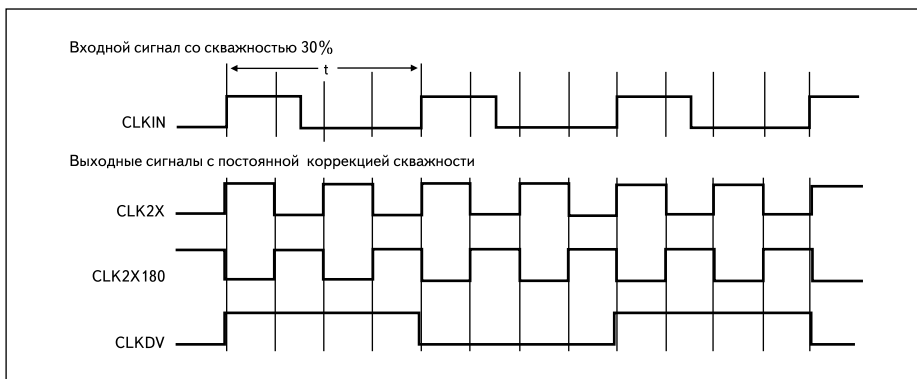


Рис. 62. Временные диаграммы тактовых сигналов, формируемых на выходах CLK2X, CLK2X180 и CLKDV схемы автоподстройки задержки DLL

Значение коэффициента деления частоты исходного сигнала синхронизации, задаваемое параметром CLKFX_DIVIDE, можно выбирать в диапазоне от 1 до 32. Допустимые значения коэффициента умножения частоты входного тактового сигнала находятся в интервале от 2 до 32. Кроме того, нужно с помощью параметра настройки DFS_FREQUENCY_MODE выбрать режим работы цифрового синтезатора частот. По умолчанию данный параметр принимает значение "LOW", соответствующее низкочастотному режиму функционирования DFS. Для включения высокочастотного режима работы цифрового синтезатора частот следует для параметра DFS_FREQUENCY_MODE указать значение "HIGH".

Если необходимо задержать завершение процесса конфигурирования ПЛИС до момента нормального захвата следящей системы схемы автоподстройки задержек DLL, нужно для параметра настройки STARTUP_WAIT указать значение TRUE. По умолчанию для данного параметра предлагается значение FALSE, при котором в процессе конфигурирования кристалла не будет выполняться ожидание вхождения схемы DLL в режим нормального захвата.

Процесс подготовки описания конкретного экземпляра цифрового модуля управления синхронизацией завершается вводом идентификаторов сигналов, подключаемых к соответствующим входным и выходным портам этого компонента.

На рис. 63 показан условный графический образ цифрового модуля управления синхронизацией, формируемого с помощью шаблона Digital Clock Manager (DCM), который предназначен для использования в составе VHDL-

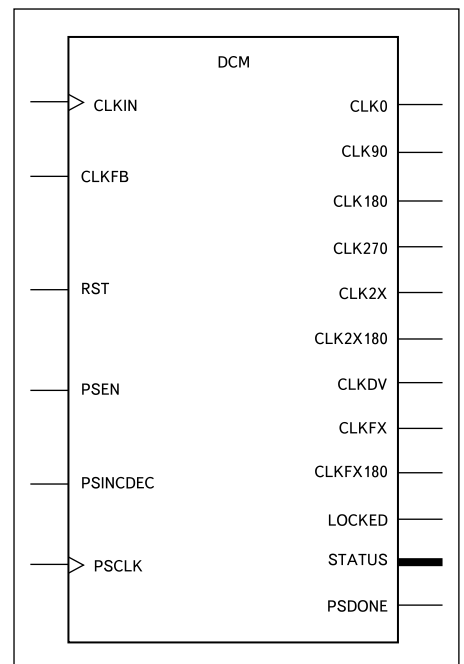


Рис. 63. Условный графический образ цифрового модуля управления синхронизацией DCM

По умолчанию для параметра CLKIN_DIVIDE_BY_2 предлагается значение FALSE, при котором двукратное понижение частоты входного сигнала синхронизации не производится.

Если в проектируемом устройстве необходим тактовый сигнал, формируемый цифровым синтезатором частот DFS, то в первую очередь следует указать требуемые значения параметров CLKFX_DIVIDE и CLKFX_

MULTIPLY. Эти параметры определяют значение частоты синтезируемого сигнала F_{CLKFX} в соответствии с выражением (2).

$$F_{CLKFX} = F_{CLKIN} \cdot (CLKFX_MULTIPLY / CLKFX_DIVIDE), \quad (2)$$

где F_{CLKIN} — частота исходного сигнала синхронизации, поступающего на вход CLKIN.

описаний проектируемых устройств, реализуемых на базе ПЛИС семейства Spartan-3. ■

Продолжение следует.

Литература

1. Кузелин М. О., Кнышев Д. А., Зотов В. Ю. Современные семейства ПЛИС фирмы Xilinx. Справочное пособие. М.: Горячая линия – Телеком, 2004.
2. Бибило П. Н. Основы языка VHDL. М.: Солон-Р, 2000.
3. Бибило П. Н. Синтез логических схем с использованием языка VHDL. М.: Солон-Р, 2002.
4. Уэйкерли Дж. Ф. Проектирование цифровых устройств. Том 1. М.: Постмаркет, 2002.
5. Поляков А. К. Языки VHDL и Verilog в проектировании цифровой аппаратуры. М.: Солон-Пресс, 2003.
6. Зотов В. Инструментальный комплект Spartan-3 Starter Kit для практического освоения методов проектирования встраиваемых микропроцессорных систем на основе ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2005. № 7.
7. 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.
8. Зотов В. Новый инструментальный комплект Spartan-3A Starter Kit для практического освоения методов проектирования и отладки цифровых устройств с аппаратной и программной реализацией операций, реализуемых на основе ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2007. № 9.
9. Зотов В. Новый инструментальный комплект от компании Avnet на основе ПЛИС FPGA семейства Spartan-3A фирмы Xilinx // Компоненты и технологии. 2008. № 8.
10. Зотов В. Инструментальный модуль компании Avnet для отладки проектов встраиваемых систем, разрабатываемых на базе нового семейства ПЛИС FPGA фирмы Xilinx Virtex-5 FXT // Компоненты и технологии. 2008. № 9.
11. Зотов В. Особенности архитектуры нового поколения высокопроизводительных ПЛИС FPGA фирмы Xilinx серии Virtex-6 // Компоненты и технологии. 2009. № 8.
12. Зотов В. Особенности архитектуры нового поколения ПЛИС FPGA фирмы Xilinx серии Spartan-6 // Компоненты и технологии. 2009. № 9.
13. Зотов В. Новое семейство высокопроизводительных ПЛИС с архитектурой FPGA фирмы Xilinx Virtex-6 HXT // Компоненты и технологии. 2010. № 1.
14. IEEE Standard VHDL Language Reference Manual — IEEE Std 1076–2002.
15. Суворова Е. А., Шейнин Ю. Е. Проектирование цифровых систем на VHDL. СПб.: БХВ-Петербург, 2003.
16. Сергиенко А. М. VHDL для проектирования вычислительных устройств. К.: ЧП «Корнейчук», ООО «ТИД «ДС», 2003.
17. Зотов В. Проектирование цифровых устройств, реализуемых на базе ПЛИС FPGA фирмы Xilinx, с использованием средств CORE Generator // Компоненты и технологии. 2006. № 12. 2007. № 1.
18. Зотов В. Разработка компонентов устройств цифровой обработки сигналов, реализуемых на базе аппаратных модулей DSP48E в ПЛИС FPGA серии Virtex-5, с помощью «мастера» Architecture Wizard САПР серии Xilinx ISE // Компоненты и технологии. 2008. № 12. 2009. № 1, 4–7.
19. Spartan-3 Generation FPGA User Guide. Xilinx, 2009.
20. Spartan-3 FPGA Family: Complete Data Sheet. Xilinx, 2009.