

Продолжение. Начало в № 12`2008

## Разработка компонентов устройств ЦОС, реализуемых на базе аппаратных модулей DSP48E в ПЛИС FPGA серии Virtex-5, с помощью «мастера» Architecture Wizard САПР серии Xilinx ISE

Валерий ЗОТОВ  
walery@km.ru

В пятой части статьи продолжается изучение процесса формирования компонентов высокопроизводительных устройств ЦОС, реализуемых на базе аппаратных секций DSP48E в ПЛИС с архитектурой FPGA [1] семейств Virtex-5 LX, Virtex-5 LXT, Virtex-5 SXT, Virtex-5 FXT и Virtex-5 TXT, выполняемого с помощью «мастера» Architecture Wizard САПР серии Xilinx ISE [3, 4]. Основное внимание уделяется рассмотрению процедуры подготовки описаний высокоскоростных элементов, выполняющих операции умножения с накоплением (вычитанием). Для каждого варианта операции, выполняемой аккумулятором в указанных элементах, приводятся примеры сформированных описаний, предназначенных для реализации на основе кристаллов серии Virtex-5 [2, 5–7].

### Формирование описаний элементов, осуществляющих операции умножения с накоплением (вычитанием), реализуемых на базе аппаратных секций DSP48E в ПЛИС FPGA серии Virtex-5, с помощью «мастера» Architecture Wizard САПР серии Xilinx ISE

Элементы, выполняющие операции перемножения значений входных данных с последующим накоплением или вычитанием полученных результатов, находят широкое применение в составе цифровых устройств различного назначения. Наиболее часто данные компоненты востребованы в процессе проектирования устройств цифровой обработки сигналов и, в первую очередь, при разработке цифровых фильтров. Для быстрой подготовки описаний умножителей-накопителей, реализуемых на базе ПЛИС с архитектурой FPGA предыдущих поколений, разработчикам предлагается параметризованный модуль *Multiply Accumulator v4.0*, который представлен одновременно в составе группы ядер *Math Functions*, предназначенных для реализации математических функций, и группы ядер *Digital Signal Processing*, используемых

в устройствах цифровой обработки сигналов. Процесс формирования описаний рассматриваемых элементов на основе этого ядра с помощью генератора параметризованных модулей CORE Generator [8] был подробно рассмотрен в [9].

Особенности архитектуры аппаратных модулей DSP48E в ПЛИС семейств Virtex-5 LX, Virtex-5 LXT, Virtex-5 SXT, Virtex-5 FXT и Virtex-5 TXT, представленные в первой части настоящей статьи (см. КиТ № 12`2008), позволяют создавать на основе этих секций не только элементы, выполняющие операции умножения с накоплением, с традиционной структурой, но и компоненты с расширенным набором функциональных возможностей. В частности, на базе аппаратных модулей ЦОС в кристаллах серии Virtex-5 могут быть реализованы элементы, в состав которых входит умножитель с трехвходным сумматором или вычитающим устройством. В данном разделе рассматривается процесс формирования описаний компонентов, выполняющих операции умножения с накоплением (вычитанием) полученных результатов, с традиционной структурой, включающей в себя умножитель и аккумулятор, с помощью «мастера» Architecture Wizard.

Если для подготовки описаний элементов, осуществляющих перемножение значений входных данных с последующим вычислением суммы или разности произведения и содержимого аккумулятора, выбран автономный режим работы «мастера» Architecture Wizard, то в списке поддерживаемых типов элементов, который представлен в диалоговой панели с заголовком *Selection*, приведенной на рис. 6 (см. КиТ № 12`2008, стр. 34), нужно указать вариант *Multiplier and Adder/Accumulator* (МАСС). При формировании описаний элементов, выполняющих операции умножения с накоплением (вычитанием), непосредственно в среде управляющей оболочки САПР серии Xilinx ISE Навигатора проекта (*Project Navigator*), в диалоговой панели выбора типа параметризованного модуля *Select IP* (рис. 8, КиТ № 12`2008, стр. 35) следует выделить строку *Multiply Accumulator v9.1i*.

По окончании процедуры определения типа генерируемого компонента на экране появляется стартовая диалоговая панель «мастера» настройки параметров формируемого элемента, осуществляющего операцию умножения с накоплением (вычитанием). Вид этой диалоговой панели, которая имеет заголовок *Input/Output Data Setup – МАСС*, показан на рис. 29.

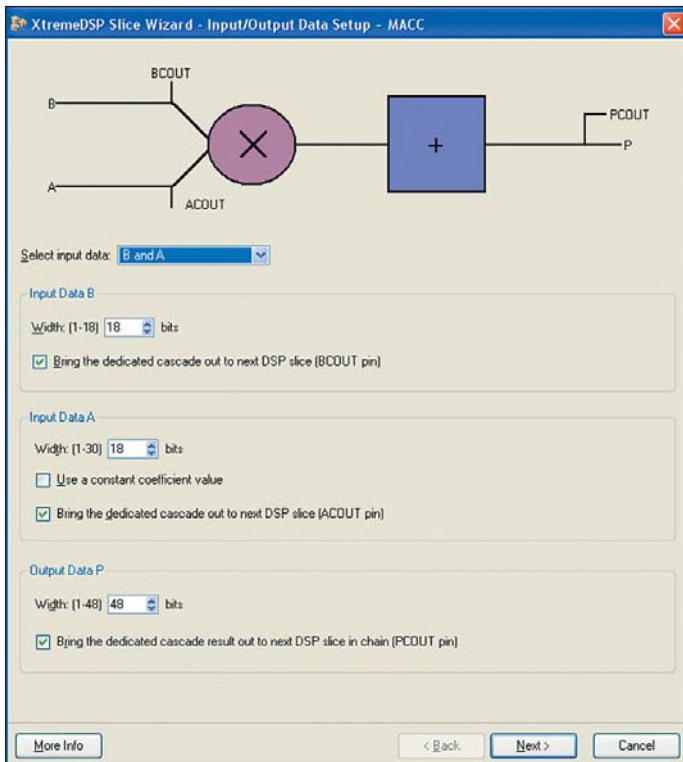


Рис. 29. Вид стартовой диалоговой панели — Input/Output Data Setup — MACC

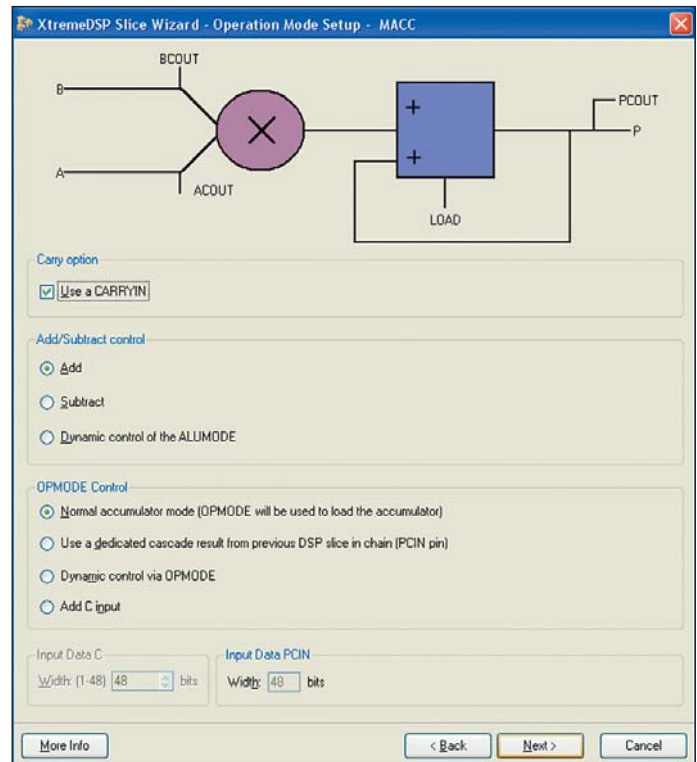


Рис. 30. Вид второй диалоговой панели — Operation Mode Setup — MACC

В стартовой диалоговой панели, прежде всего, необходимо выбрать входные шины данных аппаратной секции DSP48E, совокупности сигналов которых будут определять значения сомножителей в создаваемом элементе, выполняющем операции умножения с последующим накоплением или вычитанием полученных результатов. В качестве входных информационных портов в формируемом компоненте могут использоваться шины данных A, B, ACIN и BCIN в тех же сочетаниях, что и при формировании описаний умножителей, процесс подготовки которых был рассмотрен в предыдущей части статьи (см. Кит № 5'2009). Требуемая комбинация перечисленных шин данных указывается в поле выбора *Select input data*. Если необходимо подготовить описание элемента, осуществляющего операции умножения входных значений на постоянный коэффициент с последующим накоплением (вычитанием), то в этом поле выбора следует указать вариант *B and A* или *BCIN and A*.

Далее нужно определить разрядность входных информационных портов формируемого элемента. Количество разрядов для шин данных A, ACIN и B указывается в полях редактирования *Width*, которые представлены во встроенных панелях *Input Data A*, *Input Data ACIN* и *Input Data B* соответственно. Значения этих параметров не должны выходить за границы допустимых диапазонов разрядности соответствующих входных шин данных аппаратной секции DSP48E. При использовании шин данных A и ACIN в качестве входных информационных портов значения

разрядности этих портов можно устанавливать в диапазоне от 1 до 30 бит. Количество разрядов входного порта, выполненного на основе шины данных B, можно выбирать в пределах от 1 до 18. В том случае, когда одним из входных информационных портов создаваемого элемента, осуществляющего операции умножения с накоплением (вычитанием), является шина данных BCIN, разрядность этого порта автоматически устанавливается равной максимальному значению, которое составляет 18 двоичных разрядов.

Если формируемый элемент предназначен для выполнения операции умножения входных значений на константу с последующим накоплением или вычитанием полученных произведений, то в поле редактирования *Width*, расположенном во встроенной панели *Input Data A*, следует определить разрядность представления значения постоянного коэффициента. Потом нужно установить индикатор *Use a constant coefficient value*, который находится в этой же встроенной панели, в состояние «Включено». При этом справа от указанного индикатора состояния появляется поле редактирования, предназначенное для ввода значения постоянного коэффициента. Процедура определения значения константы, включая формат представления и допустимый диапазон возможных значений, была подробно рассмотрена в разделе, посвященном подготовке описаний умножителей.

Затем следует указать требуемое значение разрядности выходной шины данных формируемого элемента в поле редактирования *Width*, которое расположено во встроенной

панели *Output Data P* (рис. 29). Допустимый диапазон возможных значений этого параметра находится в пределах от 1 до 48 двоичных разрядов. По умолчанию в этом поле редактирования предлагается максимально-возможное значение из указанного диапазона.

Если формируемый элемент предполагается использовать в составе каскадных соединений аппаратных модулей ЦОС, то в его интерфейс целесообразно добавить дополнительные выходные шины ACOUT, BCOUT и PCOUT, которые предназначены для сопряжения с входными шинами ACIN, BCIN и PCIN следующей секции DSP48E. Для этой цели необходимо установить во включенное состояние индикаторы *Bring the dedicated cascade out to next DSP slice (ACOUT pin)*, *Bring the dedicated cascade out to next DSP slice (BCOUT pin)* и *Bring the dedicated cascade result out to next DSP Slice in chain (PCOUT pin)*, которые расположены во встроенных панелях *Input Data A (Input Data ACIN)*, *Input Data B* и *Output Data P* соответственно (рис. 29).

После выбора требуемой комбинации входных и выходных информационных шин данных для генерируемого элемента и определения их разрядности нужно перейти ко второй диалоговой панели «мастера» настройки параметров элементов, осуществляющих операции умножения с накоплением (вычитанием), предназначенных для реализации на базе аппаратных модулей DSP48E в ПЛИС серии Virtex-5. Эта диалоговая панель имеет заголовок *Operation Mode Setup – MACC* и предназначена для выбора структуры создаваемого элемента и типа выполняемой

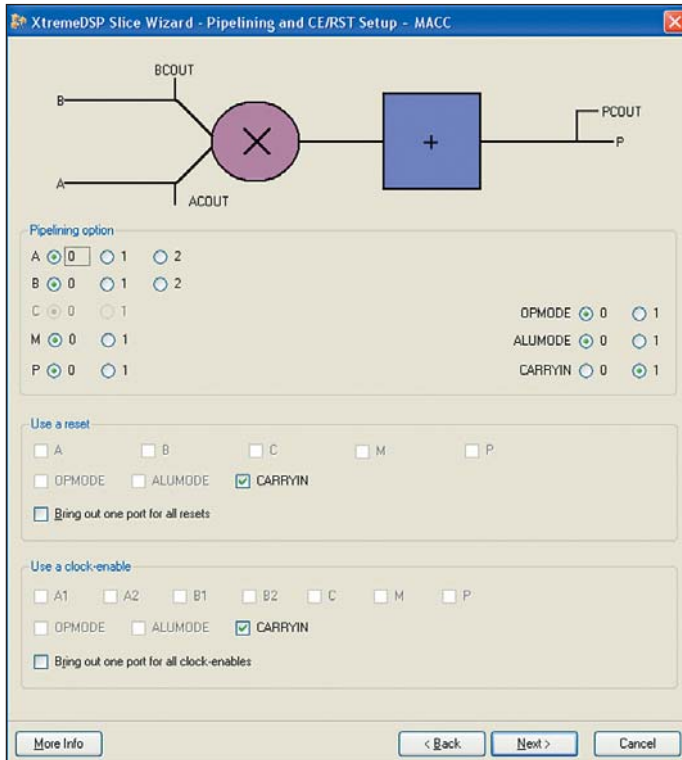


Рис. 31. Вид третьей диалоговой панели — Pipelining and CE/RST Setup – MACC

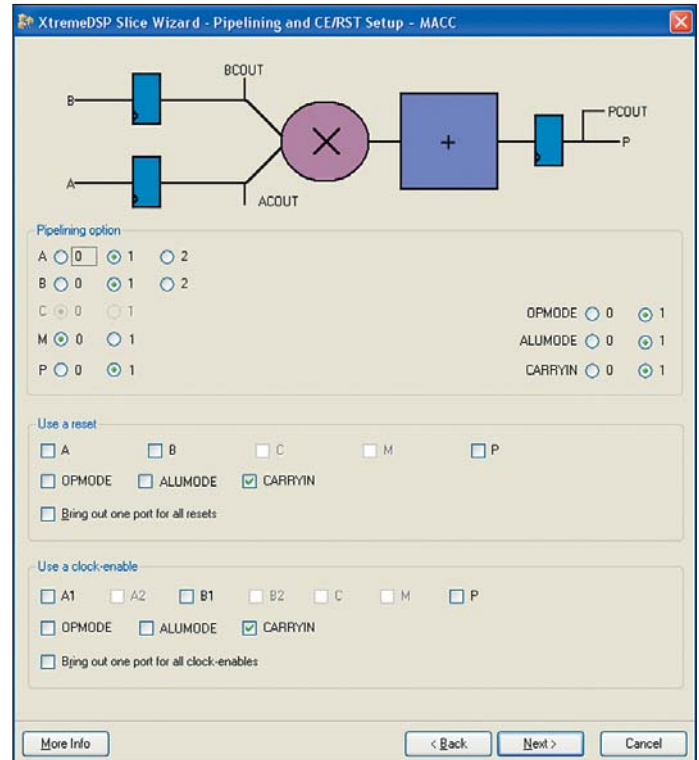


Рис. 32. Вид третьей диалоговой панели Pipelining and CE/RST Setup – MACC при выборе одноступенчатого варианта буферизации входных и выходных данных

операции. Вид второй диалоговой панели в случае подготовки описания умножителей-накопителей с традиционной структурой представлен на рис. 30.

В этой диалоговой панели для выбора типа выполняемой операции в аккумуляторе формируемого элемента следует воспользоваться группой кнопок с зависимой фиксацией *Add/Subtract control*. Чтобы сгенерировать описание элементов, выполняющих операции умножения с накоплением, нужно нажать кнопку *Add*. Для подготовки описания компонента, осуществляющего операции перемножения значений входных данных с последующим вычитанием полученных произведений из содержимого аккумулятора, следует нажать кнопку *Subtract*. Если необходимо сформировать описание элемента, в котором тип операции, выполняемой в аккумуляторе (сложение или вычитание), должен динамически выбираться в процессе функционирования разрабатываемого устройства ЦОС, то нажать нужно кнопку *Dynamic control of the ALUMODE*.

После определения типа выполняемой операции, осуществляемой в аккумуляторе создаваемого элемента, следует с помощью индикатора состояния *Use a CARRYIN*, находящегося во встроенной панели *Carry option* (рис. 30), указать необходимость включения порта сигнала входного переноса в состав интерфейса этого компонента. Если при выполнении операций в генерируемом элементе должно учитываться значение внешнего сигнала входного переноса, то данный ин-

дикатор следует установить во включенное состояние. В противном случае индикатор *Use a CARRYIN* целесообразно переключить в состояние «Выключено».

Выбор структуры создаваемого элемента осуществляется с помощью группы кнопок с зависимой фиксацией, которые представлены во встроенной панели *OPMODE Control* (рис. 30). Каждая из этих кнопок предназначена для выбора соответствующей конфигурации и источников входных данных арифметическо-логического блока аппаратной секции DSP48E, используемой для реализации формируемого элемента. Чтобы сгенерировать описание элемента, выполняющего операции умножения с накоплением (вычитанием), с традиционной структурой, включающей в себя умножитель и обычный аккумулятор, нужно нажать кнопку *Normal accumulator mode (OPMODE will be used to load the accumulator)*. Остальные кнопки, расположенные во встроенной панели *OPMODE Control*, используются для подготовки описаний элементов с расширенным набором функциональных возможностей. Назначение этих кнопок будет рассмотрено в следующей части статьи.

Далее нужно открыть очередную, третью, диалоговую панель «мастера» настройки параметров элементов, выполняющих операции умножения с накоплением (вычитанием), предназначенных для реализации на базе аппаратных модулей DSP48E в ПЛИС семейств Virtex-5 LX, Virtex-5 LXT, Virtex-5 SXT, Virtex-5 FXT и Virtex-5 TXT. Данная диалоговая панель, имеющая заголовок *Pipelining and*

*CE/RST Setup – MACC*, позволяет включить в состав структуры формируемого элемента необходимые входные, выходные и буферные элементы. На рис. 31 приведен вид указанной диалоговой панели, соответствующий варианту архитектуры генерируемого элемента, в котором не применяется конвейерная организация обработки входных и выходных данных.

Чтобы задействовать входные регистры в создаваемом элементе, нужно воспользоваться двумя группами кнопок с зависимой фиксацией A и B, расположенными во встроенной панели *Pipelining option*. При этом разработчику предоставляется возможность выбора одно- или двухступенчатой схемы буферизации входных данных. Для установки одного буферного регистра на каждой из входных информационных шин данных следует нажать кнопку 1. После этого диалоговая панель *Pipelining and CE/RST Setup – MACC* автоматически приобретает вид, изображенный на рис. 32.

Кроме входных информационных шин данных, буферные регистры также могут быть установлены на шинах управления и цепи внешнего сигнала входного переноса. Для применения буферных регистров на входах шины управления выбором функции, выполняемой арифметическо-логическим блоком используемой аппаратной секции DSP48E, нужно нажать кнопку 1, входящую в состав группы *ALUMODE*, которая также доступна во встроенной панели *Pipelining option* (рис. 32). Чтобы включить в состав струк-

туры формируемых элементов, выполняющих операции умножения с накоплением (вычитанием), регистры на входах шины управления выбором источников входных данных для арифметико-логического блока аппаратного модуля DSP48E, следует аналогичным образом воспользоваться группой кнопок с зависимой фиксацией *OPMODE*. Необходимость применения буферного регистра в цепи внешнего сигнала входного переноса определяется с помощью группы кнопок *CARRYIN*.

Для реализации двухступенчатого варианта буферизации входных данных в генерируемом элементе необходимо в группах кнопок A и B нажать кнопки 2. В этом случае диалоговая панель *Pipelining and CE/RST Setup – MACC* автоматически преобразуется к виду, представленному на рис. 33.

Чтобы использовать в формируемом элементе буферный регистр, установленный на выходе умножителя, следует в группе кнопок с зависимой фиксацией *M* нажать кнопку 1, как показано на рис. 33. Включение выходного регистра в состав структуры создаваемого элемента, осуществляющего операции умножения с последующим накоплением (вычитанием) полученных произведений, осуществляется таким же образом с помощью группы кнопок с зависимой фиксацией *P*.

После определения совокупности входных, выходных и буферных регистров, применяемых в архитектуре генерируемого элемента, при необходимости нужно указать входы управления для каждого из этих регистров. Для этой цели следует воспользоваться соответствующими индикаторами состояния, которые становятся доступными во встроенных панелях *Use a reset* и *Use a clock-enable* (рис. 32, 33) после включения одноименного регистра в состав структуры формируемого компонента. Чтобы задействовать вход сброса в каком-либо из применяемых регистров и добавить его в состав интерфейса создаваемого элемента, необходимо установить соответствующий индикатор во встроенной панели *Use a reset* в состояние «Включено». Таким же образом, с помощью одноименных индикаторов состояния, расположенных во встроенной панели *Use a clock-enable*, вводятся в состав интерфейса генерируемого компонента входы сигнала разрешения синхронизации для каждого используемого регистра. Для создания общих входов сброса и разрешения синхронизации для всех применяемых входных, выходных и буферных регистров нужно перевести во включенное состояние индикаторы *Bring out one port for all resets* и *Bring out one port for all clock-enables* соответственно (рис. 32, 33).

Далее следует перейти к заключительной информационной панели «мастера» настройки параметров элементов, выполняющих операции умножения с накоплением (вычитанием), которая имеет заголовок *Summary – MACC*. После контроля и возможной коррекции ус-

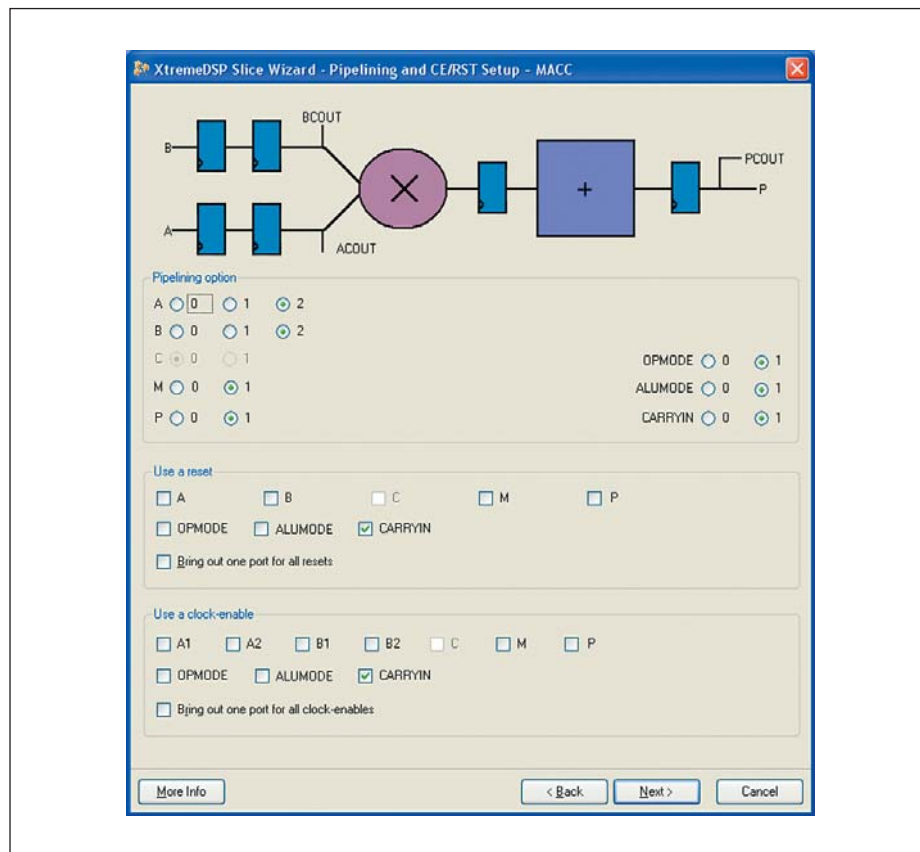


Рис. 33. Вид третьей диалоговой панели *Pipelining and CE/RST Setup – MACC* при выборе двухступенчатого варианта буферизации входных данных

тановленных значений основных параметров формируемого компонента можно активизировать процесс автоматической генерации его описания. В последующих разделах приводятся примеры описаний элементов, осуществляющих операции умножения в сочетании с различными операциями аккумулятора и предназначенных для реализации на базе аппаратных секций DSP48E в ПЛИС FPGA семейств Virtex-5 LX, Virtex-5 LXT, Virtex-5 SXT, Virtex-5 FXT и Virtex-5 TXT.

### Пример описания элемента, выполняющего операции умножения с накоплением, сформированного с помощью «мастера» Architecture Wizard для реализации на базе аппаратных модулей DSP48E в ПЛИС FPGA серии Virtex-5

Подготовка высокопроизводительных компонентов, выполняющих операции умножения с накоплением, с помощью «мастера» Architecture Wizard для последующей реализации на основе аппаратных секций DSP48E в ПЛИС с архитектурой FPGA семейств Virtex-5 LX, Virtex-5 LXT, Virtex-5 SXT, Virtex-5 FXT и Virtex-5 TXT продемонстрирована на примере элемента *macc\_12\_16\_const*. Этот элемент предназначен для вычисления произведения значений входных данных на

постоянный коэффициент с последующим накоплением результата. Операция сложения в рассматриваемом элементе выполняется с учетом значения внешнего сигнала входного переноса. Компоненты такого типа часто используются в процессе разработки устройств цифровой фильтрации, обнаружения импульсных сигналов и других узлов ЦОС. В элементе *macc\_12\_16\_const* в качестве входного информационного порта данных используется шина B аппаратной секции DSP48E. Входные значения представлены в виде 12-разрядного двоичного кода, а постоянный коэффициент — в 16-разрядном двоичном формате. Результат выполнения операций умножения с накоплением отображается на выходной шине данных сгенерированного элемента в форме 30-разрядного двоичного значения.

Структура элемента *macc\_12\_16\_const* выполнена с применением двухступенчатой организации конвейерной обработки входных и выходных данных. В состав интерфейса сформированного элемента включены отдельные входы сброса и разрешения синхронизации для используемых буферных регистров, а также дополнительные выходные шины ACOUT, BCOUT и PCOUT, которые предоставляют возможность каскадного соединения нескольких компонентов. Текст VHDL-описания элемента *macc\_12\_16\_const* выглядит следующим образом:

```
--Command: xaw2vhdl-st D:\PRJ\macc_12_16_const.xaw D:\PRJ\
macc_12_16_const
--Design Name: macc_12_16_const
--Device: xc5vfx50t-ff1136-3
--
-- Module macc_12_16_const
-- Generated by Xilinx Architecture Wizard
-- Written for synthesis tool: XST
--
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
library UNISIM;
use UNISIM.vcomponents.ALL;
--
entity macc_12_16_const is
port (
    B_IN : in std_logic_vector (11 downto 0);
    CARRYIN_IN : in std_logic;
    CE1_IN : in std_logic;
    CE2_IN : in std_logic;
    CE1_IN : in std_logic;
    CE2_IN : in std_logic;
    CECARRYIN_IN : in std_logic;
    CECTRL_IN : in std_logic;
    CEMULTCARRYIN_IN : in std_logic;
    CEM_IN : in std_logic;
    CEP_IN : in std_logic;
    CLK_IN : in std_logic;
    LOAD_IN : in std_logic;
    RSTALLCARRYIN_IN : in std_logic;
    RSTA_IN : in std_logic;
    RSTB_IN : in std_logic;
    RSTCTRL_IN : in std_logic;
    RSTM_IN : in std_logic;
    RSTP_IN : in std_logic;
    ACOUT_OUT : out std_logic_vector (29 downto 0);
    BCOUT_OUT : out std_logic_vector (17 downto 0);
    PCOUT_OUT : out std_logic_vector (47 downto 0);
    P_OUT : out std_logic_vector (29 downto 0)
);
end macc_12_16_const;
--
architecture BEHAVIORAL of macc_12_16_const is
    signal GND_A : std_logic;
    signal GND_ALUMODE : std_logic;
    signal GND_BUS_3 : std_logic_vector (2 downto 0);
    signal GND_BUS_18 : std_logic_vector (17 downto 0);
    signal GND_BUS_30 : std_logic_vector (29 downto 0);
    signal GND_BUS_48 : std_logic_vector (47 downto 0);
    signal GND_OPMODE : std_logic;
    signal P_float : std_logic_vector (17 downto 0);
    signal VCC_A : std_logic;
    signal VCC_OPMODE : std_logic;
begin
    GND_A <= '0';
    GND_ALUMODE <= '0';
    GND_BUS_3(2 downto 0) <= «000»;
    GND_BUS_18(17 downto 0) <= «00000000000000000000»;
    GND_BUS_30(29 downto 0) <= «00000000000000000000000000000000»;
    GND_BUS_48(47 downto 0) <=
        «0000000000000000000000000000000000000000000000000000000»;
    GND_OPMODE <= '0';
    VCC_A <= '1';
    VCC_OPMODE <= '1';
    DSP48E_INST : DSP48E
    generic map(
        ACASCREG => 1,
        ALUMODEREG => 0,
        AREG => 2,
        AUTORESET_PATTERN_DETECT => FALSE,
        AUTORESET_PATTERN_DETECT_OPTINV => «MATCH»,
        A_INPUT => «DIRECT»,
        BCASCREG => 1,
        BREG => 2,
        B_INPUT => «DIRECT»,
        CARRYINREG => 1,
        CARRYINSELREG => 0,
        CREG => 0,
        MASK => x«3FFFFFFF»,
        MREG => 1,
        MULTCARRYINREG => 1,
        OPMODEREG => 1,
        PATTERN => x«000000000000»,
        PREG => 1,
        SEL_MASK => «MASK»,
        SEL_PATTERN => «PATTERN»,
        SEL_ROUNDING_MASK => «SEL_MASK»,
        USE_MULT => «MULT_S»,
        USE_PATTERN_DETECT => «NO_PATDET»,
        USE_SIMD => «ONE48»
    )
    port map (
        A(29)=>GND_A,
        A(28)=>GND_A,
        A(27)=>GND_A,
        A(26)=>GND_A,
        A(25)=>GND_A,
        A(24)=>GND_A,
```

```

        A(23)=>GND_A,
        A(22)=>GND_A,
        A(21)=>GND_A,
        A(20)=>GND_A,
        A(19)=>GND_A,
        A(18)=>GND_A,
        A(17)=>GND_A,
        A(16)=>GND_A,
        A(15)=>GND_A,
        A(14)=>GND_A,
        A(13)=>GND_A,
        A(12)=>GND_A,
        A(11)=>GND_A,
        A(10)=>GND_A,
        A(9)=>GND_A,
        A(8)=>GND_A,
        A(7)=>GND_A,
        A(6)=>GND_A,
        A(5)=>VCC_A,
        A(4)=>VCC_A,
        A(3)=>GND_A,
        A(2)=>VCC_A,
        A(1)=>VCC_A,
        A(0)=>VCC_A,
        ACIN(29 downto 0)=>GND_BUS_30(29 downto 0),
        ALUMODE(3)=>GND_ALUMODE,
        ALUMODE(2)=>GND_ALUMODE,
        ALUMODE(1)=>GND_ALUMODE,
        ALUMODE(0)=>GND_ALUMODE,
        B(17)=>B_IN(11),
        B(16)=>B_IN(11),
        B(15)=>B_IN(11),
        B(14)=>B_IN(11),
        B(13)=>B_IN(11),
        B(12)=>B_IN(11),
        B(11 downto 0)=>B_IN(11 downto 0),
        BCIN(17 downto 0)=>GND_BUS_18(17 downto 0),
        C(47 downto 0)=>GND_BUS_48(47 downto 0),
        CARRYASCIN=>GND_A,
        CARRYIN=>CARRYIN_IN,
        CARRYINSEL(2 downto 0)=>GND_BUS_3(2 downto 0),
        CEALUMODE=>VCC_OPMODE,
        CE1=>CE1_IN,
        CE2=>CE2_IN,
        CE1=>CE1_IN,
        CE2=>CE2_IN,
        CEC=>VCC_OPMODE,
        CECARRYIN=>CECARRYIN_IN,
        CECTRL=>CECTRL_IN,
        CEM=>CEM_IN,
        CEMULTCARRYIN=>CEMULTCARRYIN_IN,
        CEP=>CEP_IN,
        CLK=>CLK_IN,
        MULTSIGNIN=>GND_A,
        OPMODE(6)=>GND_OPMODE,
        OPMODE(5)=>LOAD_IN,
        OPMODE(4)=>GND_OPMODE,
        OPMODE(3)=>GND_OPMODE,
        OPMODE(2)=>VCC_OPMODE,
        OPMODE(1)=>GND_OPMODE,
        OPMODE(0)=>VCC_OPMODE,
        PCIN(47 downto 0)=>GND_BUS_48(47 downto 0),
        RSTA=>RSTA_IN,
        RSTALLCARRYIN=>RSTALLCARRYIN_IN,
        RSTALUMODE=>GND_A,
        RSTB=>RSTB_IN,
        RSTC=>GND_A,
        RSTCTRL=>RSTCTRL_IN,
        RSTM=>RSTM_IN,
        RSTP=>RSTP_IN,
        ACOUT(29 downto 0)=>ACOUT_OUT(29 downto 0),
        BCOUT(17 downto 0)=>BCOUT_OUT(17 downto 0),
        CARRYASCOUT=>open,
        CARRYOUT=>open,
        MULTSIGNOUT=>open,
        OVERFLOW=>open,
        P(47 downto 30)=>P_float(17 downto 0),
        P(29 downto 0)=>P_OUT(29 downto 0),
        PATTERNBDetect=>open,
        PATTERNDETECT=>open,
        PCOUT(47 downto 0)=>PCOUT_OUT(47 downto 0),
        UNDERFLOW=>open
    );
end BEHAVIORAL;
```

Значение константы, на которую умножаются входные данные в сформированном элементе, устанавливается в форме определенной комбинации высокого и низкого логических уровней сигнала на входах используемого библиотечного примитива DSP48E, которые соответствуют шестнадцати младшим значащим разрядам шины A. Для изменения этого значения без повторного запуска «мастера»

Architecture Wizard достаточно открыть описание элемента *macc\_12\_16\_const* в интегрированном HDL-редакторе САПР серии Xilinx ISE и изменить подключение указанных линий входной шины A компонента DSP48E\_INST. Значение логической единицы в каком-либо разряде входного порта A этого компонента устанавливается при подключении соответствующего проводника этой шины к цепи сигнала VCC\_A, а логического нуля — к цепи сигнала GND\_A.

В представленном VHDL-описании элемента *macc\_12\_16\_const* используется следующая система условных обозначений входных и выходных портов:

- B\_IN[11:0] — 12-разрядная входная шина, совокупность сигналов которой определяет значения отсчетов входных данных;
- CARRYIN\_IN — вход внешнего сигнала переноса;
- CE1\_IN, CE2\_IN, CE1\_IN и CE2\_IN — входы сигналов разрешения синхронизации для буферных регистров, задействованных на входных информационных шинах данных;
- CECARRYIN\_IN — вход сигнала разрешения синхронизации для буферного регистра, установленного в цепи внешнего сигнала входного переноса;
- CECTRL\_IN — вход сигнала разрешения синхронизации буферного регистра, установленного на шине управления путем выбора источников входных данных для арифметическо-логического блока аппаратной секции DSP48E, используемой для реализации сформированного элемента;
- CEMULTCARRYIN\_IN — вход сигнала разрешения синхронизации для буферного регистра, установленного в цепи внутреннего сигнала входного переноса;
- CEM\_IN — вход сигнала разрешения синхронизации для буферного регистра, установленного на выходе умножителя (на входе арифметическо-логического блока);
- CEP\_IN — вход сигнала разрешения синхронизации для выходного регистра сгенерированного элемента;
- CLK\_IN — вход тактового сигнала;
- LOAD\_IN — вход сигнала управления загрузкой значений входных данных в аккумулятор сформированного элемента;
- RSTALLCARRYIN\_IN — вход сигнала сброса буферных регистров, задействованных в цепях входного переноса;
- RSTA\_IN и RSTB\_IN — входы сигналов сброса буферных регистров, установленных на входных информационных шинах данных;
- RSTCTRL\_IN — вход сигнала сброса буферного регистра, установленного на шине управления путем выбора источников входных данных для арифметическо-логического блока используемой аппаратной секции DSP48E;
- RSTM\_IN — вход сигнала сброса буферного регистра, задействованного на выходе

умножителя (на входе арифметическо-логического блока);

- RSTP\_IN — вход сигнала сброса для выходного регистра сгенерированного элемента;
- ACOUT\_OUT[29:0] — 30-разрядная выходная шина данных, используемая для подключения к входной шине ACIN\_IN следующего модуля ЦОС при каскадном соединении;
- VCOOUT\_OUT[17:0] — 18-разрядная выходная шина данных, предназначенная для подключения к входной шине данных B\_IN следующего модуля ЦОС при каскадном соединении;
- PCOUT\_OUT[47:0] — 48-разрядная выходная шина данных, применяемая при каскадном соединении модулей ЦОС;
- P\_OUT[29:0] — 30-разрядная выходная шина данных, на которой отображается значение результата выполняемых операций умножения с накоплением.

Определение требуемых значений всех необходимых атрибутов библиотечного примитива DSP48E, образующего основу сформированного VHDL-кода описания элемента *macc\_12\_16\_const*, осуществляется с помощью следующей совокупности выражений, которые содержатся в дополнительном файле с расширением *ucf*, сгенерированном «мастером» Architecture Wizard:

```
# Generated by Xilinx Architecture Wizard
# --- UCF Template Only ---
# Cut and paste these attributes into the project's UCF file, if desired
INST DSP48E_INST ACASCREG = 1;
INST DSP48E_INST ALUMODEREG = 0;
INST DSP48E_INST AREG = 2;
INST DSP48E_INST AUTORESET_PATTERN_DETECT = FALSE;
INST DSP48E_INST AUTORESET_PATTERN_DETECT_OPTINV = MATCH;
INST DSP48E_INST A_INPUT = DIRECT;
INST DSP48E_INST BCASCREG = 1;
INST DSP48E_INST BREG = 2;
INST DSP48E_INST B_INPUT = DIRECT;
INST DSP48E_INST CARRYINREG = 1;
INST DSP48E_INST CARRYINSELREG = 0;
INST DSP48E_INST CREG = 0;
INST DSP48E_INST MASK = 3FFFFFFF;
INST DSP48E_INST MREG = 1;
INST DSP48E_INST MULTCARRYINREG = 1;
INST DSP48E_INST OPMODEREG = 1;
INST DSP48E_INST PATTERN = 000000000000;
INST DSP48E_INST PREG = 1;
INST DSP48E_INST SEL_MASK = MASK;
INST DSP48E_INST SEL_PATTERN = PATTERN;
INST DSP48E_INST SEL_ROUNDING_MASK = SEL_MASK;
INST DSP48E_INST USE_MULT = MULT_S;
INST DSP48E_INST USE_PATTERN_DETECT = NO_PATDET;
INST DSP48E_INST USE_SIMD = ONE48;
```

### Пример описания элемента, осуществляющего операции умножения с вычитанием, сформированного с помощью «мастера» Architecture Wizard для реализации на базе аппаратных модулей DSP48E в ПЛИС FPGA серии Virtex-5

В качестве примера описания компонента, выполняющего операции умножения с вычитанием, предназначенного для реализации на основе аппаратных секций ЦОС в кристаллах серии Virtex-5, приводится VHDL-описание

элемента *macc\_sub\_15\_18\_34*, сформированное с помощью «мастера» Architecture Wizard. Этот элемент осуществляет перемножение значений, поступающих в виде 15- и 18-разрядного двоичного кода на соответствующие информационные входные шины данных, с последующим вычислением разности произведения и содержимого аккумулятора с учетом значения внешнего сигнала входного переноса (заема). Итоговый результат выполнения указанных операций представляется на выходной шине сгенерированного элемента в форме 34-разрядного двоичного числа. Функцию входных информационных портов в элементе *macc\_sub\_15\_18\_34* выполняют шины данных A и BCIN аппаратного модуля DSP48E:

```
--Command: xaw2vhdl-st D:\PR\macc_sub_15_18_34.xaw D:\PR\
macc_sub_15_18_34
--Design Name: macc_sub_15_18_34
--Device: xc5vxx50t-ff1136-3
--
-- Module macc_sub_15_18_34
-- Generated by Xilinx Architecture Wizard
-- Written for synthesis tool: XST
--
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
library UNISIM;
use UNISIM.Vcomponents.ALL;
--
entity macc_sub_15_18_34 is
port (
    A_IN : in std_logic_vector (14 downto 0);
    BCIN_IN : in std_logic_vector (17 downto 0);
    CARRYIN_IN : in std_logic;
    CE_A1_IN : in std_logic;
    CE_B1_IN : in std_logic;
    CE_CARRYIN_IN : in std_logic;
    CE_CTRL_IN : in std_logic;
    CEMULTCARRYIN_IN : in std_logic;
    CEM_IN : in std_logic;
    CEP_IN : in std_logic;
    CLK_IN : in std_logic;
    LOAD_IN : in std_logic;
    RSTALLCARRYIN_IN : in std_logic;
    RSTA_IN : in std_logic;
    RSTB_IN : in std_logic;
    RSTCTRL_IN : in std_logic;
    RSTM_IN : in std_logic;
    RSTP_IN : in std_logic;
    ACOUT_OUT : out std_logic_vector (29 downto 0);
    BCOUT_OUT : out std_logic_vector (17 downto 0);
    PCOUT_OUT : out std_logic_vector (47 downto 0);
    P_OUT : out std_logic_vector (33 downto 0)
);
end macc_sub_15_18_34;
--
architecture BEHAVIORAL of macc_sub_15_18_34 is
    signal GND_ALUMODE : std_logic;
    signal GND_BUS_3 : std_logic_vector (2 downto 0);
    signal GND_BUS_18 : std_logic_vector (17 downto 0);
    signal GND_BUS_30 : std_logic_vector (29 downto 0);
    signal GND_BUS_48 : std_logic_vector (47 downto 0);
    signal GND_OPMODE : std_logic;
    signal P_float : std_logic_vector (13 downto 0);
    signal VCC_ALUMODE : std_logic;
    signal VCC_OPMODE : std_logic;
begin
    GND_ALUMODE <= '0';
    GND_BUS_3(2 downto 0) <= «000»;
    GND_BUS_18(17 downto 0) <= «00000000000000000000»;
    GND_BUS_30(29 downto 0) <= «00000000000000000000000000000000»;
    GND_BUS_48(47 downto 0) <=
        «0000000000000000000000000000000000000000000000000000000000000000»;
    GND_OPMODE <= '0';
    VCC_ALUMODE <= '1';
    VCC_OPMODE <= '1';
    DSP48E_INST : DSP48E
        generic map(
            ACASCREG => 1,
            ALUMODEREG => 0,
            AREG => 1,
            AUTORESET_PATTERN_DETECT => FALSE,
            AUTORESET_PATTERN_DETECT_OPTINV => «MATCH»,
            A_INPUT => «DIRECT»,
            BCASCREG => 1,
            BREG => 1,
            B_INPUT => «CASCADE»,
            CARRYINREG => 1,
```

```
CARRYINSELREG => 0,
CREG => 0,
MASK => x«3FFFFFFF»,
MREG => 1,
MULTCARRYINREG => 1,
OPMODEREG => 1,
PATTERN => x«00000000000000»,
PREG => 1,
SEL_MASK => «MASK»,
SEL_PATTERN => «PATTERN»,
SEL_ROUNDING_MASK => «SEL_MASK»,
USE_MULT => «MULT_S»,
USE_PATTERN_DETECT => «NO_PATDET»,
USE_SIMD => «ONE48»
)
port map (
    A(29)=>A_IN(14),
    A(28)=>A_IN(14),
    A(27)=>A_IN(14),
    A(26)=>A_IN(14),
    A(25)=>A_IN(14),
    A(24)=>A_IN(14),
    A(23)=>A_IN(14),
    A(22)=>A_IN(14),
    A(21)=>A_IN(14),
    A(20)=>A_IN(14),
    A(19)=>A_IN(14),
    A(18)=>A_IN(14),
    A(17)=>A_IN(14),
    A(16)=>A_IN(14),
    A(15)=>A_IN(14),
    A(14 downto 0)=>A_IN(14 downto 0),
    ACIN(29 downto 0)=>GND_BUS_30(29 downto 0),
    ALUMODE(3)=>GND_ALUMODE,
    ALUMODE(2)=>GND_ALUMODE,
    ALUMODE(1)=>VCC_ALUMODE,
    ALUMODE(0)=>VCC_ALUMODE,
    B(17 downto 0)=>GND_BUS_18(17 downto 0),
    BCIN(17 downto 0)=>BCIN_IN(17 downto 0),
    C(47 downto 0)=>GND_BUS_48(47 downto 0),
    CARRYCASCIN=>GND_ALUMODE,
    CARRYIN=>CARRYIN_IN,
    CARRYINSEL(2 downto 0)=>GND_BUS_3(2 downto 0),
    CEALUMODE=>VCC_OPMODE,
    CE_A1=>CE_A1_IN,
    CE_A2=>VCC_OPMODE,
    CE_B1=>CE_B1_IN,
    CE_B2=>VCC_OPMODE,
    CE_C=>VCC_OPMODE,
    CE_CARRYIN=>CE_CARRYIN_IN,
    CE_CTRL=>CE_CTRL_IN,
    CEM=>CEM_IN,
    CEMULTCARRYIN=>CEMULTCARRYIN_IN,
    CEP=>CEP_IN,
    CLK=>CLK_IN,
    MULTSIGNIN=>GND_ALUMODE,
    OPMODE(6)=>GND_OPMODE,
    OPMODE(5)=>LOAD_IN,
    OPMODE(4)=>GND_OPMODE,
    OPMODE(3)=>GND_OPMODE,
    OPMODE(2)=>VCC_OPMODE,
    OPMODE(1)=>GND_OPMODE,
    OPMODE(0)=>VCC_OPMODE,
    PCIN(47 downto 0)=>GND_BUS_48(47 downto 0),
    RSTA=>RSTA_IN,
    RSTALLCARRYIN=>RSTALLCARRYIN_IN,
    RSTALUMODE=>GND_ALUMODE,
    RSTB=>RSTB_IN,
    RSTC=>GND_ALUMODE,
    RSTCTRL=>RSTCTRL_IN,
    RSTM=>RSTM_IN,
    RSTP=>RSTP_IN,
    ACOUT(29 downto 0)=>ACOUT_OUT(29 downto 0),
    BCOUT(17 downto 0)=>BCOUT_OUT(17 downto 0),
    CARRYCASCOUT=>open,
    CARRYOUT=>open,
    MULTSIGNOUT=>open,
    OVERFLOW=>open,
    P(47 downto 34)=>P_float(13 downto 0),
    P(33 downto 0)=>P_OUT(33 downto 0),
    PATTERNBDetect=>open,
    PATTERNDETECT=>open,
    PCOUT(47 downto 0)=>PCOUT_OUT(47 downto 0),
    UNDERFLOW=>open
);
end BEHAVIORAL;
```

В структуру сгенерированного элемента *macc\_sub\_15\_18\_34* включены входные и выходные регистры. Каждый из этих регистров имеет индивидуальные входы сигналов сброса и разрешения синхронизации, которые представлены в составе интерфейса этого элемента. Кроме того, в интерфейсе присутствуют дополнительные выходные шины ACOUT,

BCOUT и PCOUT, предоставляющие возможность использования сформированного элемента в составе каскадного соединения аппаратных модулей ЦОС.

В систему условных обозначений входных и выходных портов элемента *macc\_sub\_15\_18\_34* входит большинство идентификаторов, представленных в описании интерфейса умножителя-накопителя *macc\_12\_16\_const*, который был рассмотрен в предыдущем разделе. Кроме этих обозначений, в описании интерфейса элемента *macc\_sub\_15\_18\_34* дополнительно применяются следующие идентификаторы:

- *A\_IN[14:0]* — 15-разрядная входная шина данных, совокупность сигналов которой определяет значение первого сомножителя;
  - *BCIN\_IN[17:0]* — 18-разрядная входная шина данных, совокупность сигналов которой определяет значение второго сомножителя.
- В случае использования представленного VHDL-кода элемента *macc\_sub\_15\_18\_34* в составе описания разрабатываемого устройства ЦОС следует включить в файл временных и топологических ограничений проекта САПР серии Xilinx ISE следующую совокупность выражений, предназначенных для установки требуемых значений атрибутов библиотечного примитива DSP48E.

```
# Generated by Xilinx Architecture Wizard
# --- UCF Template Only ---
# Cut and paste these attributes into the project's UCF file, if desired
INST DSP48E_INST ACASCREG = 1;
INST DSP48E_INST ALUMODEREG = 0;
INST DSP48E_INST AREG = 1;
INST DSP48E_INST AUTORESET_PATTERN_DETECT = FALSE;
INST DSP48E_INST AUTORESET_PATTERN_DETECT_OPTINV = MATCH;
INST DSP48E_INST A_INPUT = DIRECT;
INST DSP48E_INST BCASCREG = 1;
INST DSP48E_INST BREG = 1;
INST DSP48E_INST B_INPUT = CASCADE;
INST DSP48E_INST CARRYINREG = 1;
INST DSP48E_INST CARRYINSELREG = 0;
INST DSP48E_INST CREG = 0;
INST DSP48E_INST MASK = 3FFFFFFF;
INST DSP48E_INST MREG = 1;
INST DSP48E_INST MULTCARRYINREG = 1;
INST DSP48E_INST OPMODEREG = 1;
INST DSP48E_INST PATTERN = 000000000000;
INST DSP48E_INST PREG = 1;
INST DSP48E_INST SEL_MASK = MASK;
INST DSP48E_INST SEL_PATTERN = PATTERN;
INST DSP48E_INST SEL_ROUNDING_MASK = SEL_MASK;
INST DSP48E_INST USE_MULT = MULT_S;
INST DSP48E_INST USE_PATTERN_DETECT = NO_PATDET;
INST DSP48E_INST USE_SIMD = ONE48;
```

### Пример описания элемента, осуществляющего операции умножения с накоплением и вычитанием, подготовленного с помощью «мастера» Architecture Wizard для реализации на базе аппаратных модулей DSP48E в ПЛИС FPGA серии Virtex-5

Примером компонента, включающего в себя аппаратный умножитель и аккумулятор с возможностью динамического выбора типа выполняемой операции, реализуемого на базе аппаратных модулей DSP48E кристаллов FPGA семейств Virtex-5 LX, Virtex-5 LXT, Virtex-5 SXT, Virtex-5 FXT и Virtex-5 TXT, яв-

ляется описание элемента *macc\_add\_sub\_18\_18\_38*. Этот элемент осуществляет операции перемножения двух 18-разрядных значений входных данных с последующим вычислением суммы или разности с содержимым аккумулятора. Операции суммирования и вычитания в рассматриваемом элементе выполняются с учетом значения внешнего сигнала входного переноса. Значение результата вычислений отображается на выходной шине элемента *macc\_add\_sub\_18\_18\_38* в виде 38-разрядного двоичного кода. Входными портами этого элемента являются шины данных ACIN и BCIN, используемые, чаще всего, для подключения соответствующих выходных шин ACOUT и BCOUT предшествующей секции DSP48E при каскадном соединении аппаратных модулей ЦОС. Выбор типа операции, выполняемой аккумулятором данного элемента, производится подачей соответствующего двоичного кода в 4-разрядный входной порт управления, который подключен к шине выбора функции арифметическо-логического блока ALUMODE аппаратного модуля ЦОС.

Текст VHDL-описания элемента *macc\_add\_sub\_18\_18\_38*, сгенерированный с помощью «мастера» Architecture Wizard, имеет следующий вид:

```
--Command: xaw2vhdl-st D:\PRJ\macc_add_sub_18_18_38.xaw
D:\PRJ\macc_add_sub_18_18_38
--Design Name: macc_add_sub_18_18_38
--Device: xc5vfx30t-ff665-3
--
-- Module macc_add_sub_18_18_38
-- Generated by Xilinx Architecture Wizard
-- Written for synthesis tool: XST
--
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
library UNISIM;
use UNISIM.vcomponents.ALL;
--
entity macc_add_sub_18_18_38 is
  port (
    ACIN_IN : in std_logic_vector (17 downto 0);
    ALUMODE_IN : in std_logic_vector (3 downto 0);
    BCIN_IN : in std_logic_vector (17 downto 0);
    CARRYIN_IN : in std_logic;
    CEALUMODE_IN : in std_logic;
    CEA1_IN : in std_logic;
    CEA2_IN : in std_logic;
    CEB1_IN : in std_logic;
    CEB2_IN : in std_logic;
    CECARRYIN_IN : in std_logic;
    CECTRL_IN : in std_logic;
    CEMULTCARRYIN_IN : in std_logic;
    CEM_IN : in std_logic;
    CEP_IN : in std_logic;
    CLK_IN : in std_logic;
    LOAD_IN : in std_logic;
    RSTALLCARRYIN_IN : in std_logic;
    RSTALUMODE_IN : in std_logic;
    RSTA_IN : in std_logic;
    RSTB_IN : in std_logic;
    RSTCTRL_IN : in std_logic;
    RSTM_IN : in std_logic;
    RSTP_IN : in std_logic;
    ACOUT_OUT : out std_logic_vector (29 downto 0);
    BCOUT_OUT : out std_logic_vector (17 downto 0);
    PCOUT_OUT : out std_logic_vector (47 downto 0);
    P_OUT : out std_logic_vector (37 downto 0)
  );
end macc_add_sub_18_18_38;
--
architecture BEHAVIORAL of macc_add_sub_18_18_38 is
  signal GND_BUS_3 : std_logic_vector (2 downto 0);
  signal GND_BUS_18 : std_logic_vector (17 downto 0);
  signal GND_BUS_30 : std_logic_vector (29 downto 0);
  signal GND_BUS_48 : std_logic_vector (47 downto 0);
  signal GND_OPMODE : std_logic;
  signal P_float : std_logic_vector (9 downto 0);
  signal VCC_OPMODE : std_logic;
begin
  GND_BUS_3(2 downto 0) <= «000»;
  GND_BUS_18(17 downto 0) <= «00000000000000000000»;
  GND_BUS_30(29 downto 0) <= «00000000000000000000000000000000»;
  GND_BUS_48(47 downto 0) <=
    «0000000000000000000000000000000000000000000000000000000000000000»;
  GND_OPMODE <= '0';
  VCC_OPMODE <= '1';
  DSP48E_INST : DSP48E
  generic map(
    ACASCREG => 1,
    ALUMODEREG => 1,
    AREG => 2,
    AUTORESET_PATTERN_DETECT => FALSE,
    AUTORESET_PATTERN_DETECT_OPTINV => «MATCH»,
    A_INPUT => «CASCADE»,
    BCASCREG => 1,
    BREG => 2,
    B_INPUT => «CASCADE»,
    CARRYINREG => 1,
    CARRYINSELREG => 0,
    CREG => 0,
    MASK => x«3FFFFFFF»,
    MREG => 1,
    MULTCARRYINREG => 1,
    OPMODEREG => 1,
    PATTERN => x«000000000000»,
    PREG => 1,
    SEL_MASK => «MASK»,
    SEL_PATTERN => «PATTERN»,
    SEL_ROUNDING_MASK => «SEL_MASK»,
    USE_MULT => «MULT_S»,
    USE_PATTERN_DETECT => «NO_PATDET»,
    USE_SIMD => «ONE48»
  )
  port map (
    A(29 downto 0) => GND_BUS_30(29 downto 0),
    ACIN(29) => ACIN_IN(17),
    ACIN(28) => ACIN_IN(17),
    ACIN(27) => ACIN_IN(17),
    ACIN(26) => ACIN_IN(17),
    ACIN(25) => ACIN_IN(17),
    ACIN(24) => ACIN_IN(17),
    ACIN(23) => ACIN_IN(17),
    ACIN(22) => ACIN_IN(17),
    ACIN(21) => ACIN_IN(17),
    ACIN(20) => ACIN_IN(17),
    ACIN(19) => ACIN_IN(17),
    ACIN(18) => ACIN_IN(17),
    ACIN(17 downto 0) => ACIN_IN(17 downto 0),
    ALUMODE(3 downto 0) => ALUMODE_IN(3 downto 0),
    B(17 downto 0) => GND_BUS_18(17 downto 0),
    BCIN(17 downto 0) => BCIN_IN(17 downto 0),
    C(47 downto 0) => GND_BUS_48(47 downto 0),
    CARRYASCIN => GND_OPMODE,
    CARRYIN => CARRYIN_IN,
    CARRYINSEL(2 downto 0) => GND_BUS_3(2 downto 0),
    CEALUMODE => CEALUMODE_IN,
    CEA1 => CEA1_IN,
    CEA2 => CEA2_IN,
    CEB1 => CEB1_IN,
    CEB2 => CEB2_IN,
    CEC => VCC_OPMODE,
    CECARRYIN => CECARRYIN_IN,
    CECTRL => CECTRL_IN,
    CEM => CEM_IN,
    CEMULTCARRYIN => CEMULTCARRYIN_IN,
    CEP => CEP_IN,
    CLK => CLK_IN,
    MULTSIGNIN => GND_OPMODE,
    OPMODE(6) => GND_OPMODE,
    OPMODE(5) => LOAD_IN,
    OPMODE(4) => GND_OPMODE,
    OPMODE(3) => GND_OPMODE,
    OPMODE(2) => VCC_OPMODE,
    OPMODE(1) => GND_OPMODE,
    OPMODE(0) => VCC_OPMODE,
    PCIN(47 downto 0) => GND_BUS_48(47 downto 0),
    RSTA => RSTA_IN,
    RSTALLCARRYIN => RSTALLCARRYIN_IN,
    RSTALUMODE => RSTALUMODE_IN,
    RSTB => RSTB_IN,
    RSTC => GND_OPMODE,
    RSTCTRL => RSTCTRL_IN,
    RSTM => RSTM_IN,
    RSTP => RSTP_IN,
    ACOUT(29 downto 0) => ACOUT_OUT(29 downto 0),
    BCOUT(17 downto 0) => BCOUT_OUT(17 downto 0),
    CARRYASCOUT => open,
    CARRYOUT => open,
    MULTSIGNOUT => open,
    OVERFLOW => open,
    P(47 downto 38) => P_float(9 downto 0),
    P(37 downto 0) => P_OUT(37 downto 0),
    PATTERNBDETECT => open,
    PATTERNDETECT => open,
    PCOUT(47 downto 0) => PCOUT_OUT(47 downto 0),
    UNDERFLOW => open
  );
end BEHAVIORAL;
```

В структуре представленного элемента задействованы все входные, выходные и буферные регистры, предусмотренные архитектурой аппаратной секции DSP48E кристаллов серии Virtex-5. Входы сброса и разрешения синхронизации для каждого из этих регистров включены в виде соответствующих портов в состав интерфейса элемента `mac_add_sub_18_18_38`.

В описании интерфейса сформированного элемента кроме условных обозначений входных и выходных портов, представленных в предыдущих разделах, дополнительно используются следующие идентификаторы:

- `ACIN_IN[17:0]` — 18-разрядная входная шина данных, совокупность сигналов которой определяет значение первого операнда;
- `ALUMODE_IN[3:0]` — 4-разрядная входная шина управления выбором типа операции, выполняемой аккумулятором;
- `CEALUMODE_IN` — вход сигнала разрешения синхронизации для буферного регистра, установленного на шине управления `ALUMODE_IN`;
- `RSTALUMODE_IN` — вход сигнала сброса для буферного регистра, установленного на шине управления `ALUMODE_IN`.

Значения всех необходимых атрибутов библиотечного примитива DSP48E, на базе

которого сгенерировано VHDL-описание элемента `mac_add_sub_18_18_38`, задаются с помощью следующей совокупности выражений, которые нужно добавить в состав файла временных и топологических ограничений проекта разрабатываемого устройства ЦОС в САПР серии Xilinx ISE:

```
# Generated by Xilinx Architecture Wizard
# --- UCF Template Only ---
# Cut and paste these attributes into the project's UCF file, if desired
INST DSP48E_INST ACASCREG = 1;
INST DSP48E_INST ALUMODEREG = 1;
INST DSP48E_INST AREG = 2;
INST DSP48E_INST AUTORESET_PATTERN_DETECT = FALSE;
INST DSP48E_INST AUTORESET_PATTERN_DETECT_OPTINV = MATCH;
INST DSP48E_INST A_INPUT = CASCADE;
INST DSP48E_INST BCASCREG = 1;
INST DSP48E_INST BREG = 2;
INST DSP48E_INST B_INPUT = CASCADE;
INST DSP48E_INST CARRYINREG = 1;
INST DSP48E_INST CARRYINSELREG = 0;
INST DSP48E_INST CREG = 0;
INST DSP48E_INST MASK = 3FFFFFFF;
INST DSP48E_INST MREG = 1;
INST DSP48E_INST MULTCARRYINREG = 1;
INST DSP48E_INST OPMODEREG = 1;
INST DSP48E_INST PATTERN = 000000000000;
INST DSP48E_INST PREG = 1;
INST DSP48E_INST SEL_MASK = MASK;
INST DSP48E_INST SEL_PATTERN = PATTERN;
INST DSP48E_INST SEL_ROUNDING_MASK = SEL_MASK;
INST DSP48E_INST USE_MULT = MULT_S;
INST DSP48E_INST USE_PATTERN_DETECT = NO_PATDET;
INST DSP48E_INST USE_SIMD = ONE48;
```

*Окончание 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. 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.
3. Зотов В. Ю. Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPack ISE. М.: Горячая линия – Телеком, 2003.
4. Зотов В. Ю. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx. М.: Горячая линия – Телеком, 2006.
5. Virtex-5 FPGA XtremeDSP Design Considerations. User Guide. Xilinx, 2009.
6. Virtex-5 Family Overview. Xilinx, 2009.
7. Virtex -5 FPGA User Guide. Xilinx, 2009.
8. Зотов В. Проектирование цифровых устройств, реализуемых на базе ПЛИС FPGA фирмы Xilinx, с использованием средств CORE Generator // Компоненты и технологии. 2006. № 12. 2007. № 1.
9. Зотов В. Разработка базовых компонентов цифровых устройств, реализуемых на базе ПЛИС FPGA фирмы Xilinx, с помощью генератора параметризованных модулей CORE Generator // Компоненты и технологии. 2007. № 2–12. 2008. № 1–8.