

УДК 004.315.5

Опанасенко В. Н., канд. техн. наук,  
Лисовый А. Н.

## ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ ДЛЯ РЕАЛИЗАЦИИ ФУНКЦИИ МАС НА БАЗЕ ПЛИС

Институт кибернетики им. В.М. Глушкова НАН Украины

*Предложена структурная реализация функции умножения с накоплением в элементном базисе ПЛИС типа FPGA, выполненная путем поведенческого описания алгоритма на языке VHDL. Реализована проверка функционирования модуля методом моделирования в системе ModelSim Xilinx Edition - MXE II с помощью проверочного стенда.*

### Введение

Внедрение программируемых логических интегральных схем (ПЛИС) и HDL-технологии для выполнения проектов в этом элементном базисе при проектировании цифровых устройств интенсифицировало разработку цифровых модулей, представляющих собой готовые технические решения. Эти решения, называемые IP (Intellectual Property) – Core блоками, могут быть настроены под конкретные требования нового проекта. Они относятся к *soft cores* или «мягким» заготовкам, описанным с помощью HDL-языка [1]. Такие заготовки можно легко настраивать в соответствии с требованиями нового проекта и они, как правило, независимы от технологии изготовления ПЛИС.

Архитектура современных кристаллов FPGA типа Virtex и Spartan фирмы Xilinx [2] оптимизирована для использования как *hard*, так и *soft cores*, и позволяет легко интегрировать их в проекты. Данные кристаллы имеют встроенные умножители в виде *hard cores* – специализированных областей кристалла FPGA, выделенных для этих функций. В этих областях создаются блоки фиксированной структуры, оптимизированные для реализации заданной функции.

Встроенные умножители в кристаллах FPGA позволяют создавать быстродействующие арифметические устройства. Использование семейства кристаллов FPGA, развитого инструментального программного обеспечения и широкого набора IP-core позволяет реализовать сложные проекты на одном кристалле. Блоки IP-

*core* для реализации функции умножения с накоплением находят широкое применение при построении математических сопроцессоров, DSP-процессоров, встроенных арифметических сопроцессоров [3]. В связи с этим рядом фирм разрабатываются собственные *soft cores* для реализации таких арифметических операций. В данной статье предлагается подход к разработке устройств на основе современных кристаллов ПЛИС, осуществляющих выполнение операции умножения с накоплением, с использованием современных инструментальных средств для проектирования, моделирования и верификации.

Работа выполнена в рамках ДНТІ «Нові вітчизняні інтелектуальні комп'ютерні засоби» Министерства образования и науки Украины.

### Постановка задачи

При проектировании цифровых устройств на базе кристаллов ПЛИС в настоящее время широко используются язык VHDL [4]. Данный язык предоставляет пользователю возможность структурного либо поведенческого описания проекта. Одним из основных способов повышения быстродействия вычислительных устройств является организация конвейерной обработки [5]. Конвейерное устройство характеризуется определенным количеством ступеней конвейера (на каждой из которых выполняется заданное преобразование с помощью функционального генератора) и временем синхронизации этих ступеней. При правильной организации конвейера увеличение сту-

пней приводит к уменьшению времени синхронизации, но при этом увеличиваются аппаратные затраты и усложняется организация схемы управления.

Базовый фрагмент схемы без конвейеризации и с конвейеризацией представлены соответственно на рис. 1 (а, б).

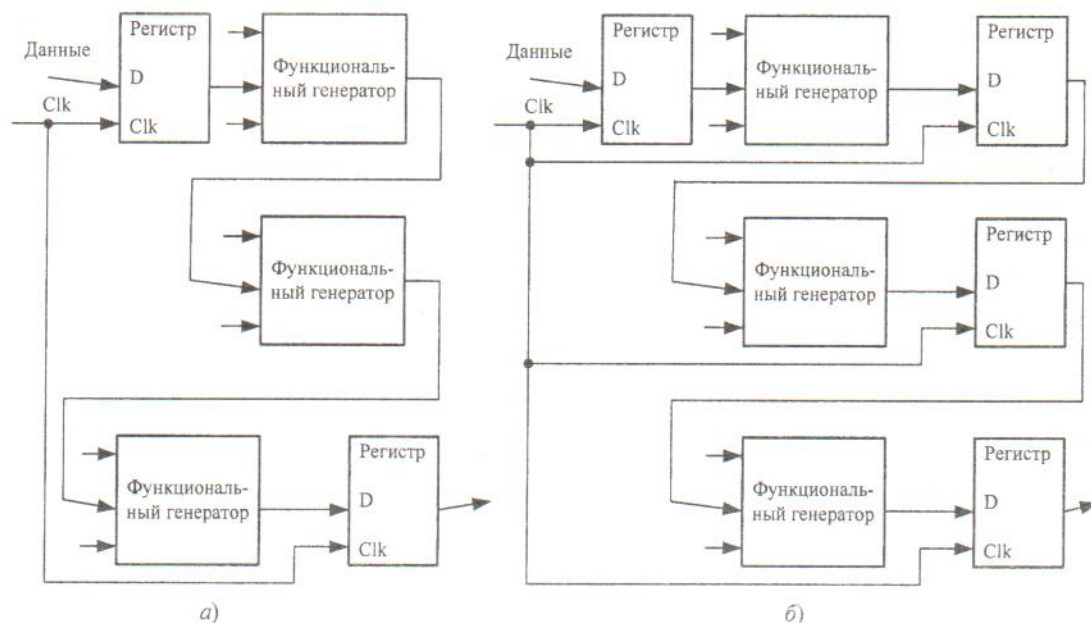


Рис. 1. Базовый фрагмент схемы: а) без конвейеризации; б) с конвейеризацией

### Реализация MAC функции

Рассмотрим схему (MAC1) умножения двух 16-разрядных чисел с накоплением 32-разрядной суммы без конвейеризации. Данная схема представлена в виде поведенческого описания на языке VHDL на рис. 2.

Хорошей альтернативой схемы MAC1 может быть схема MAC2, которая реализована в виде конвейера.

Различие структур без конвейеризации и с конвейеризацией заключается в разделении комбинационной логики последней на две части путем введения двух регистров: регистра произведения (входной сигнал  $p1$ , выходной –  $p2$ ) и регистра слагаемого (входной сигнал  $c1$ , выходной –  $c2$ ). Регистр слагаемого вводится для синхронизации подготовки слагаемых путем задержки одного из них на один такт.

Отличие схемы с конвейеризацией состоит в наличии фиксаторов в виде регистров (синхронных запоминающих элементов) для обеспечения синхронизации функциональных генераторов.

Задание регистров производится объявлением сигналов с присвоением им соответствующих значений при наличии положительного фронта тактирующего сигнала  $clk$ .

### Реализация MAC функции с конвейерным умножителем

В структуру ПЛИС последних поколений введены встроенные (*hard cores*) умножители  $MULT18X18$ . На рис. 3 приведено описание схемы MAC3, в котором конвейерный умножитель декларирован в качестве компонента *mult\_pipe*. Идея использования компонентов заключается в том, что реализация функции разбита на части (компоненты), которые взаимодействуют между собой с помощью сигналов. Декларация компонента осуществляется при помощи зарезервированного слова *component*.

```
entity mult_add is
Port ( a : in std_logic_vector(15 downto 0);
      b : in std_logic_vector(15 downto 0);
      c : in std_logic_vector(31 downto 0);
      clk : in std_logic;
      rst : in std_logic;
      ce : in std_logic;
      s : out std_logic_vector(31 downto 0));
end entity;
architecture mult_add of mult_add is
signal s1 : std_logic_vector (31 downto 0);
signal a1 : std_logic_vector (15 downto 0);
signal b1 : std_logic_vector (15 downto 0);
signal c1 : std_logic_vector (31 downto 0);
begin
s1 <= a1*b1 + c1;
process (clk)
begin
if clk'event and clk = '1' then
a1<=a; b1<=b; c1<=c;
if rst = '1' then
s <= (others => '0');
elsif ce = '1' then
s <=s1;
end if;
end if;
end process;
end architecture;
```

Рис. 2. Описание MAC без конвейеризации

Компенсация задержек, вносимых ступенями конвейера умножителя для сигнала *s*, поступающего в качестве слагаемого на сумматор, осуществляется регистрами, представленными в описании сигналами *c1*, *c2*, *c3*, *c4* и *c5*.

```
entity mult_add_struc is
Port ( a : in std_logic_vector(15 downto 0);
      b : in std_logic_vector(15 downto 0);
      c : in std_logic_vector(31 downto 0);
      clk : in std_logic;
      s : out std_logic_vector(31 downto 0));
end mult_add_struc;
architecture Struct of mult_add_struc is
component mult_pipe
Port ( clk : in std_logic;
      ain : in std_logic_vector (15 downto 0);
      bin : in std_logic_vector (15 downto 0);
      q : out std_logic_vector (31 downto 0));
end component;
signal s1,p : std_logic_vector (31 downto 0);
signal c1,c2,c3,c4,c5 : std_logic_vector (31 downto 0);
begin
s1 <= p + c5;
process (clk)
begin
if clk'event and clk = '1' then
c1<=c; c2<=c1; c3<=c2; c4<=c3; c5<=c4; s <= s1;
end if;
end process;
u0: mult_pipe port map (clk => clk,
ain => a,
bin => b,
q => p);
end Struct;
```

Рис. 3. Описание MAC с конвейерным умножителем

В табл. 1 приведены оценки аппаратурных (в количестве слайсов) и временных затрат (максимальное быстродействие) разработанных модулей при их реализации в кристаллах ПЛИС. Схема *MAC3* реализована в двух типах кристаллов (*XC2S50-5PQ208* серии *Spartan 2* и *XC2V80-5FG256* серии *Virtex-II*).

Таблица 1.

Оценочные характеристики разработанных модулей

Тип кристалла ПЛИС	Тип модуля	Аппаратные ресурсы (кол-во слайсов)	Параметры быстродействия (нс) / Тактовая частота (МГц)	Наличие конвейера
<i>XC2S30-5PQ208</i> ( <i>Spartan 2</i> )	<i>MAC1</i>	157 (36% ресурсов кристалла)	28,6 нс / 35 МГц	-
<i>XC2S30-5PQ208</i> ( <i>Spartan 2</i> )	<i>MAC2</i>	173 (40% ресурсов кристалла)	17,5 нс / 57 МГц	+
<i>XC2S50-5PQ208</i> ( <i>Spartan 2</i> )	<i>MAC3</i>	316 (41% ресурсов кристалла)	10,0 нс / 100 МГц	+
<i>XC2V80-5FG256</i> ( <i>Virtex-II</i> )	<i>MAC3</i>	316 (61% ресурсов кристалла)	8,6 нс / 116 МГц	+

### Верификация модулей

Верификация полученных результатов производилась моделированием с использованием системы *ModelSim Xilinx Edition – MXE II* [11]. На рис. 4 приведена временная диаграмма работы конвейерного умножителя. Для наглядности множи-

тели и произведение представлены в аналоговой форме. Изменение множителей во времени осуществляется по линейному закону, при этом один из них изменяется с положительным приращением, другой – с отрицательным.

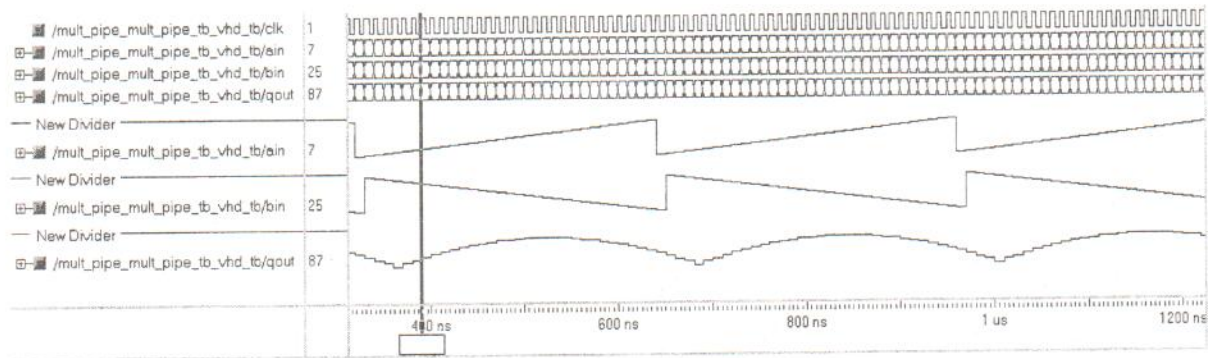


Рис. 4. Временная диаграмма работы конвейерного умножителя.

### Выводы

Структурная реализация *MAC3* на базе кристалла *XC2V80-5FG256* обладает максимальным быстродействием среди рассмотренных, но требует максимум аппаратных ресурсов. Таких же аппаратных ресурсов требует реализация *MAC3* на базе кристалла *XC2S50-5PQ208*, но имеет меньшее быстродействие, что связано, в первую очередь с технологическими особенностями кристаллов серии *Spartan 2*. Реализация *MAC1* за счет отсутствия аппаратных ступеней конвейера имеет минимальные аппаратные затраты среди рассмотренных структур, но обладает и минимальным быстродействием. Реализация *MAC2* занимает промежуточное место между реализациями *MAC1* и *MAC3* по временным и аппаратным затратам.

Синтезированные структуры верифицированы методом моделирования в системе *ModelSim Xilinx Edition – MXE II* с помощью стенда для проверки, что подтверждает правильность их функционирования.

Полученные результаты будут использованы при разработке сложных вычислительных устройств с использованием типовых технических решений (в виде библиотечных элементов *IP Core*) в каче-

стве составных компонентов для реализации таких устройств на одном кристалле. Такие типовые решения в совокупности представляют собой «открытую» библиотеку файлов конфигураций, входящих в состав реконфигурируемой вычислительной системы с виртуальной архитектурой.

### Список литературы

1. Палагин А. В., Опанасенко В. Н., Сахарин В. Г. Проектирование цифровых устройств на кристаллах ПЛИС с использованием Core – ядер // Вестник СевГТУ: Сб. научных трудов. – Севастополь, 2004. – Вып. 57. – С. 33-47.
2. *Xilinx Data Book 2004* / Available at <http://support.xilinx.com/partinfo/databook.htm>.
3. Прэтт У. Цифровая обработка изображений. В 2-х книгах. Кн.2. – М.: Мир, 1982. – 792 с.
4. *VHDL'93. IEEE Standard VHDL Language Reference Manual*. IEEE Std 1076-1993.
5. Коугли П. М. Архитектура конвейерных ЭВМ: Пер. с англ. – М.: Радио и связь, 1985. – 360 с.
6. *ModelSim. Xilinx Tutorial* / Available at <http://www.xilinx.com>.