

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

V. Opanasenko, A. Lisovyi,
E. Soroka

THE DESIGNING, MODELING AND PHYSICAL VERIFICATION OF DIGITAL DEVICES

The soft-hardware complex to designing, modeling and verification arithmetic devise based on PLD is offered.

Пропонується програмно-апаратний комплекс для проектування, моделювання та верифікації арифметичних пристроїв на ПЛИС.

Предложен программно-аппаратный комплекс для проектирования, моделирования и верификации арифметических устройств на ПЛИС.

© В.Н. Опанасенко,
А.Н. Лисовый, Е.В. Сорока,
2008

УДК 004.315.5

В.Н. ОПАНАСЕНКО, А.Н. ЛИСОВЫЙ,
Е.В. СОРОКА

ПРОЕКТИРОВАНИЕ И ФИЗИЧЕСКАЯ ВЕРИФИКАЦИЯ ЦИФРОВЫХ УСТРОЙСТВ НА ПЛИС

Программируемые логические интегральные схемы (ПЛИС) используются для реализации цифровых устройств разного уровня сложности, что обеспечивается большим выбором кристаллов и наличием мощных программных средств проектирования, верификации.

Использование готовых технических блоков (IP-Core), моделирующих плат, которые кроме ПЛИС, содержат модули памяти, интерфейсы и другие компоненты, сокращает время разработки проектов и улучшает их качество.

На сегодняшний день фирма Xilinx [1] предлагает для построения моделирующих плат ПЛИС семейства Virtex-5, которое имеет несколько платформ: LXT, LX, SXT, FXT, TXT, оптимизированных под разные задачи или характеристики; SXT оптимизирована под цифровую обработку сигналов (ЦОС). Тактовая частота кристаллов этого семейства достигает 550 МГц, а также предусмотрена возможность обмена данными через шину PCI-Express. Одна из современных моделирующих плат на базе Virtex-5 – плата VenONE-PCIe фирмы Nallatech [2].

Верификация сложных проектов моделированием, как необходимый этап разработки, в отдельных случаях является недостаточной и требует физического моделирования, реализуемого с помощью целевой системы, которая служит для верификации и отработки проектов устройств, взаимодействующих с реальной окружающей средой.

Предлагается инструмент, позволяющий проектировщику организовать интерфейс

моделирующей платы с окружающей средой без привлечения прикладного программиста, разработчика драйвера, поиска (и анализа) контроллера интерфейса, а также использовать имеющиеся библиотечные элементы для построения функциональных устройств. В совокупности все это позволит существенно упростить процесс физической верификации проектируемых устройств.

Постановка задачи

Целевая система состоит из Host-компьютера с установленной системой САПР ПЛИС, системы моделирования, библиотеки функциональных компонентов (для проектирования функционального поля (ФП) и драйвера моделирующей платы), библиотеки для работы с драйвером, прикладной программы (для обмена данными с моделирующей платой). Для гибкости разработки прикладных программ или в случае необходимости другого пользовательского интерфейса программы проектировщику предоставляется библиотека для работы с драйвером (набор функций для работы с устройством). Моделирующая плата состоит из контроллера шины, ОЗУ, аппаратного блока интерпретирующего сигналы на контроллере, который читает/записывает данные с шины в регистр или память и формирует сигнал начала работы "load". После завершения обработки формируется "rdy" сигнал готовности результата. Проектировщик конфигурирует только ФП, обмен данными с компьютером осуществляется через блок регистров. Блок схема такой системы представлена на рис. 1.

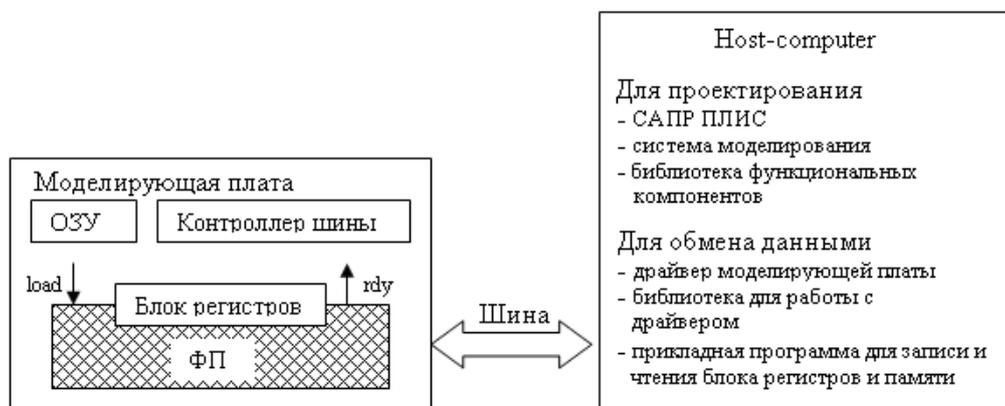


РИС. 1. Блок-схема компьютерной системы

В качестве примера построен инструмент на моделирующей плате (МП) ADS-XLX-SP3-EVL400 [3] фирмы Avnet, которая подключается в слот шины PCI. Используется кристалл ПЛИС семейства Spartan-3 типа XC3S400-FG456, асинхронная память ОЗУ (1Мбайт) фирмы Cypress.

Библиотека функциональных блоков постоянно расширяется. В настоящее время разработаны следующие блоки: сложения/вычитания, умножения, деле-

ния для чисел в формате с фиксированной и плавающей точкой, сложения/вычитания, умножения, деления для комплексных чисел, преобразования форматов с фиксированной точкой в плавающую и обратно. Некоторые блоки для одной и той же операции имеют несколько реализаций (функционируют по разным алгоритмам) и архитектур (конвейер, машина состояний, комбинационная схема). Все блоки формируют флаги специальных значений, которые имеют место в процессе вычислений, согласно стандартам (для плавающей точки IEEE-754).

Деление по методу гарвардской итерации

Метод гарвардской итерации – один из наиболее эффективных алгоритмов [4]. Этот алгоритм часто используется для реализации на аппаратных устройствах, так как его можно разбить на более простые составляющие части, что позволяет реализовать их отдельно и соединить в одно целое. Данный алгоритм может быть использован для нахождения квадратного корня. Универсальность метода в том, что при реализации на аппаратном и программном уровнях используется одинаковое количество итераций.

Метод гарвардской итерации предназначен в основном для деления чисел с плавающей точкой. В данной работе этот метод модифицируется и используется для деления целых чисел.

Алгоритм обеспечивает квадратичную сходимость результата. На каждой итерации множитель R одновременно перемножается на делимое и делитель. В результате этих действий делитель квадратично сходится к единице, а делимое – к результату.

Так как имеет место квадратичная сходимость и каждая итерация удваивает количество единиц после запятой, то новый делитель будет иметь вид $0,11xxx\dots$. В данном алгоритме умножение происходит до тех пор, пока делитель не будет равен единице ($0,1111\dots111$).

Алгоритм деления по методу гарвардской итерации:

- 1) нормализация делителя;
- 2) определение первого множителя R ;
- 3) умножение Y на R , вычисление Y_1 ;
- 4) умножение X на R , вычисление X_1 ;
- 5) вычисление X и Y из данных, полученных в п. 3) и п. 4);
- 6) переход к п. 3).

Структурная схема блока деления представлена на рис. 2.

Блоки “fd” – регистры для хранения входных операндов и сигнала загрузки “load”. В “conv_idle” осуществляется анализ входных операндов на равенство машинному нулю. В “conv_s1” – поиск старшей единицы в делителе и его «нормализация» (сдвиг влево до первой «1»). Блоки “conv_s2_alg”, “conv_s3_alg” – алгоритм по методу гарвардской итерации, “conv_last” – формирование результата.

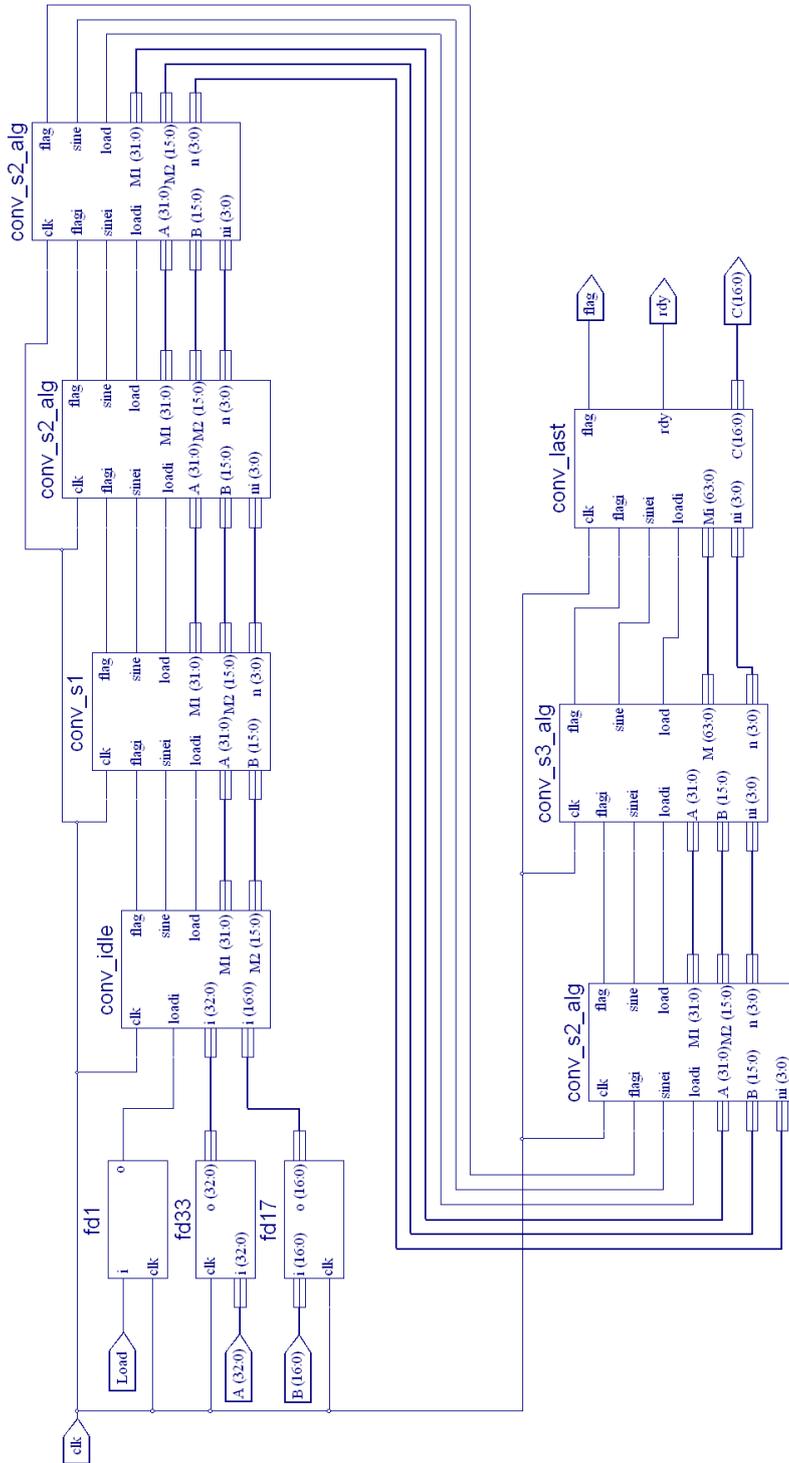


РИС. 2. Структурная схема блока деления

Алгоритм по методу гарвардской итерации в блоках “conv_s2_alg” и “conv_s3_alg” описывается процессом на VHDL [4]:

<pre> s2:process(clk,loadi) variable R : std_logic_vector (17 downto 0); variable M3 : std_logic_vector (33 downto 0); variable M3i : std_logic_vector (63 downto 0); begin if (clk'event and clk = '1') then FLAG<=FLAGi; --error sine<=sinei; load<=loadi; n<=ni; if loadi='1' then if FLAGi='0' then R:="10000000000000000000"-("00"&b); --R=2-y 18bit M3:=R(16 downto 0)*('0'&b); --M3=R*y 34bit M3i:=a*("0000000000000000"&R(16 downto 0)); --M3i=R*x 64bit M1<=M3i(47 downto 16); M2<=M3(31 downto 16); end if; end if; end if; end process; </pre>	Листинг
--	---------

Отличие блока “conv_s3_alg” от “conv_s2_alg” в том, что на выходные регистры блока выдается вектор M3 (с которого и формируется результат) вместо M1 и M2. Для формирования 16-разрядного результата необходимо 4 ступени из блоков, описывающих алгоритм по методу гарвардской итерации.

Разработанный модуль на кристалле x3s1000fgg456-4 может функционировать с частотой до 60 MHz и использует 4 % (375 Slices) кристалла.

Арифметика с комплексными числами

Комплексные числа используются при математическом описании многих задач физики и техники (в гидродинамике, аэромеханике, электротехнике, атомной физике и т. д.). Основные разделы классического математического анализа приобретают полную ясность и законченность только при использовании комплексных чисел, чем обуславливается центральное место, занимаемое теорией функций комплексного переменного. Решение широкого круга задач требует использования арифметических операций над комплексными числами, для аппаратной реализации которых в настоящее время используются программируемые логические интегральные схемы (ПЛИС) типа FPGA (Field Programmable Gate Array), которые позволяют реализовать не только алгоритм обработки данных, но и структуру устройства, выполнить заданный алгоритм.

Комплексными числами будем называть пары (x, y) вещественных (действительных) чисел x и y , для которых правила выполнения арифметических операций сложения, умножения и деления определены следующим образом:

$$z_1 + z_2 = (x_1 + x_2, y_1 + y_2), \quad z_1 z_2 = (x_1 x_2 - y_1 y_2, x_1 y_2 + x_2 y_1),$$

$$\frac{z_1}{z_2} = \left(\frac{x_1 x_2 + y_1 y_2}{x_2^2 + y_2^2}, \frac{x_2 y_1 - x_1 y_2}{x_2^2 + y_2^2} \right).$$

На рис. 3 показана блок-схема модуля сложения/вычитания двух комплексных чисел с одновременным получением результата действительной и мнимой частей на выходах. Блок LUT (Look-Up-Table) является комбинационным устройством, выполняющим операцию сложения или вычитания. Для суммирования (вычитания) используются функциональные генераторы, образующие разряды суммируемых чисел и располагаемые в одном ряду матрицы кристалла. Эти разряды соединяются между собой через логику ускоренного переноса.

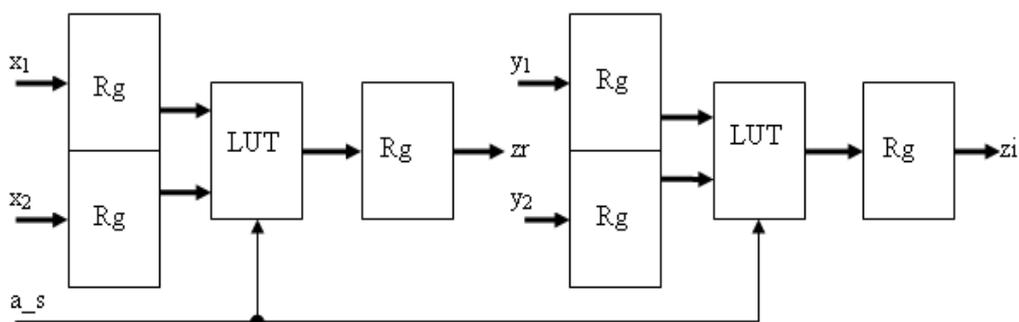


РИС. 3. Блок-схема модуля сложения/вычитания двух комплексных чисел

Цепи синхронизации, связанные с синхровходами входных и выходных регистров (Rg), на схеме условно не показаны. Состояние '0' сигнала a_s соответствует выполнению операции сложения, состояние '1' – операции вычитания.

В таблице приведены аппаратные (количество Slices, триггеров (FF) и LUT) и временные оценки модуля.

ТАБЛИЦА. Аппаратные и временные оценки модуля сложение/вычитание

Тип кристалла	Разрядность модуля	Slices	FF	LUT	Частота, МГц
XC3S400-5FG456	16	48	96	32	230
	32	96	192	64	192
XC5VLX30-3FF324	16	24	96	32	525
	32	56	192	64	455

Блок-схема модуля умножения двух комплексных чисел с одновременным получением результата действительной и мнимой частей на двух выходах представлена на рис. 4.

Операция умножения в проекте может быть выполнена путем создания соответствующего модуля различными способами: описанием поведения или структуры модуля на языке VHDL, включением в схему элемента умножения из библиотеки при работе со схематическим редактором, формированием IP-Core с помощью средства Core Generator.

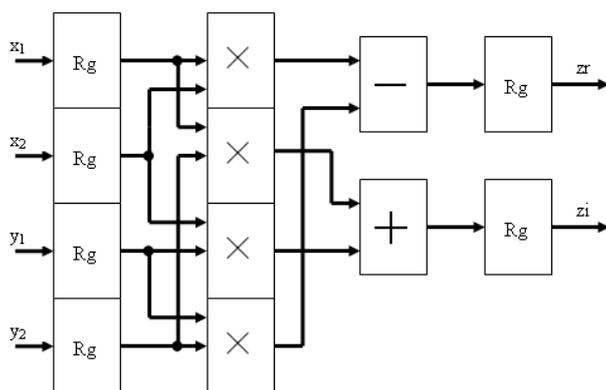


РИС. 4. Блок-схема модуля умножения двух комплексных чисел

Кроме модулей умножения, создаваемых с использованием сумматоров на основе LUT, в кристаллах серии Spartan-3 имеются встроенные умножители Mult18x18, а в кристаллах серии Virtex-5 – специализированные секции DSP48E, реализующие выполнение умножения 25x18 бит, а также другие функции, требуемые при цифровой обработке сигналов.

При использовании DSP48E в 16-разрядном модуле все операции выполняются внутри этих элементов, которые программируются на выполнение умножения с комплексными числами. На рис. 5 показана блок-схема модуля на базе четырех DSP48E, реализующая операцию умножения.

При использовании DSP48E в 32-разрядном модуле внутри этих элементов выполняются операции запоминания и умножения входных сигналов, операции сложения (вычитания) и запоминания выходных сигналов осуществляются элементами LUT и FF кристалла.

Если умножение выполняется с двумя 32-разрядными числами со знаком, представленными в дополнительных кодах, то для выполнения одной операции умножения требуется 4 умножителя (Mult18x18 или DSP48E), два сумматора и дополнительная логика, обеспечивающая масштабирование частичных произведений и сумм. Таким образом, общее число умножителей, реализующих операцию умножения с 32-разрядными числами, составляет 16.

Операция деления может быть выполнена IP-модулем, сформированным с помощью инструментальных средств Core Generator, или модулем, создаваемым компилятором Xilinx Synthesis Technology (XST) на основе поведенческого HDL-описания. IP-модуль деления с фиксированной точкой выполняет деление в двух режимах: с целым остатком и дробным остатком. Делимое в режиме деления с целым остатком равно сумме частного, умноженного на делитель, и остатка. Результат (частное и остаток) представлены на двух разных выходах

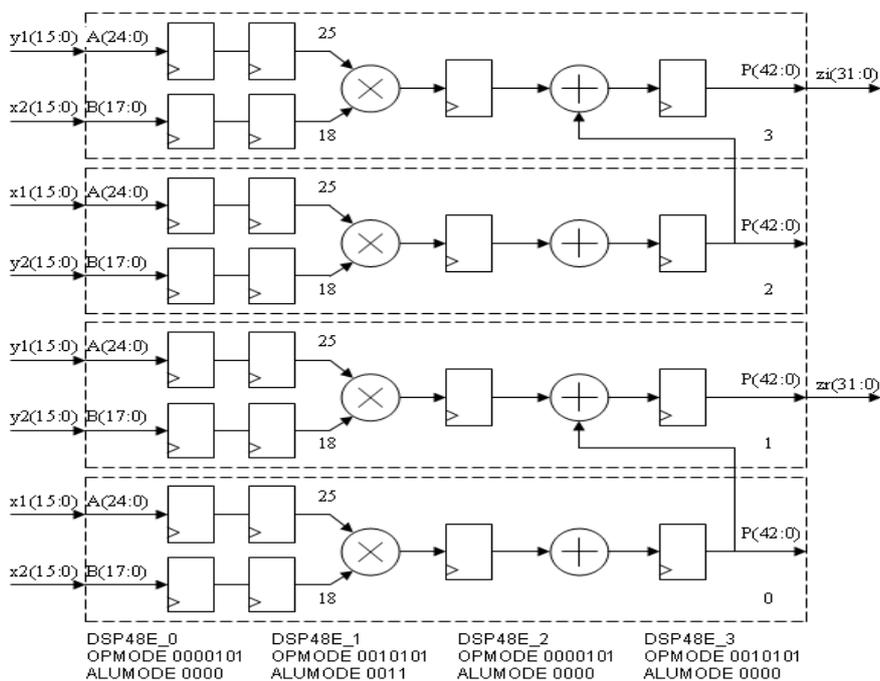


РИС. 5. Блок-схема модуля умножения на базе 4-х DSP48E

целыми числами. В режиме деления с дробным остатком последний равен частному от деления целого остатка на делитель и представлен на выходе целым числом в виде: $(\text{IntRmd}/\text{Divisor}) \times 2^F$ для чисел без знака и $(\text{IntRmd}/\text{Divisor}) \times 2^{(F-1)}$ – для чисел со знаком, где F – разрядность двоичного кода остатка ($F_{\max} = 32$). IP-модуль использует конвейерный принцип вычисления и позволяет выбрать количество синхросигналов (Clk), требуемых для выполнения одной операции деления из ряда 1, 2, 4, 8. Задержка появления результата на выходе от начала загрузки операндов определяется произведением $T_{\text{Clk}} \times L$, где L – число импульсов Clk, сформированных за этот отрезок времени. Величина такой задержки зависит от разрядности делимого (M), типа модуля, числа синхросигналов, требуемых для выполнения одной операции, разрядности двоичного кода остатка.

Временная диаграмма работы модуля деления изображена на рис. 6.

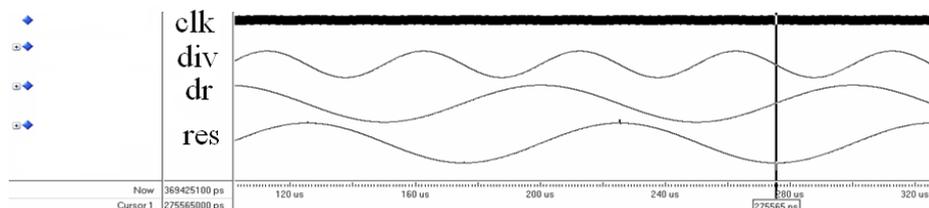


РИС. 6. Временная диаграмма работы модуля деления двух чисел со знаком

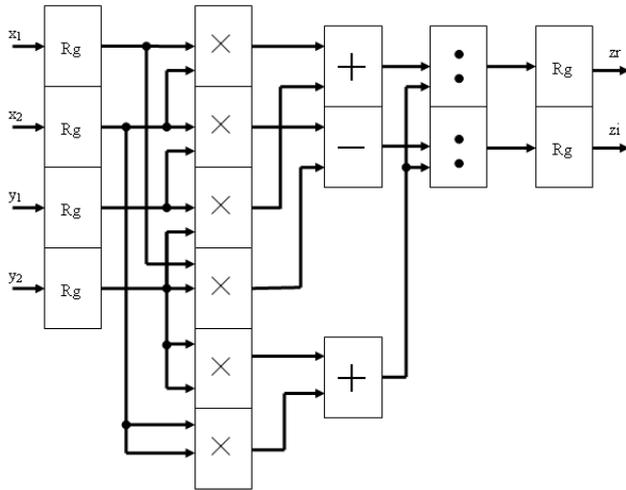


РИС. 7. Блок-схема модуля деления двух комплексных чисел

Блок-схема модуля деления изображена на рис.7. Модуль содержит 16-разрядные умножители с 32-разрядными выходами, 32-разрядные сумматоры и делители.

На рис. 8 показана структурная схема модуля умножения двух комплексных чисел в формате с плавающей точкой с одновременным получением результата действительной и мнимой частей на двух выходах. В модуле “kernl_flag” осуществляется анализ и генерируется общий флаг для действительной и мнимой частей

результата. Блок “add_sub_fp – сложение/вычитание в формате с плавающей точкой имеет конвейерную архитектуру и состоит из 6 ступеней, “kernl_flag” – комбинационная схема, поэтому модуль можно использовать как конвейер.

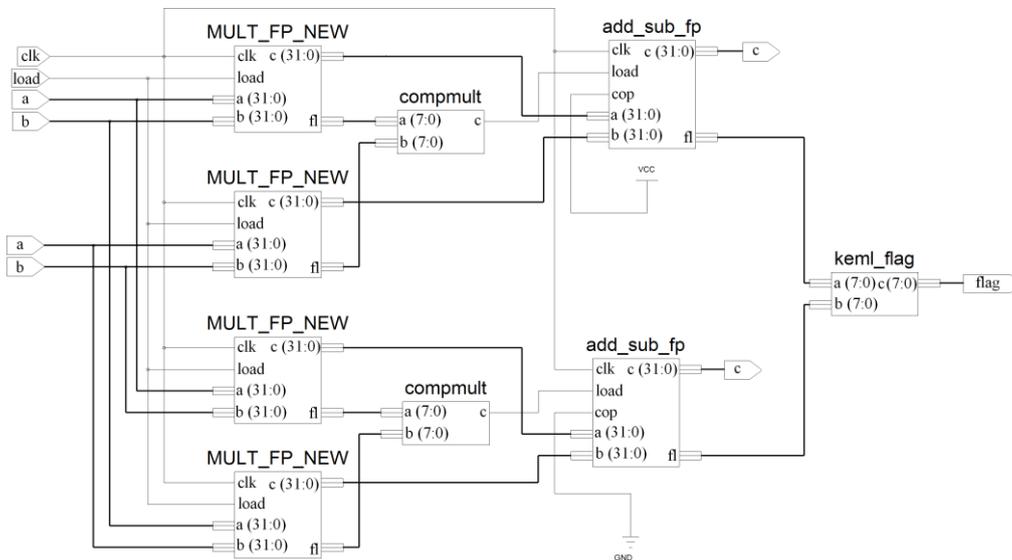


РИС. 8. Структурная схема модуля умножения двух комплексных чисел

Блок “Mult_fp_new” – комбинационная схема с регистрами на выходе, осуществляет перемножение чисел в формате с плавающей точкой; “Compmult” – осуществляет анализ регистров флагов блоков “Mult_fp_new” и генерирует сигнал загрузки в блок “add_sub_fp”. Модуль является конвейером с 7 ступенями.

При реализации рассмотренных блоков в кристалле XC3S400-FG456 были получены следующие оценки: модуль сложения/вычитания задействует 974 слайсов (27 % от общего числа) – 6-ступенчатый конвейер с периодом синхросигнала 13,3 нс; модуль умножения задействует 1220 слайсов (34 % от общего числа) – 7-ступенчатый конвейер с периодом синхросигнала 26,37 нс.

Выводы. Синтезированные модули верифицированы методом моделирования в системе ModelSim Xilinx Edition – МХЕ III 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) функциональных блоков в качестве составных компонентов для реализации таких устройств на одном кристалле.

1. Xilinx ISE 9 Software Manuals / Available at <http://www.xilinx.com>.
2. Virtex-5 LX Platform Overview / Available at <http://www.xilinx.com>.
3. Семенец В.В., Хаханова И.В., Хаханов В.И. Проектирование цифровых систем с использованием языка VHDL. – Харьков: ХНУРЕ, 2003. – 492 с.
4. Anderson S.F., Earle J.G., Goldschmidt R.F., Power D.M. The IBM system / 360 model 91: floating point execution unit // IBM J. Res. And developm. – 1967. – 11, N 1. – P. 34–53.

Получено 01.04.2008