

## ОРГАНИЗАЦИЯ КОНВЕЙЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ НА ОСНОВЕ VHDL – ТЕХНОЛОГИИ

Институт кибернетики им. В. М. Глушкова НАН Украины

*Предложена структурная реализация 32-разрядных модулей сложения с плавающей точкой, соответствующих стандарту IEEE-754, в элементном базисе ПЛИС типа FPGA, выполненная путем поведенческого описания алгоритма на языке VHDL. Реализована проверка функционирования модулей сложения методом моделирования в системе ModelSim Xilinx Edition - MXE II с помощью проверочного стенда.*

### Введение

Повышение производительности вычислительных устройств может быть получено при использовании конвейерного подхода к построению их структур.

При решении ряда задач целесообразно использование формата представления чисел с плавающей точкой вместо формата с фиксированной точкой, обладающего такими недостатками, как ограничение диапазона представления и потеря точности при делении двух больших чисел. Функциональные IP-блоки с плавающей точкой находят широкое применение при построении математических сопроцессоров, DSP – процессоров, встроенных арифметических сопроцессоров. В связи с этим рядом фирм разрабатываются собственные IP-блоки. Например, фирма Nallatech разработала ядра для обработки 32 разрядных операндов с плавающей точкой (согласно стандарту IEEE-754 [1]) под кристаллы серии Virtex [2].

Проблема проектирования арифметических устройств и алгоритмов для обработки операндов в формате с плавающей точкой при реализации таких операций, как сложение, умножение, и деление, актуальна и в настоящее время. Стандарт IEEE-754 дает наиболее общее представление для чисел с плавающей точкой в современных компьютерах. В данной статье предлагается подход к разработке устройства на основе современных кри-

сталлов ПЛИС, осуществляющего выполнение операции сложения в соответствии со стандартом IEEE-754, с использованием современных инструментальных средств для проектирования, моделирования и верификации.

Работа выполнена в рамках работ по программе ДНТП "Нові комп'ютерні засоби та технології інформатизації суспільства" Министерства образования и науки Украины.

### Постановка задачи

Конвейеризированное вычислительное устройство [3] характеризуется: количеством ступеней ( $b$ ) и временем синхронизации, которое равно наибольшему времени при прохождении всех ступеней ( $t_{cx}$ ). Зная массив входных данных ( $p$ ) можно определить коэффициент общей загруженности ( $K_{oz}$ ), который характеризует эффективность использования конвейера для заданного массива данных. Этот коэффициент можно определить отношением суммарного времени выполнения полезной работы всеми его ступенями к суммарному времени использования, включая вынужденные простои.

Обобщенная структура конвейера представлена на рис. 1. Каждая из ступеней – комбинационная схема со своим временем прохождения. В идеальной структуре конвейера времена прохождения через все ступени должны быть равны.

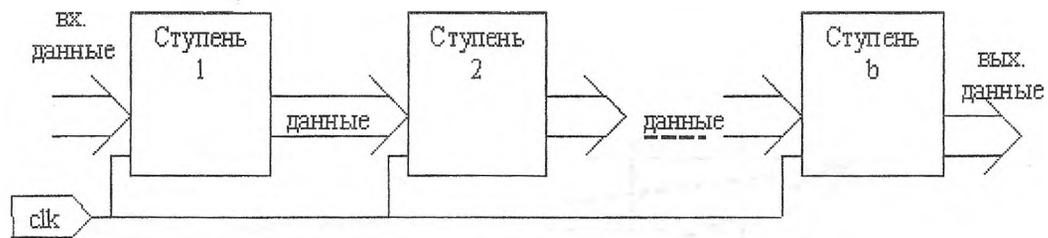


Рис. 1. Обобщенная структура конвейера

Время ( $T$ ) необходимое для обработки массива данных определяется суммой двух составляющих – временем загрузки конвейера ( $T_z$ ) и собственно временем работы ( $T_p$ ). Время загрузки – время, необходимое для того, чтобы первые обрабатываемые данные дошли до ступени  $b$ . Время работы – время, затраченное на прохождения массива данных через ступень  $b$ .

$$T = T_z + T_p,$$

$$T_z = (b - 1) \times t_{cx},$$

$$T_p = p \times t_{cx},$$

$$T = t_{cx} \times (b + p - 1).$$

Коэффициент общей загруженности конвейера [4] определяется выражением:

$$K_{oz} = \frac{t_{cx} \times b \times p}{t_{cx} \times (b - 1) \times b + t_{cx} \times b \times p},$$

$$K_{oz} = \frac{p}{p + b - 1}.$$

Данный коэффициент может принимать значения (0,1). Когда устройство не разделяется на ступени, т.е. функция реализуется за один такт, то  $K_{oz}=1$ .

Когда число ступеней конечное, а массив входных данных бесконечен, то:

$$\lim_{p \rightarrow \infty} K_{oz} = \lim_{p \rightarrow \infty} \frac{p}{p + b - 1} = 1.$$

График зависимости коэффициента от количества ступеней конвейера для трех вариантов массива входных данных ( $p_1=1, p_2=5, p_3=20$ ) приведен на рис. 2.

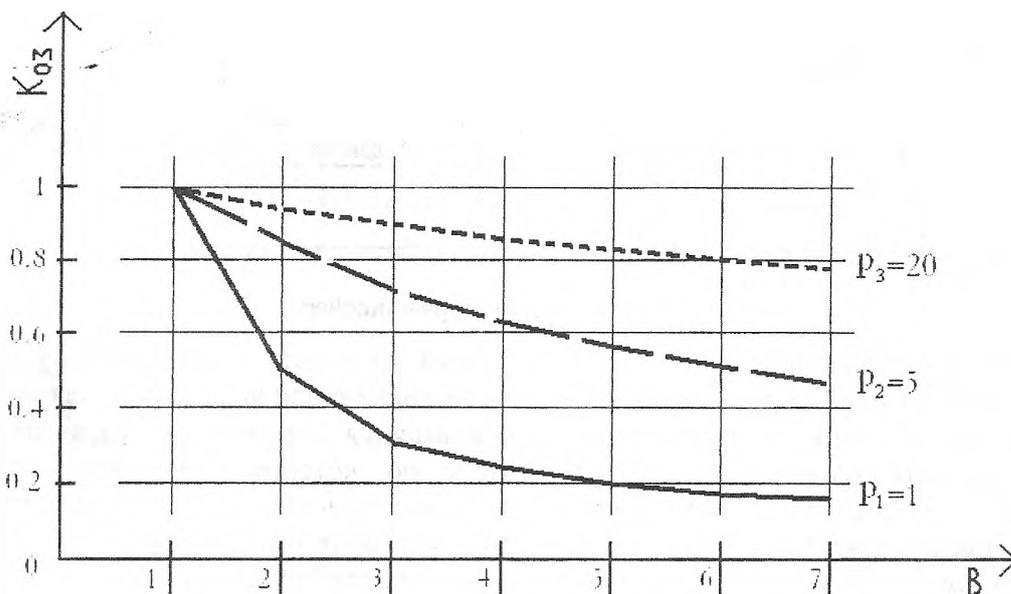
Конвейеры можно классифицировать по их возможностям. Однофункциональный конвейер – конвейер, который вычисляет одну функцию. Многофункциональный конвейер – конвейер, спо-

собный вычислять более одной функции, в том числе различных типов, для этого в нем имеется специальный управляющий вход, по которому указывают какую функцию выполнять. Структура конвейера может быть последовательной или последовательно-паралельной. В последовательной структуре нет параллельно работающих ступеней, в последовательно-паралельной имеются параллельно работающие ступени конвейера, которые реализуются последовательно.

#### Реализация модулей сложения

Рассмотрим реализацию модулей сложения с плавающей точкой в элементном базисе ПЛИС типа *FPGA* фирмы *Xilinx*.

Модуль сложения содержит четыре входа:  $a(31:0)$ ,  $b(31:0)$  – входные операнды, *Load* – сигнал загрузки операндов (состояние '1' соответствует операции загрузки, '0' – простой устройства), *Clk* – последовательность синхроимпульсов и три выхода:  $c(31:0)$  – результат сложения чисел 'a' и 'b', *Rdy* – сигнал готовности или окончания операции (состояние '1' соответствует окончанию операции, '0' – началу операции сложения), *Error(6:0)* – регистр ошибок. Регистр может принимать следующие значения: "0000000" – отсутствие ошибок, "0000001" – операнды 'a' и/или 'b' являются специальными значениями, "0000011" – результат ненормализованное число, "0000101" – результат машинный нуль, "0001001" – результат "+ бесконечность", "0010001" – результат "- бесконечность", "0100001" – результат QNAN, "1000001" – результат SNAN.

Рис. 2. График зависимости  $K_{03} = f(p, b)$ 

Блок сложения можно представить в виде четырех функциональных блоков, связанных между собой условными и безусловными переходами (рис. 3.).

В Блоке А анализируются входящие операнды 'а', 'б' на равенство специальным значениям. Алгоритм, по которому определяется флаг результата и результат, который может быть определен в данном блоке, строится на основе таблицы совместимости входных операндов с результатом (табл. 1.).

Из табл. 1 можно определить какой результат получается при заданных входных операндах 'а' и 'б'. Символ '?' в таблице означает, что в блоке А результат не может быть определен, и далее выполняется процедура в блоке Б, кроме тех случаев когда  $a=b$ , тогда результат машинный нуль.

Где  $A$  – нормализованное число;  $B$  – ненормализованное число;  $\pm 0$  – машинный нуль;  $+\infty$ ,  $-\infty$  – плюс и минус бесконечности;  $SN$ ,  $QN$ ,  $N$  – NAN-числа.

В Блоке Б выполняется формирование прогнозируемого порядка результата, и десятичных эквивалентов входных операндов.

Таблица 1.

Таблица совместимости входных операндов с результатом

a б	A	B	$\pm 0$	$+\infty$	$-\infty$	SN	QN
A	?	?	A	$+\infty$	$-\infty$	N	N
B	?	?	B	$+\infty$	$-\infty$	N	N
$\pm 0$	A	B	0	$+\infty$	$-\infty$	N	N
$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	N	N	N
$-\infty$	$-\infty$	$-\infty$	$-\infty$	N	$-\infty$	N	N
SN	N	N	N	N	N	N	N
QN	N	N	N	N	N	N	N

На данном этапе может быть получен результат, если степень одного из входных операндов больше степени второго на 24. В этом случае десятичный эквивалент операнда с меньшей степенью будет равным нулю, и дальнейшее сложение этих эквивалентов не имеет смысла, результатом сложения будет операнд с большей степенью. Если разница степеней меньше 24 то прогнозируемая степень определяется большей степенью одного из операндов, а десятичный эквивалент операнда с меньшей степенью сдвигается на количество раз равное разнице степеней.

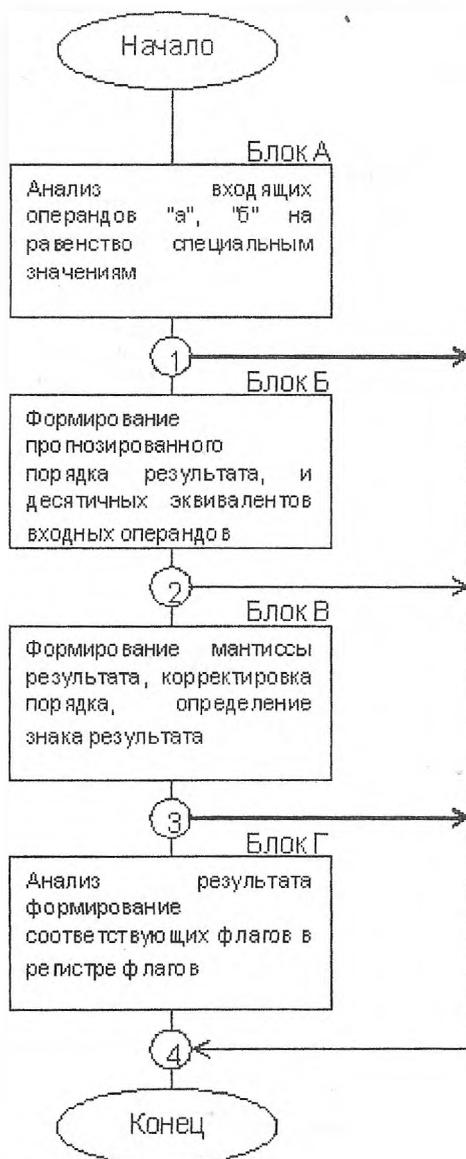


Рис. 3. Блок-схема работы устройства

В Блоке В формируется мантисса результата, корректируется порядок, определяется знак результата. В данном блоке может быть определен результат, который не входит в диапазон допустимых значений используемого формата, тогда формируется флаг ошибки.

В Блоке Г происходит анализ результата и формируется соответствующий флаг в регистре флагов.

Предложены следующие подходы для реализации блока сложения:

Архитектура модуля *ADD1* описана на *VHDL* [5]. Особенностью данного модуля есть использования функций сдвига (*SHL*, *SHR*), что входят в состав базовых библиотек и сложения десятичных эквивалентов входных операндов в прямом коде. Структура модуля *ADD2* оптимизирована по аппаратным ресурсам. Сложение десятичных эквивалентов выполняется в обратном коде. Модуль *ADD3* является комбинацией структур *ADD1* и *ADD2*. Сложение происходит в обратном коде.

Предложен ряд конвейерных организаций блока сложения.

Структура *ADD4* базируется на основе *ADD2*. Алгоритм функционирования разбит на 5 ступеней, связанных между собой элементами памяти. По синхросигналу срабатывает только первая ступень, а последующие срабатывают от фиксатора, в котором установлен результат предыдущей ступени. Данный конвейер является однофункциональным последовательно-паралельным (ОПП).

*ADD5* отличается от *ADD4* тем, что все ступени срабатывают от синхросигнала. Часть комбинационной логики, которая служит для связи между ступенями, описана на *VHDL* и встроенная в ступени. Этот конвейер также является (ОПП).

Существенно отличен от двух предыдущих конвейер *ADD6*. Управляющие схемы и регистры реализованы в самих ступенях конвейера, этот конвейер является последовательным (ОПС).

Модули сложения, синтезированы и реализованы в кристалле серии *VirtexE - XCV400E-6BG432* с помощью средств *XST VHDL (Xilinx Synthesis Technology)* и *ECS* системы проектирования *Xilinx ISE (Integrated Synthesis Environment) Foundation*.

Таблица 2. Оценочных характеристики модулей сложения

Имя модуля	Оценка аппаратных ресурсов, $N$ (SLICES)	Оценка быстрой работы, $T_{clk}$ (ns)	Временные затраты для обработки массива данных $p$ (ns)			Наличие и тип конвейера
			$p_1=2$	$p_2=20$	$p_3=90$	
Add1	987	46.706	93.412	934.12	4203.54	-
Add2	784	54.859	109.718	1097.18	4937.31	-
Add3	809	59.651	119.302	1193.02	5368.59	-
Add4	784	13.491	80.946	323.784	1268.154	ОПП
Add5	603	19.885	119.31	477.24	1869.19	ОПП
Add6	520	19.523	117.138	468.552	1835.162	ОПС

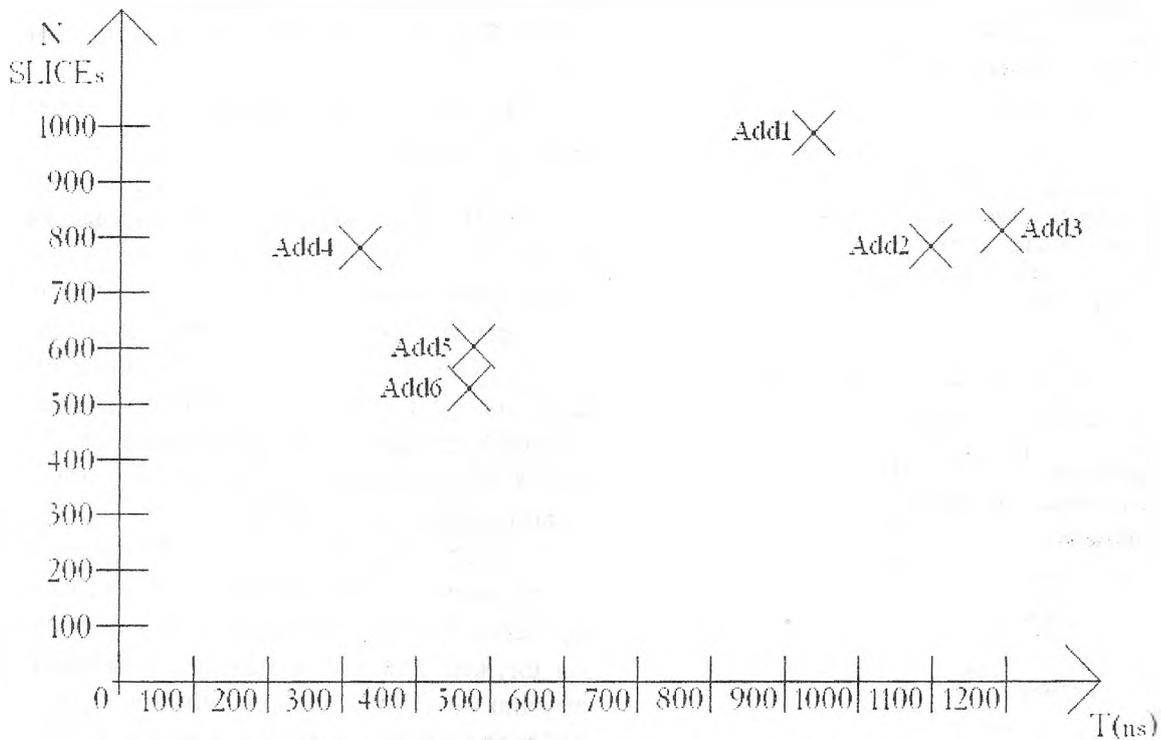
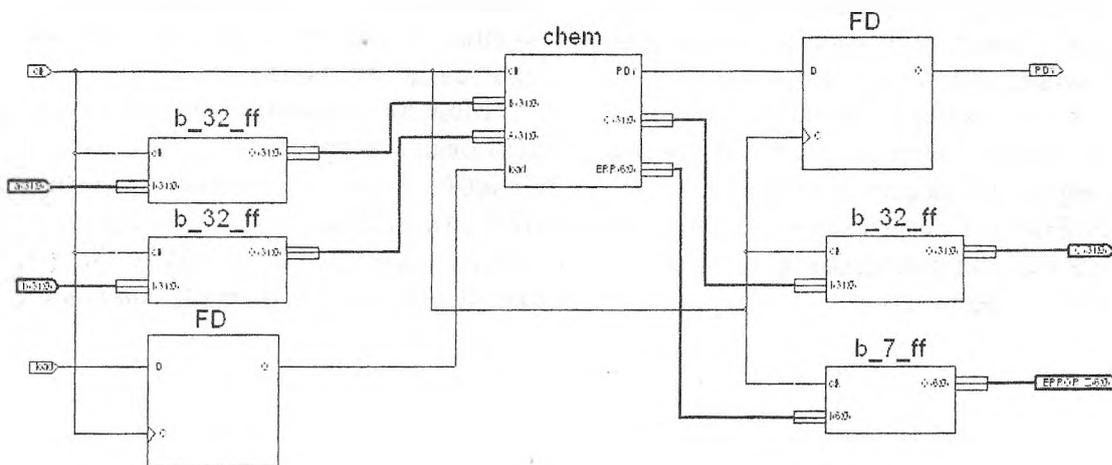
Рис. 4. График зависимости  $N=f(T, p)$ , при  $p=20$ 

Рис. 5 Стэнд для проверки модулей сложения

В табл. 2 приведені данні об апаратних ресурсах, максимальному швидкодії розроблених модулів і часові витрати для обробки масивів даних р.

**Верифікація модулів складення**

Перевірка функціонування модулів складення здійснюється методом мо-

делювання в системі *ModelSim Xilinx Edition – MXE II*.

На рис. 5 представлена схема стенда, виконана засобами схематичного редактора *Engineering Capture System (ECS)*, що входить до складу системи *Xilinx ISE Foundation*. В табл. 3 представлені результати верифікації.

Таблиця 3. Исходные и результирующие данные системы моделирования

/testbench/uut/xlxi_7/a	/testbench/uut/xlxi_7/b	/testbench/uut/xlxi_7/c
01001011000000000000000000000000	01000000000000000000000000000000	01001011000000000000000000000010
01001011000000000000000000000000	01000000100000000000000000000000	01001011000000000000000000000100
01001011000000000000000000000000	01000001000000000000000000000000	010010110000000000000000000001000
01001011000000000000000000000000	01000001100000000000000000000000	0100101100000000000000000000010000
01001011000000000000000000000000	01000010000000000000000000000000	01001011000000000000000000000100000
01001011000000000000000000000000	01000011000000000000000000000000	010010110000000000000000000001000000
01001011000000000000000000000000	01000011100000000000000000000000	0100101100000000000000000000010000000
01001011000000000000000000000000	01000100000000000000000000000000	01001011000000000000000000000100000000
01001011000000000000000000000000	01000100100000000000000000000000	010010110000000000000000000001000000000
01001011000000000000000000000000	01000101000000000000000000000000	0100101100000000000000000000010000000000
01001011000000000000000000000000	01000101100000000000000000000000	0100101100000000000000000000010000000000
01001011000000000000000000000000	01000110000000000000000000000000	01001011000000000000000000000100000000000
01001011000000000000000000000000	01000110000000000000000000000000	010010110000000000000000000001000000000000
01001011000000000000000000000000	01000110100000000000000000000000	0100101100000000000000000000010000000000000
01001011000000000000000000000000	01000111000000000000000000000000	01001011000000000000000000000100000000000000
01001011000000000000000000000000	01000111100000000000000000000000	010010110000000000000000000001000000000000000

**Выводы**

На основе данных из табл. 2 можно сделать следующие выводы. Реализации *Add1*, *Add2* и *Add3* представляются последовательными структурами без конвейеризации. В результате моделирования и реализации в ПЛИС, данные структуры по своим характеристикам заведомо хуже структур *Add4*, *Add5* и *Add6* (конвейерные структуры с пятью ступенями).

Структурная реализация *Add4* требует максимум аппаратных ресурсов и обладает максимальным быстродействием. Реализация *Add6* за счет совмещенных аппаратных узлов имеет меньшие аппаратные затраты, но обладает меньшим быстродействием, по сравнению с реализацией *Add4*. Реализация *Add5* требует больших временных и аппаратных затрат по сравнению с *Add6*.

Таким образом, для окончательного выбора варианта аппаратной реализации модулей сложения остаются два варианта – *Add4* и *Add6*.

Верификация всех вариантов модулей сложения с помощью разработанного стенда подтвердила правильность функционирования предложенных структур.

**Список литературы**

1. *Hollash S.* IEEE Standard 754 Floating Point Numbers / S. Hollash/ Available at [http:// IEEE / IEEE Standard 754 Floating-Point.htm](http://IEEE/IEEE%20Standard%20754%20Floating-Point.htm).
2. Available at <http://xilinx.com>
3. *Козги П. М.* Архитектура конвейерных ЭВМ: Пер. с англ. – М.: Радио и связь, 1985. – 360 с.
4. *Воеводин В.В.* Математическая модель конвейерных вычислений // Конвейеризация вычислений: Сб. научных трудов Научного совета по проблеме "Кибернетика": – Москва, 1986. – С. 36-62.
5. *Семенец В.В., Хаханова И.В., Хаханов В.И.* Проектирование цифровых систем с использованием языка VHDL. – Харьков: ХНУРЕ, 2003. – 492 с.