

УДК 621.382

## ВОЗМОЖНОСТИ ПЛИС АРХИТЕКТУРЫ FPGA ДЛЯ РЕАЛИЗАЦИИ ОПЕРАЦИИ СЛОЖЕНИЯ В ФОРМАТЕ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

**Кармышова Ю.В., Ушенина И.В.**

*Пензенский государственный технологический университет, Пенза, e-mail: Alexey314@yandex.ru*

В статье выполнен сравнительный анализ двух подходов к реализации на программируемых логических интегральных схемах (ПЛИС) архитектуры FPGA сумматоров, работающих в формате с плавающей запятой. Один из подходов заключается в использовании исключительно программируемой логики FPGA, а именно – конфигурируемых логических блоков. Второй подход заключается в использовании блоков цифровой обработки сигналов, встроенных в FPGA. Для подготовки исходных описаний сумматоров использован модуль интеллектуальной собственности Floating Point Operator, доступный в среде проектирования ISE Design Suite фирмы Xilinx. В рамках обоих подходов исследовано, каким образом латентность сумматора влияет на его производительность и ресурсоемкость. Показано, что увеличение латентности в большинстве случаев позволяет повысить тактовую частоту сумматора, но при этом общее время выполнения операции суммирования может возрасти. При реализации вычислений на микроконтроллерах или сигнальных процессорах оптимизация операции сложения, равно как и других операций, выполняется производителем элементной базы. В тех случаях, когда вычисления реализуются на ПЛИС архитектуры FPGA, оптимизация выполняемых операций является задачей разработчика устройств цифровой обработки информации (УЦОИ). Целью оптимизации, как правило, является достижение максимальной производительности (рабочей частоты) УЦОИ.

**Ключевые слова:** ПЛИС FPGA, сумматор, формат с фиксированной запятой, формат с плавающей запятой

## FPGA CAPABILITIES TO IMPLEMENT FLOATING-POINT ADDERS

**Karmyshova U.V., Ushenina I.V.**

*Penza State Technological University, Penza, e-mail: Alexey314@yandex.ru*

In this paper, the author comparatively analyses two approaches to FPGA implementation of floating-point adders. One of these approaches consists of only using FPGA programmable logic sources, namely, configurable logic blocks. Another approach consists of using DSP blocks embedded in FPGA. To prepare source codes for adders, Floating Point Operator IP core was used. This IP core is available in Xilinx ISE Design Suite. Under both approaches, it was studied how the adder latency affects its performance and logic source consumption. It was shown that in most cases the growth of adder latency allows its clock frequency to rise, but the absolute time required to perform the summation can increase. When implementing calculations on microcontrollers or signal processors, the optimization of the operation of addition, as well as other operations, is performed by the manufacturer of the element base. In cases where the calculations are implemented on the FPGA architecture, optimization of the operations performed is the task of the developer of digital information processing devices (UCIs). The purpose of optimization, as a rule, is to achieve maximum performance (operating frequency) of the UCI.

**Keywords:** FPGA, adder, fixed-point format, floating-point format

При аппаратной реализации вычислений числа могут быть представлены в форматах с фиксированной или с плавающей запятой. Вычислительные устройства, работающие в формате с фиксированной запятой, менее требовательны к объему аппаратных ресурсов. Однако вычисления в формате с фиксированной запятой не всегда могут обеспечить достаточную точность, а также подвержены влиянию эффектов конечной разрядности. Формат с плавающей запятой более устойчив к сбоям и дает высокую точность вычислений, но его применение сопровождается большим расходом аппаратных ресурсов и (или) снижением скорости вычислений.

Наиболее востребованным среди арифметических устройств является сумматор (вычитатель), который используют, например, до 50% устройств цифровой обработки сигналов (ЦОС). Вместе с тем, суммирование в формате с плавающей запятой наиболее сложно для реализации [1].

При реализации вычислений на микроконтроллерах или сигнальных процессорах оптимизация операции сложения, равно как и других операций, выполняется производителем элементной базы. В тех случаях, когда вычисления реализуются на ПЛИС архитектуры FPGA, оптимизация выполняемых операций является задачей разработчика. Целями оптимизации, как правило, являются:

- достижение максимальной производительности (рабочей частоты) устройства;
- минимизация латентности (задержки появления результатов вычислений относительно момента поступления исходных данных);
- минимизация ресурсоемкости устройства.

В данной статье рассмотрено, каким образом можно реализовать сложение в формате с плавающей запятой на современных FPGA и исследовано влияние на достижение перечисленных выше целей таких фак-

торов, как используемые ресурсы FPGA и латентность сумматоров.

Ресурсы FPGA, пригодные для реализации сумматоров. Операция сложения в формате с плавающей запятой может быть реализована на FPGA в двух вариантах: с использованием исключительно ресурсов программируемой логики или с привлечением блоков цифровой обработки сигналов (ЦОС).

При реализации сумматоров на программируемой логике важную роль в ускорении вычислений играют цепи переноса (carry logic), находящиеся в конфигурируемых логических блоках FPGA. Кроме того, требуется большое количество табличных преобразователей (LUT).

Ожидаемым преимуществом использования ЦОС-блоков является компактность размещения элементов сумматора на FPGA, а также более высокая производительность за счет выделенных логических и трассировочных ресурсов. Однако нужно учитывать, что в современных FPGA, за исключением [2], ЦОС-блоки ориентированы на вычисления в формате с фиксированной запятой. Работа в формате с плавающей запятой возможна, но требует использования наряду с ЦОС-блоками программируемой логики.

Способы описания и инструменты проектирования сумматоров. Для выполнения исследования выбрана FPGA XC7A200T-3 фирмы Xilinx. Проектирование сумматоров выполнялось в среде ISE Design Suite 14.7.

Производительность, ресурсоемкость и другие характеристики вычислительных устройств, работающих в формате с плавающей запятой, во многом зависят от способа их описания. В [3] отмечается, что одним из наиболее эффективных способов исходного описания подобных устройств является использование настраиваемых модулей интеллектуальной собственности Floating Point Operator и инструмента Core Generator. Такой способ описания выбран и в данной работе.

Кроме того, Core Generator предоставляет графический интерфейс для настройки параметров модуля интеллектуальной собственности, в том числе для указания используемых ресурсов FPGA и латентности сумматоров, выраженной в количестве тактовых импульсов (рисунки).

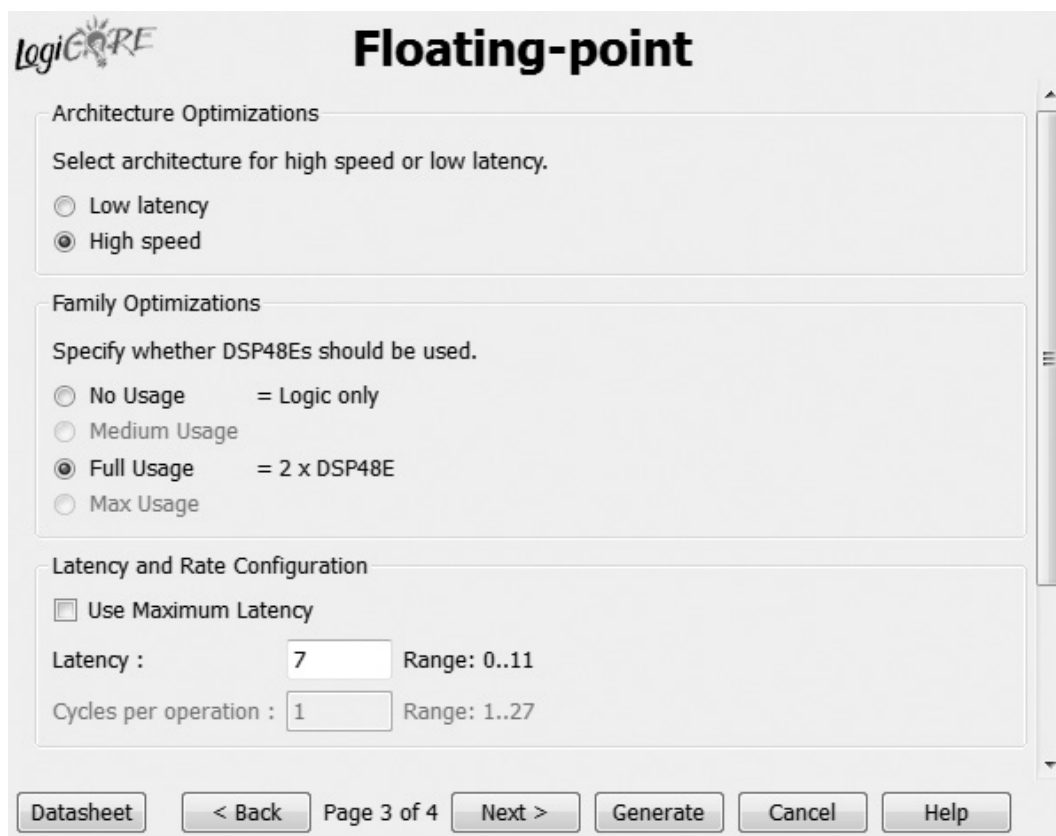
Сумматоры, рассмотренные в данной работе, рассчитаны на числа, представленные в соответствии со стандартом IEEE 754: для знака числа отводится 1 разряд, для экспоненты – 8 разрядов, для мантиссы – 23 разряда без учета целой части.

Чтобы временной анализатор среды ISE Design Suite выполнил оценку производительности проектируемого устройства, необходимо наложить на него временные ограничения [4]. Максимальной производительностью реализованных на ПЛИС устройств является производительность отдельных логических ресурсов (конфигурируемых логических блоков, блоков ЦОС). При выполнении исследования на проекты сумматоров накладывалось временное ограничение PERIOD, куда в качестве периода тактового сигнала подставлялись значения, взятые из [5].

Результаты оценки производительности и ресурсоемкости сумматоров при различных настройках оценивались по итогам размещения и трассировки соответствующих проектов с помощью инструмента PlanAhead.

Результаты исследований и их обсуждение. В табл. 1 и 2 приведены результаты реализации сумматоров исключительно на программируемой логике FPGA и с использованием блоков ЦОС при различных настройках латентности. В таблицах показаны результаты оценки производительности (максимальной тактовой частоты  $f_{max}$ ) сумматоров, а также требуемое для их реализации количество D-триггеров (FF) и табличных преобразователей (LUT). Количество блоков ЦОС не показано в таблице 2, так как в рамках исследования требовалось 2 блока при любых настройках латентности.

Из табл. 1 следует, что повышение латентности сумматора, реализованного на программируемой логике FPGA, сопровождается повышением его максимальной тактовой частоты  $f_{max}$ . Однако если выразить латентность в единицах измерения времени, подставив при расчетах полученные значения  $f_{max}$ , будет очевидно, что достигнутый прирост производительности не оправдывает дополнительного расхода триггеров, так как абсолютное время выполнения сложения с повышением количества тактов увеличивается. В целом можно отметить, что производительность сумматоров заметно ниже предельной производительности отдельных конфигурируемых логических блоков FPGA [5]. В первую очередь это связано с неизбежными задержками распространения сигналов на программируемых соединениях FPGA. Однако потери производительности могут быть связаны и с тем, что Floating Point Operator не является специализированным средством для реализации сумматоров. Отдельные операции, выполняемые при суммировании в формате с плавающей запятой, могут быть реализованы не оптимально.



*Настройки модуля Floating Point Operator*

**Таблица 1**  
 Результаты реализации сумматоров на программируемой логике FPGA

Латентность (такт. импульсов)	2	4	6	8	10	12
$f_{max}$ , МГц	147	267	356	331	476	452
LUT	417	387	356	379	379	389
FF	73	181	275	394	474	541

**Таблица 2**  
 Результаты реализации сумматоров с использованием блоков ЦОС FPGA

Латентность (такт. импульсов)	1	3	5	7	9	11
$f_{max}$ , МГц	628	155	195	218	334	366
LUT	185	237	212	223	211	209
FF	32	74	127	228	264	319

Из табл. 2 следует, что при использовании блоков ЦОС для реализации сумматора наилучшие показатели производительности и ресурсоемкости обеспечивает латентность, равная одному такту. Повышение латентности приводит к спаду производительности, который можно объяснить увеличением доли программируемой логики в составе сумматора и усложнением его размещения и трассировки. Данные, представленные в табл. 2, позволяют также заключить, что использование блоков ЦОС значительно сокращает требуемый объем триггеров и табличных преобразователей.

### Заключение

На основе результатов, представленных в табл. 1 и 2, можно заключить, что в большинстве случаев предпочтительным способом реализации на FPGA операции арифметического сложения в формате с плавающей запятой является использование блоков ЦОС. В случае использования ресурсов про-

граммируемой логики устанавливать высокую латентность сумматора имеет смысл, если от устройства, в состав которого он входит, требуется высокая частота обработки данных.

### Список литературы

1. Айфичер Э.С., Джервис Б.У. Цифровая обработка сигналов: практический подход, 2-е изд.: пер. с англ. – М.: Издательский дом «Вильямс», 2008. – 992с.
2. Intel Stratix 10 FPGA Features. – URL: <https://www.intel.com/content/www/us/en/products/programmable/fpga/stratix-10/features.html> (дата обращения 11.02.2019).
3. Ушенина И.В., Сальников И.И., Слесарев Ю.Н., Михайлов П.Г., Ремонтов А.П. Подходы к описанию вычислений в формате с плавающей запятой, реализуемых на ПЛИС // XXI век: итоги прошлого и проблемы настоящего плюс. – 2018. – №4 (44). – С. 122–127.
4. Ушенина И.В. Использование временных ограничений PERIOD и OFFSET при проектировании цифровых устройств на ПЛИС фирмы Xilinx // Компоненты и технологии. – 2013. – №5. – С. 97–106.
5. Artix 7 FPGAs Data Sheet: DC and AC switching characteristics. – URL: [https://www.xilinx.com/support/documentation/data\\_sheets/ds181\\_Artix\\_7\\_Data\\_Sheet.pdf](https://www.xilinx.com/support/documentation/data_sheets/ds181_Artix_7_Data_Sheet.pdf) (дата обращения 11.02.2019).