

# Методы разработки аппаратных блоков для ПЛИС на платформе USRP с использованием технологии RFNoC

Федор СЕРКИН  
Николай ВАЖЕНИН  
Александр НЕСТЕРОВ

Использование технологии программно-определяемого радио существенно ускоряет и упрощает процедуру написания модулей, позволяя автоматически формировать необходимую обвязку блоков и тестовую среду. В статье на примере разработки простейших модулей усилителя и программируемого КИХ-фильтра изложены основные особенности проектирования модулей для ПЛИС с помощью указанных средств.

## Введение

Классический подход к проектированию алгоритмов на программируемой логической интегральной схеме (ПЛИС) предполагает создание проекта с нуля в рамках некоторой отладочной платы, например KC705 в случае с кристаллами серии Kintex-7. Однако существует и более эффективный путь проектирования — применение технологии программно-определяемого радио (ПОР). Устройства, использующие данную технологию, как правило, состоят из трех базовых частей: радиотракта, ПЛИС и интерфейса связи с компьютером общего назначения. В данном случае такой продукт может поставляться с проектом для ПЛИС, в котором заложены специфические для радиомодемов блоки управления АЦП/ЦАП, аналоговой частью и алгоритмы цифровой обработки сигналов [1]. На примере USRP X310 покажем далее, как в рамках такой аппаратно-программной среды возможна реализация пользовательских модулей обработки сигналов на ПЛИС.

## Блок-схема и описание функциональных возможностей USRP X310

Вначале обратимся к самому устройству, фотография которого и упрощенная блок-схема показаны на рис. 1 и 2 соответственно.

Это устройство представляет собой отладочную плату с ПЛИС и подключенными радиотрактами в корпусе. Устройство оборудовано высокоскоростными АЦП/ЦАП (до 200 МГц), блоками формирования стабильных тактовых сигналов, в том числе с возможностью тактирования извне, а также интерфейсной частью, отвечающей за обмен и управление потоками данных с компьютером.



Рис. 1. Внешний вид устройства USRP X310

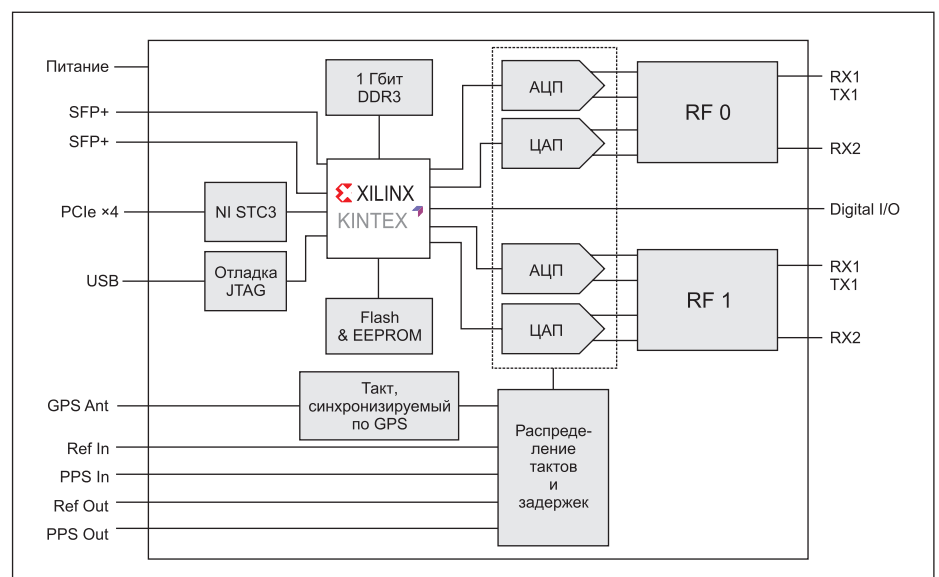


Рис. 2. Упрощенная блок-схема USRP X310

В основе метода проектирования лежит концепция RF Network-On-Chip (RFNoC) [1, 3, 4] компании Ettus Research, предлагающая использование некоторой шины для обмена данными внутри ПЛИС с подключенными к ней блоками обработки сигналов. Программная среда UHD (Universal Hardware Driver) и GNU (GNU is Not Unix) Radio позволяют программно настраивать данные блоки на шине с помощью компьютера общего назначения, подключенного по Ethernet.

На рис. 3 изображен классический пример использования RFNoC [1], в котором ПЛИС запрограммирована таким образом, что на шине присутствуют три блока: управление радиотрактом и сбор отсчетов с АЦП, модуль БПФ и абстрактный вычислительный модуль. С помощью GNU Radio и UHD эта конфигурация настроена таким образом, что данные с выхода АЦП поступают на модуль БПФ, а затем на компьютер общего назначения в среду GNU Radio.

Реализация подобных механизмов предусматривает некую условно стандартизированную структуру пользовательского модуля (рис. 4) [1]. Такой модуль должен включать интерфейс к шине или NOC Shell, а также внутренний шинный интерфейс к блокам пользовательского модуля — AXI Wrapper, к которому непосредственно подсоединяется пользовательский IP. Нужно отметить, что данные следует передавать по AXI, регистры же пользовательского модуля доступны напрямую с NOC Shell. Подобная реализация пользовательских модулей позволяет добавлять достаточно большое число модулей (до 10) в стандартную сборку USRP X310. Кроме того, интерфейс NOC Shell предусматривает передачу пакетов напрямую в IP пользователя, но для этого необходимо обратиться к специальному протоколу CHDR (Compressed HeaDeR). На рис. 5 показана упрощенная схема созданного блока цифрового программируемого усилителя. Можно видеть, что данные передаются с помощью AXI-шины, а конфигурация усилителя осуществляется посредством NOC Shell.

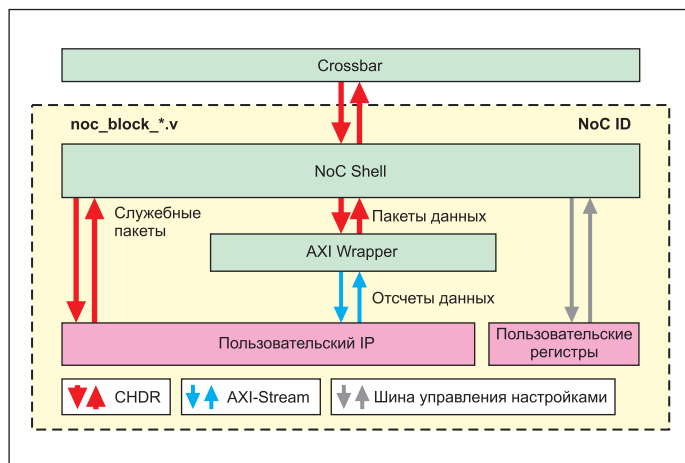


Рис. 4. Стандартизированная структура пользовательского модуля

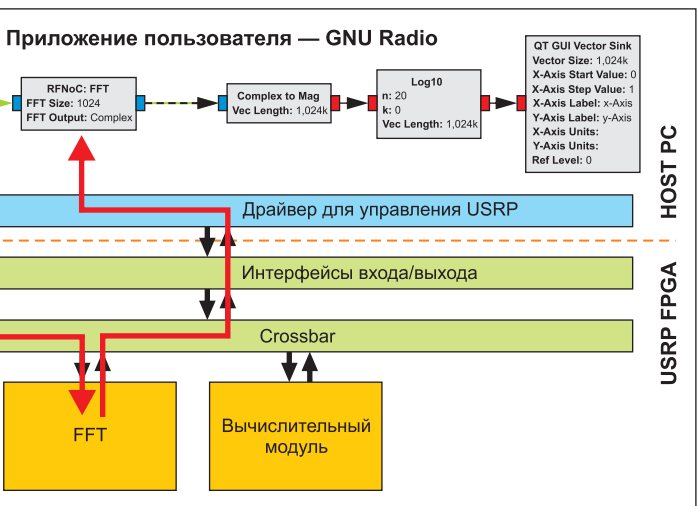


Рис. 3. Классический пример использования RFNoC

### Описание среды тестирования пользовательских модулей

При разработке модулей по классической схеме на отладочной плате написание тестовых модулей представляет собой отдельную задачу достаточно высокой сложности, поскольку необходимо создать все условия для тестирования алгоритма, в которых он будет работать в реальных условиях. Зачастую это оказывается затруднительно. Обратимся к случаю проектирования с помощью RFNoC.

Концепция RFNoC позволяет на базе библиотеки RFNoC создавать тестовые модули для пользовательских модулей и тестировать их непосредственно в тех условиях, в которых они будут функционировать в ПЛИС, то есть в данном случае формируется шина и все интерфейсы, окружающие пользовательский модуль. На примере модуля цифрового программируемого усилителя опишем основные элементы и процесс его разработки. На рис. 6 можно видеть, что созданный с помощью скриптов модуль gain помещается на шину Crossbar, тестовый модуль же

представляет собой другой подобный модуль на шине, который с помощью стандартных методов работы с USRP, а также библиотеки RFNoC управляет, передает и забирает данные из модуля. Создание тестового модуля представляет собой подачу определенного набора команд, в результате которых формируется и запускается среда моделирования и выдаются результаты. Временные диаграммы можно описать через серию последовательных этапов.

Последовательность действий тестового модуля:

1. Данные передаются с помощью библиотеки RFNoC из тестового модуля через AXI Wrapper на выход NOC Shell управляющего блока.
2. Данные передаются по Crossbar от тестового блока в пользовательский модуль gain.
3. Данные уже приняты блоком gain, и теперь NOC Shell пользовательского модуля передает данные внутрь модуля.
4. Передача данных через AXI и сам процесс работы блока.
5. NOC Shell пользовательского модуля передает данные на Crossbar.

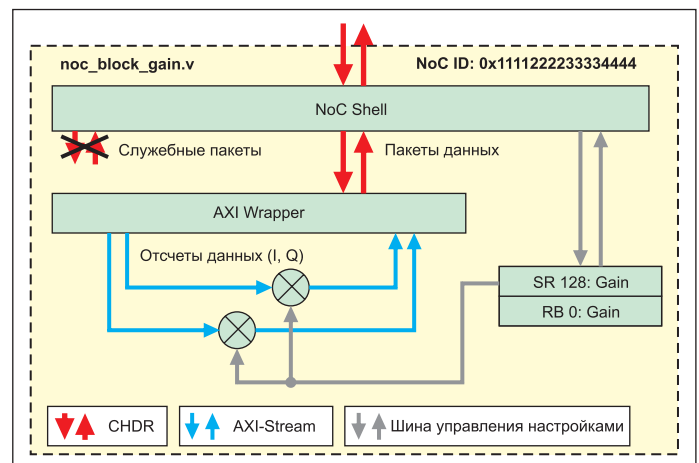


Рис. 5. Упрощенная схема созданного блока цифрового программируемого усилителя

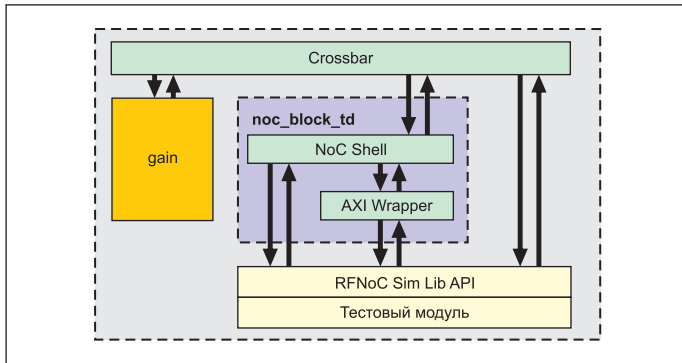


Рис. 6. Модуль цифрового программируемого усилителя

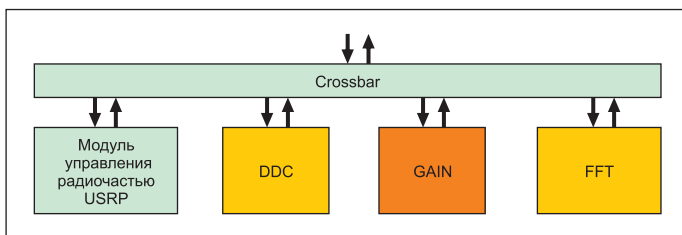


Рис. 7. Сборка для ПЛИС для рассмотренного проекта

6. Данные передаются по Crossbar на вход управляющего блока.
7. И наконец, данные через NOC Shell, AXI Wrapper и библиотеку RFNoC поступают обратно в тестовый модуль.

Последовательность действий модуля gain:

1. Точка 1 соответствует пункту 3 из последовательности действий тестового модуля.
2. Передача данных через AXI Wrapper в пользовательскую часть модуля.
3. Данные представляют собой компоненты I и Q, которые равны (1, 2, 3, ...), умножаются на коэффициент 100 и выходят обратно (100, 200, 300, ...), проходя через входной и выходной FIFO-буферы.
4. Готовые данные передаются через AXI Wrapper в NOC Shell модуля.
5. Данная точка соответствует пункту 5 из последовательности действий тестового модуля.

Таким образом, показано, что с помощью концепции RFNoC возможно формирование модулей и тестовой среды в автоматическом режиме, что дает существенные преимущества по сравнению с классическим подходом к разработке, при котором периферийные блоки обмена данными с шиной, а также тестовый модуль необходимо реализовать вручную.

### Формирование сборки для ПЛИС USRP X310 с применением пользовательских модулей

Здесь описаны результаты формирования сборки для ПЛИС для проекта, рассмотренного в предыдущем разделе, с применением пользовательского модуля, а также анализ результатов подключения и размещения схе-

мы на ПЛИС. Графически сборка показана на рис. 7. Можно видеть, что вместе с пользовательским блоком gain на шине располагаются и модули понижения частоты дискретизации DDC (Digital Down Converter), быстрого преобразования Фурье (БПФ), а также модуль интерфейса к радиотракту.

Сформировать сборку заданного вида можно посредством подготовленных скриптов, предварительно задав необходимые настройки, то есть указав, какие модули необходимо использовать и где они находятся. На рис. 8 представлены общие результаты использования ресурсов ПЛИС для заданной сборки.

Таким образом, основным преимуществом концепции RFNoC в контексте формирования сборки для ПЛИС является наличие подготовленного проекта для ПЛИС, используемого в USRP, учитывающего все особенности аппаратной части, в которую пользователю необходимо только встроить свой уникальный модуль.

### Создание и моделирование сложных перепрограммируемых пользовательских блоков

Как уже было сказано, с помощью автоматизированной среды RFNoC возможно сформировать верхний уровень пользовательского блока и тестовый модуль для верификации и валидации его функционала. Однако мы рассмотрели достаточно простой блок цифрового усилителя. Теперь обратимся к процессу разработки относительно сложного блока — перепрограммируемого фильтра с конечной симметричной импульсной характеристикой (КИХ-фильтр).

### Моделирование фильтра в среде MATLAB/Simulink

Первым шагом в создании блока КИХ-фильтра в рамках концепции модельно-ориентированного проектирования является разработка его имитационной модели в среде MATLAB/Simulink [2]. Это позволит сформировать облик блока, подать определенные воздействия и определить, соответствует ли работа модуля теории. На рис. 9 показана амплитудно-частотная характеристика (АЧХ) проектируемого фильтра для некоторых начальных коэффициентов. Также отражена разница в АЧХ при переходе к целочисленным знаковым 8-битным коэффициентам.

Один из базовых методов экспериментальной проверки функциональности КИХ-фильтра — снятие импульсной характеристики (ИХ). На рис. 10 показана модель для реализации этого. На данной схеме также можно видеть подачу значения shift, которое определяет битовый сдвиг для выходных отсчетов фильтра, позволяющий преобразовать разрядность и выдать результат фильтрации в 16-битном знаковом представлении. Для снятия импульсной характеристики, то есть в случае, когда на блок подается импульс единичной амплитуды, битовый сдвиг задается равным нулю. Поскольку фильтр предназначен для фильтрации комплексных сигналов, он имеет два входа и два выхода, но один набор коэффициентов и одно значение битового сдвига. На рис. 11 показана внутренняя структура фильтра.

В структуре фильтра на рис. 11 можно видеть, что входными значениями являют-

Аппаратные ресурсы ПЛИС	Использован	Доступно	Использован, %
LUT	114 958	254 200	45,22
LUTRAM	27 441	90 600	30,29
FF	136 877	508 400	26,92
BRAM	370	795	46,54
DSP	185	1540	12,01
IO	419	484	86,57
GT	2	16	12,5
BUFG	21	32	65,62
MMCM	4	10	40
PLL	3	10	30

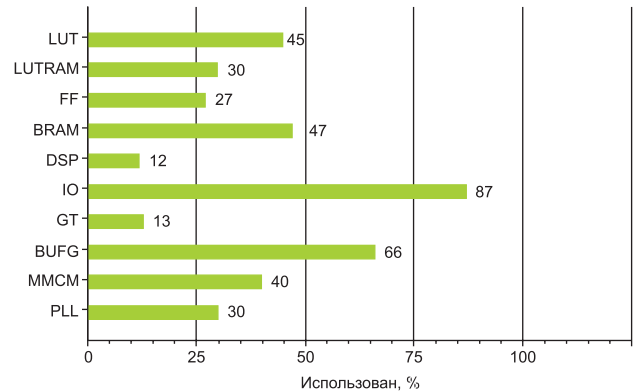


Рис. 8. Общие результаты использования ресурсов ПЛИС для заданной сборки

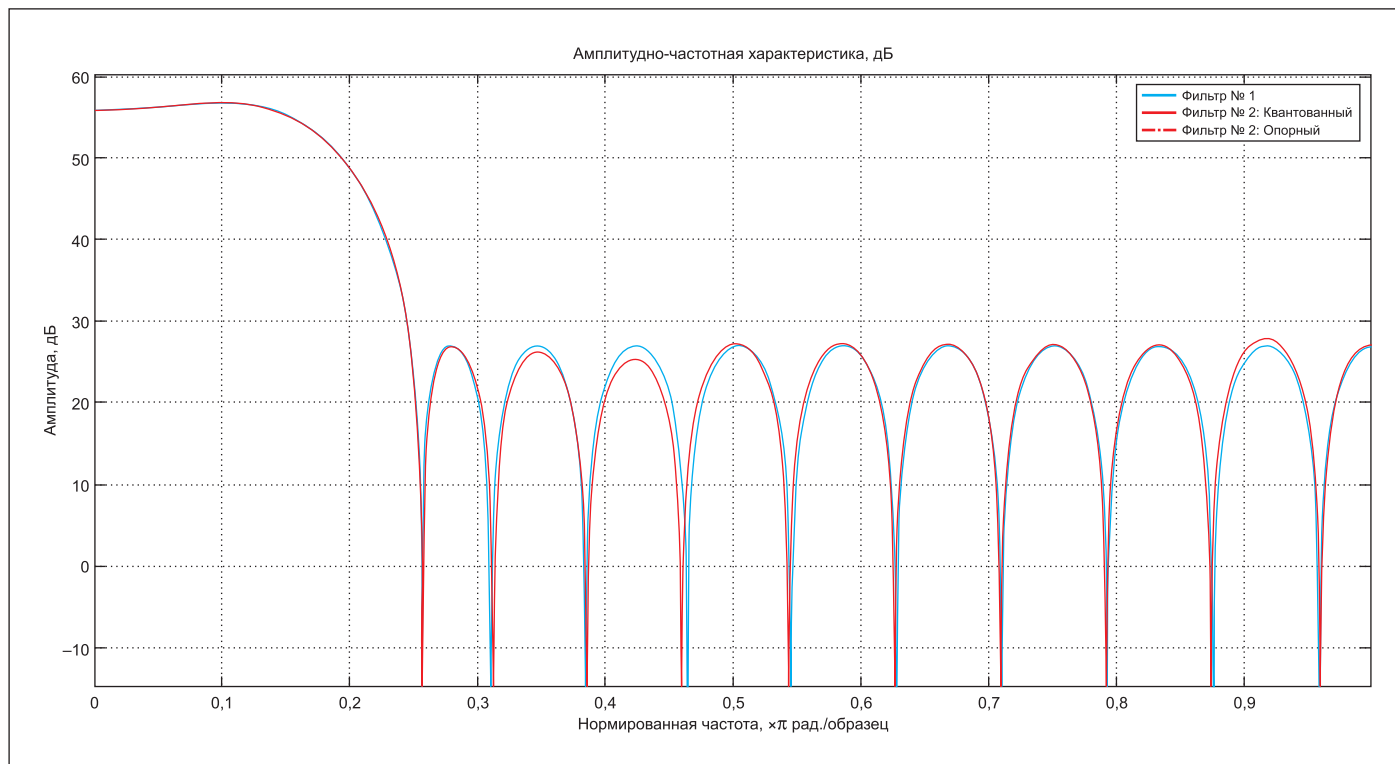


Рис. 9. Амплитудно-частотная характеристика проектируемого фильтра для некоторых начальных коэффициентов

ся знаковые 16-битные квадратуры сигнала. В фильтре реализованы две линии задержки и два набора умножителей для каждой квадратуры. При этом фильтр оптимизирован для случая симметричной импульсной характеристики и имеет меньшее количество умножителей, нежели обычная каноническая реализация. Единный вектор входных коэффициентов с помощью схем выбора подается на обе ветви фильтра. Также очевидно, что используется единое входное значение битового сдвига для формирования выходных 16-битных квадратур фильтрованного сигнала. Результат работы данной модели показан на рис. 12.

По результатам можно видеть, что экспериментально полученная ИХ фильтра повторяет коэффициенты фильтра, следовательно, схема реализована корректно.

### Моделирование фильтра в среде RFNoC и формирование сборки

Следующий этап — автоматическая генерация кода [2]. С помощью HDL Coder сформирован Verilog-код, объявление модуля в котором имеет следующий вид:

```
module fir_cplx_rld
    (clk, reset, clk_enable, data_inre_s16, data_inim_s16, coefs_s8_0,
    coefs_s8_1, coefs_s8_2, coefs_s8_3, coefs_s8_4, coefs_s8_5,
    coefs_s8_6, coefs_s8_7, coefs_s8_8, coefs_s8_9, coefs_s8_10,
    coefs_s8_11, coefs_s8_12, shift_u16, ce_out, data_outre_s16,
    data_outim_s16);
```

Видно, что блок содержит служебные сигналы, такие как `clk`, `reset`, и другие, ко-

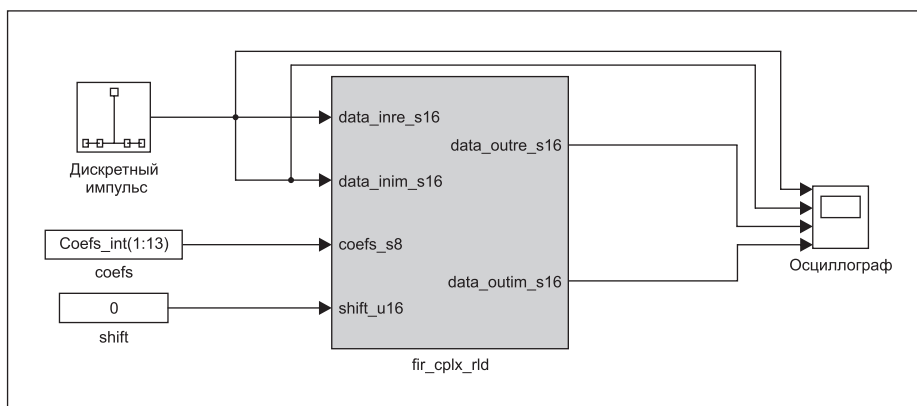


Рис. 10. Модель для снятия импульсной характеристики

торые необходимо сформировать для его корректной работы. Коэффициенты и битовый сдвиг являются входами сгенерированного фильтра, следовательно, их также необходимо подать на вход помимо значений сигнала. Далее алгоритм действий следующий:

1. Автоматически с помощью скриптов формируется облик блока по аналогии с модулем усилителя.
2. Создаются регистры для программируемых коэффициентов, сигналы с которых заводятся на вход блока в порты коэффициентов. То же самое необходимо проделать и для битового сдвига.
3. Поток входных и выходных отсчетов проходят через интерфейсы работы с AXI-шиной по аналогии с блоком цифрового усилителя.

Результаты моделирования HDL-кода созданного модуля в среде Xilinx Vivado показаны на рис. 13 (проверка записей и чтения регистров, а также программирование коэффициентов) и рис. 14 (экспериментальное снятие ИХ фильтра).

Можно видеть, что эпоры полностью совпадают с моделированием в MATLAB/Simulink за исключением задержки между входом и выходом: в модели Simulink задержки нет, в случае Vivado — два такта. Такая разница обусловлена наличием дополнительных входных и выходных регистров в HDL-реализации фильтра, добавляющих по одному такту задержки каждый.

По аналогии с цифровым усилителем далее возможно сформировать сборку для USRP с включенным в нее блоком КИХ-фильтра. Результаты формирования показаны

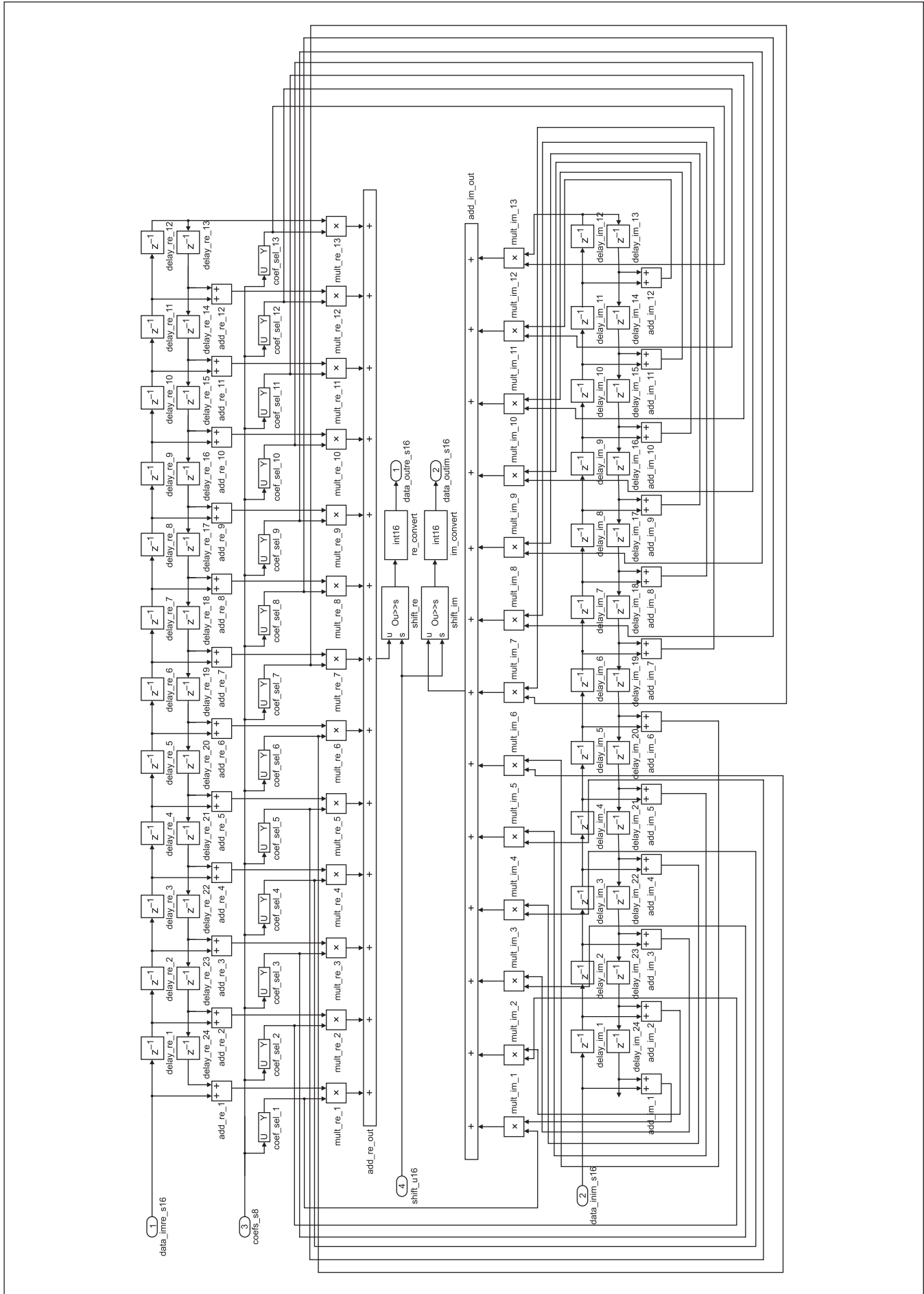


Рис. 11. Внутренняя структура КИХ-фильтра

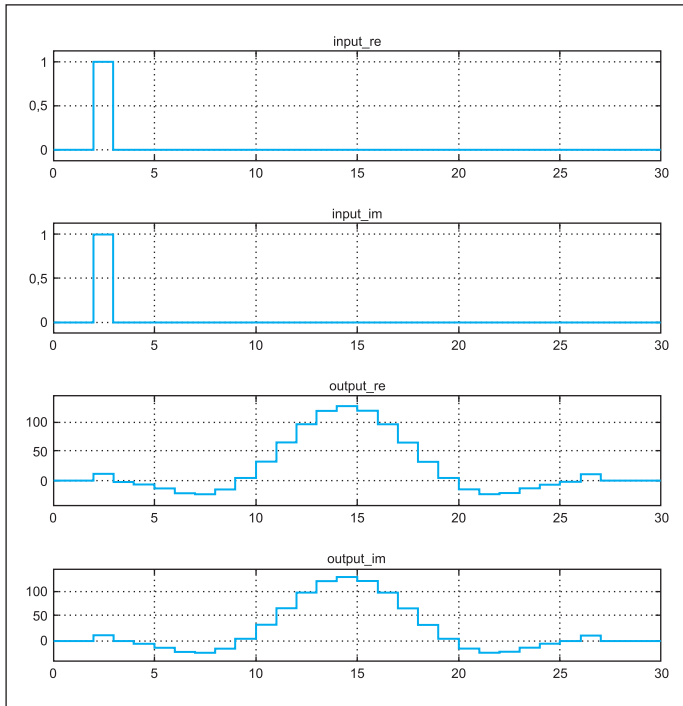


Рис. 12. Результат работы описанной модели

на рис. 15, 16. Следует отметить, что исходная сборка для USRP, входящая в комплект поставки, содержит блоки аппаратного БПФ и деци-

мации/интерполяции, которые в текущем проекте являются необязательными элементами, и их можно удалить, сэкономив ресурсы.

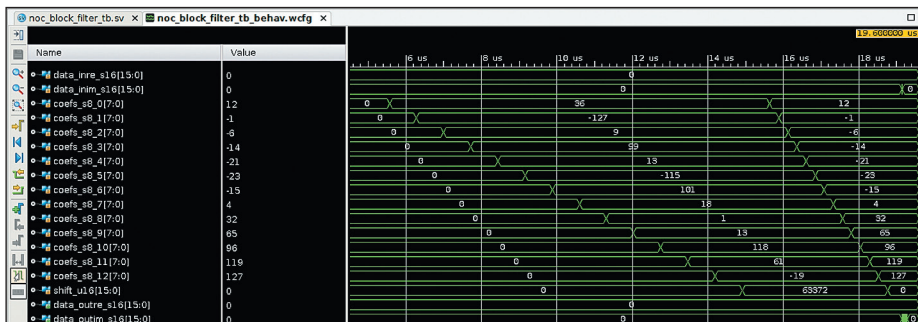


Рис. 13. Проверка записей и чтения регистров, а также программирование коэффициентов

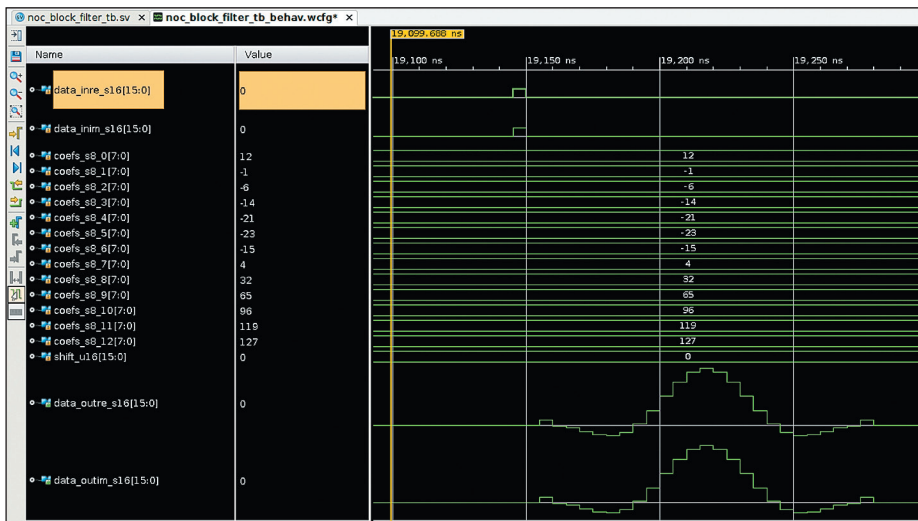


Рис. 14. Экспериментальное снятие ИХ фильтра

Аппаратные ресурсы ПЛИС	Использован	Доступно	Использован, %
LUT	87 929	254 200	34,59
LUTRAM	17 640	90 600	19,47
FF	101 809	508 400	20,03
BRAM	323	795	40,63
DSP	109	1540	7,08
IO	419	484	86,57
GT	2	16	12,5
BUFG	21	32	65,62
MCM	4	10	40
PLL	3	10	30

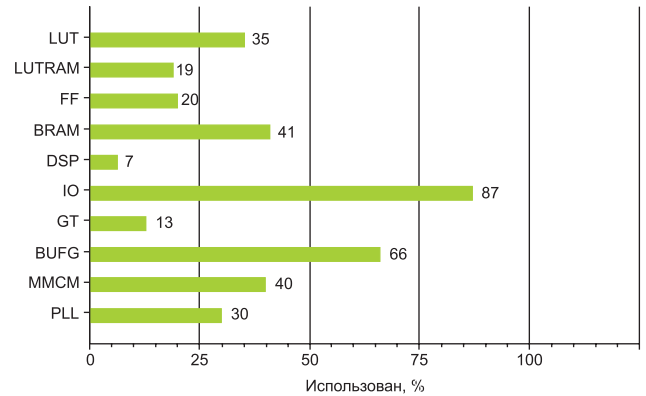


Рис. 16. Результаты использования ресурсов ПЛИС для заданной сборки с включенным блоком КИХ-фильтра

## Заключение

Основные выводы, которые можно сделать на основе выполненной работы, следующие:

1. Концепция и инструментальные средства RfNoC обеспечивают эффективную реализацию и тестирование модулей цифровой обработки сигналов на ПЛИС.
2. На основе рассмотренного примера формирования сборки для USRP X310, включающей пользовательский модуль, показано, что служебная часть сборки ПЛИС не превышает 30% ресурсов ПЛИС, что дает достаточный запас для реализации довольно сложных модулей обработки сигналов.

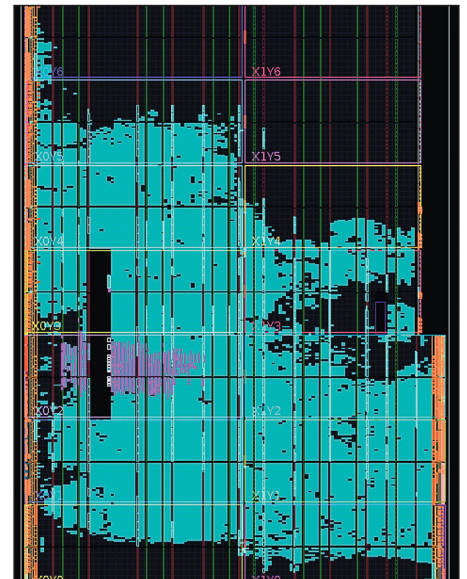


Рис. 15. Результат формирования сборки для USRP с блоком КИХ-фильтра

3. Пример реализации КИХ-фильтра с программируемыми коэффициентами показал, что применение методики проектирования пользовательских модулей для ПЛИС с помощью сразу нескольких подходов к проектированию, а именно модельно-ориентированного проектирования, автоматизированной генерации кода, технологии программно-определяемого радио, а также набора скриптов Bash, Tcl и Python, позволяет существенно сократить время разработки за счет упрощения процедуры формирования обвязки и верхнего уровня модуля, а также среды тестирования. ■

### Литература

1. Ettus Research, The National Instruments Company. [www.ettus.com](http://www.ettus.com)
2. Kintali K., Gu Y. Model-Based Design with Simulink, HDL Coder, and Xilinx System Generator for DSP. MathWorks Inc. White Paper, 2012.
3. Braun M., Pendlum J. RFNoC: RF Network on Chip. Ettus Research, The National Instruments Company, 28.05.2014.
4. Pendlum J. RFNoC Deep Dive: FPGA Side. Ettus Research, The National Instruments Company, 28.05.2014.