

УДК: 621.396.96

DOI: 10.30987/2658-6436-2021-3-4-39-48

А.В. Смоляков, А.С. Подстригаев

## ПРОГРАММНО-ОПРЕДЕЛЯЕМЫЙ ИМИТАТОР РАДИОЛОКАЦИОННЫХ ЦЕЛЕЙ НА ОСНОВЕ ТЕХНОЛОГИИ DRFM ДЛЯ АВТОМАТИЗИРОВАННОЙ ПРОВЕРКИ РЛС

*В статье описан программно-определяемый имитатор радиолокационных целей на основе технологии DRFM. Такое устройство может быть использовано для проверки разрабатываемых РЛС и оценки устойчивости их работы в условиях воздействия радиопомех. В работе подробно описаны общий алгоритм работы имитатора и входящие в него алгоритмические модули имитации дальности и скорости. Модуль сдвига частоты для имитатора реализован в Xilinx System Generator и на языке VHDL. Выполнено сравнение объема ресурсов ПЛИС, требуемого для этих реализаций.*

**Ключевые слова:** DRFM, ПЛИС, имитатор целей с DRFM, имитатор целей РЛС, суперсэмплинг, покадровая обработка, SDR, HLS.

A.V. Smolyakov, A.S. Podstrigaev

## SOFTWARE-DEFINED RADAR TARGET SIMULATOR WITH DRFM FOR AUTOMATED RADARS TESTING

*The paper describes a software-defined radar target simulator with DRFM. One may use such a device to test radars under development and evaluate their reliability in environments with radio frequency interference. The paper describes in detail a general algorithm of the target simulation and its modules for the simulation of range and speed. The authors also implemented the frequency shift module for the simulator in Xilinx System Generator and wrote this module in VHDL. One may find in the paper a comparison of the FPGA resources required for such implementations.*

**Keywords:** DRFM, FPGA, DRFM target simulator, radar target simulator, supersampling, frame-based signal processing, SDR, HLS.

### Введение

В настоящее время радиолокационные системы (РЛС) широко используются для решения задач противовоздушной обороны, управления воздушным движением, морской навигации, повышения автономности транспортных средств и т.д. Проверка – один из важнейших этапов разработки РЛС. На этом этапе разработчики должны проверить работу РЛС в различных режимах и условиях, выявить проблемы и недостатки тестируемой РЛС, которые необходимо исправить в дальнейшем, а также оценить устойчивость работы РЛС в условиях воздействия электромагнитных помех. Очевидно, что такая проверка является весьма трудоемкой и требует высокой степени автоматизации.

Автоматизация проверки РЛС возможна при использовании программно-определяемых имитаторов радиолокационных целей (РЛЦ). Такие имитаторы могут быть аналоговыми [1] и цифровыми [2, 3]. Наиболее распространенный способ реализации цифрового имитатора основан на использовании технологии DRFM (англ. Digital Radio Frequency Memory – цифровая радиочастотная память). При этом входной сигнал обычно преобразуется на промежуточную частоту, где он оцифровывается аналого-цифровым преобразователем (АЦП), а затем сохраняется в цифровой памяти. После этого сохраненные отсчеты сигнала обрабатываются для формирования РЛЦ для тестируемой РЛС и преобразовываются обратно в аналоговое представление с помощью цифро-аналогового преобразователя (ЦАП).

Полученный сигнал возвращается на исходную несущую частоту. Обработка отсчетов сигнала, хранящихся в памяти, может выполняться разными способами, например, с использованием программируемых логических интегральных схем (ПЛИС) [4–6] или цифровых сигнальных процессоров и ПЛИС [7, 8]. Известны также решения DRFM на основе универсального измерительного оборудования [9]. В данной статье рассматривается имитатор РЛЦ на основе DRFM, реализованный на ПЛИС (рис. 1).

Известно, что из-за задержки обработки имитатор РЛЦ не может имитировать цели, расположенные к РЛС ближе, чем сам имитатор. Для формирования таких целей ему потребовалось бы излучить имитационный сигнал до приема зондирующего импульса РЛС. Однако это требует синхронизации имитатора с РЛС, что делает его менее универсальным и затрудняет использование в полевых условиях. Следовательно, задержка излучения первого импульса является важным параметром для имитатора РЛЦ. Несмотря на это, в большинстве статей по данной теме [4–9] не рассматриваются конкретные алгоритмы обнаружения сигналов и реализации конвейера подготовки радиолокационных целей. Таким образом, они не охватывают проблему минимизации задержки в системах на основе DRFM.

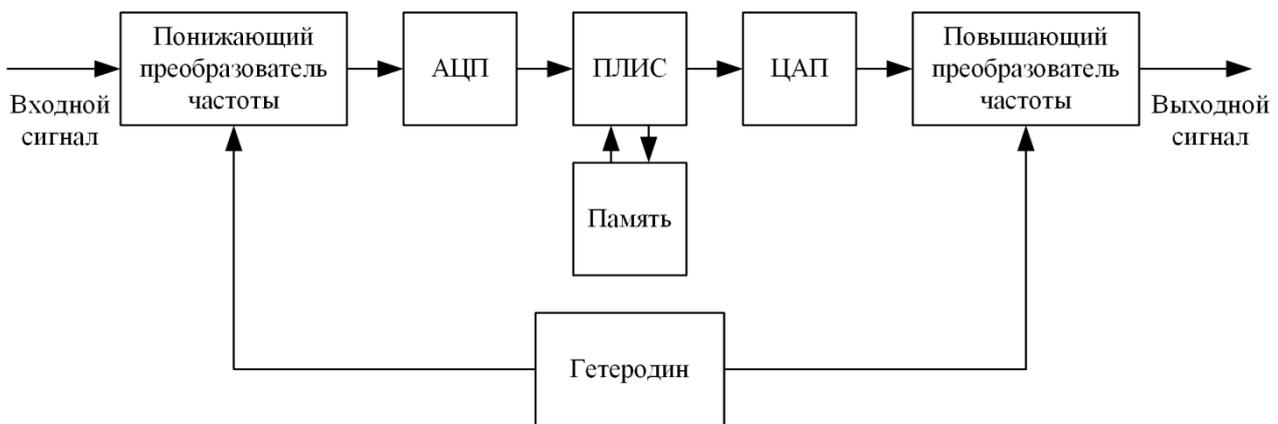


Рис. 1. Общая структура имитатора радиолокационных целей на основе DRFM

Поэтому целью настоящей статьи является разработка алгоритма управления имитатором РЛЦ на основе DRFM, а также проработка практической реализации входящих в него алгоритмических модулей обнаружения, внесения задержки и частотного сдвига в переизлучаемый сигнал.

### Аппаратная реализация имитатора

#### *Преобразователи частоты*

Для переноса принятого высокочастотного сигнала на промежуточную частоту и обратно используются соответственно понижающий и повышающий преобразователи частоты. Особенности их реализации выходят за рамки данной работы, однако здесь стоит упомянуть один аспект. Для достижения описанного поведения преобразователей гетеродин не должен изменять свою частоту во время обработки сигнала в ПЛИС. Если частота гетеродина фиксирована, а имитатор покрывает только одну конкретную полосу частот, выбранную на этапе разработки, это условие выполняется автоматически. Однако, если имитатор выполняет сканирование частотного диапазона, перестраивая гетеродин, то гетеродин и ПЛИС должны быть синхронизированы. Самый простой способ добиться такой синхронизации – контролировать частоту гетеродина с помощью ПЛИС.

*Аналого-цифровой преобразователь (АЦП) и цифро-аналоговый преобразователь (ЦАП)*

АЦП и ЦАП следует выбирать, основываясь на требованиях к ширине мгновенной рабочей полосы пропускания имитатора. Кроме того, чтобы избежать нежелательного искажения сигнала, разрешение ЦАП должно быть не меньше разрешения АЦП. На практике разрешение ЦАП должно быть выше разрешения АЦП потому, что обработка сигнала обычно увеличивает количество бит, необходимых для его представления. В результате более высокое разрешение ЦАП делает РЛЦ более правдоподобными для тестируемой РЛС.

#### *Память*

Поскольку имитатор обычно работает с широкой мгновенной полосой пропускания, то на практике объем внутренней памяти ПЛИС, как правило, является недостаточным для хранения данных. Следовательно, необходимо использовать внешнюю высокоскоростную двупортовую память. Наиболее подходящей памятью для этой задачи является синхронная динамическая оперативная память с удвоенной скоростью передачи данных (англ. – Double Data Rate Synchronous Dynamic Random-Access Memory, DDR SDRAM).

#### *ПЛИС*

ПЛИС является основным компонентом рассматриваемого имитатора целей на основе DRFM. Во-первых, она выполняет все необходимые операции по обработке сигналов. Во-вторых, ПЛИС управляет периферийными устройствами на печатной плате имитатора РЛЦ. Это может быть, например, вышеупомянутый гетеродин или выходной усилитель мощности, который иногда можно отключать для экономии энергии. Также на ПЛИС могут быть реализованы интерфейсы информационного обмена с устройствами более высокого уровня (например, интерфейс Ethernet для приема пользовательских команд, отправляемых с ПК).

Обычно имитатор должен выполнять свои задачи в широкой мгновенной полосе пропускания, которая на практике достигает 1...2 ГГц. Это требование означает, что частота дискретизации АЦП обычно в несколько раз превышает тактовую частоту ПЛИС. Таким образом, ПЛИС имитатора должна позволять разработчику реализовывать конвейерные алгоритмы с высокой пропускной способностью с обработкой нескольких отсчетов за каждый такт. Такая обработка обычно называется суперсэмплингом или покадровой обработкой сигнала. При этом ПЛИС должна работать на максимально возможной тактовой частоте, чтобы минимизировать задержку формирования цели. Кроме того, ПЛИС должна обеспечивать как можно больше арифметических логических блоков, предназначенных для операций цифровой обработки сигналов (ЦОС), таких как блоки DSP48 Xilinx или блоки DSP Intel.

Несмотря на широкие возможности обработки сигналов, ПЛИС – не лучший выбор для управления периферийными устройствами и связи с устройствами более высокого уровня. Есть два способа преодолеть этот недостаток. Первый – использовать программный процессор внутри ПЛИС, оптимизированный специально для задач цифровой обработки. Например, процессор MicroBlaze в ПЛИС серии Virtex от Xilinx или процессор Nios II в ПЛИС Intel серии Arria. Второй способ – использовать систему на кристалле (SoC), такую как Xilinx Zynq или ПЛИС семейства Intel Stratix. Второй вариант является предпочтительным в тех случаях, когда имитатор должен работать со сложными высокоскоростными интерфейсами, такими как SATA PCI-E.

### **Реализация программного обеспечения имитатора**

В первом подразделе этого раздела предлагается общая программная архитектура имитатора РЛЦ. В следующих подразделах подробно рассматриваются возможные алгоритмы обнаружения сигналов, выбора задержки и выбора сдвига частоты.

#### *Общая архитектура*

Программное обеспечение (ПО) имитатора РЛС разработано на основе функциональной схемы, представленной на рис. 2.

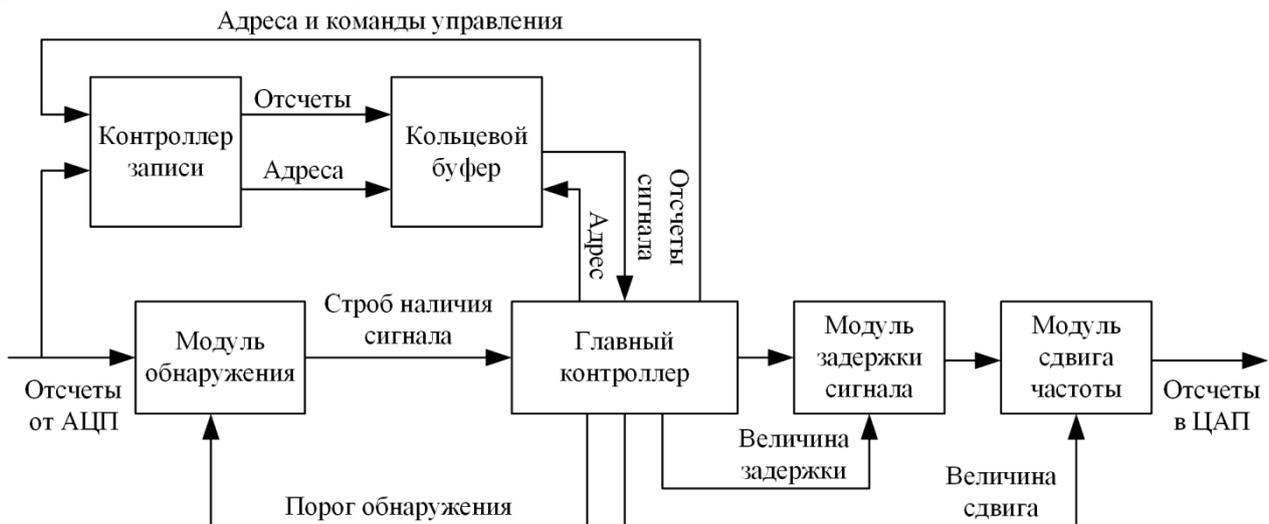


Рис. 2. Функциональная схема программного обеспечения предлагаемого имитатора

ПО работает следующим образом. Когда имитатор находится в режиме ожидания, отсчеты из АЦП через контроллер записи непрерывно записываются в кольцевой буфер, выделенный во внешней памяти. Главный контроллер непрерывно отправляет в контроллер записи адреса, по которым эти отсчеты должны быть сохранены.

Одновременно те же отсчеты поступают в модуль обнаружения, формирующий строб наличия сигнала. Как только модуль обнаруживает во входных отсчетах сигнал РЛС, он устанавливает на своем выходе состояние высокого логического уровня. Это состояние сохраняется до тех пор, пока модуль обнаружения не перестанет обнаруживать сигнал во входных отсчетах.

По фронту стога наличия сигнала главный контроллер начинает считывать отсчеты сигнала из кольцевого буфера. Зная задержку модуля обнаружения, главный контроллер может вычислить адреса требуемых отсчетов в кольцевом буфере. В этом же тактовом периоде главный контроллер переключает имитатор в рабочий режим и начинает посылать отсчеты сигнала в модуль сдвига частоты. Как только главный контроллер обнаруживает задний фронт стога наличия сигнала, он отправляет контроллеру записи команду остановить запись отсчетов в буфер и начинает циклически отправлять записанные отсчеты обнаруженного сигнала в модуль сдвига частоты, задерживая их перед каждой отправкой. Таким образом, имитатор формирует несколько РЛС, находящихся на разном удалении от РЛС. Как только контроллер сформировал необходимое количество целей, он переключает имитатор обратно в режим ожидания.

Требуемые задержки и частотные сдвиги или алгоритмы их расчета могут быть заранее определены разработчиками в ПО имитатора или периодически отправляться на главный контроллер каким-либо устройством более высокого уровня. В то же время необходимости информировать имитатор о характеристиках сигнала РЛС нет. Поскольку помеховый сигнал имитатора является набором копий импульса РЛС, нет также и необходимости в наличии информации о параметрах модуляции зондирующего сигнала РЛС. Единственные характеристики, незнание которых может создать проблему для имитатора – это длительность импульса и период повторения импульсов (ППИ).

Однако, благодаря универсальности предлагаемой структуры программного обеспечения, основанной на кольцевом буфере, имитатор адаптируется к длительности импульсов РЛС, даже если длительность импульсов является переменной. Имитатор может сохранять и ретранслировать сигналы, длительность которых может варьироваться от одного

кадра данных АЦП до размера кольцевого буфера. Если имитатор принимает сигнал, длительность которого превышает требуемую задержку, во избежание искажения сигнала, вызванного его обрезкой, он может автоматически увеличить значение задержки, например, чтобы оно соответствовало длительности входного импульса.

Точно так же имитатор может адаптироваться к ППИ РЛС: система более высокого уровня может дать команду главному контроллеру либо игнорировать новые стробы от модуля обнаружения, пока имитатор находится в рабочем режиме, либо немедленно начать обработку следующего импульса РЛС, прерывая текущую передачу. Последний способ оптимален, если есть уверенность, что в зоне действия имитатора находится только одна РЛС. В этом случае имитатор будет выдавать одинаковое количество РЛЦ в ответ на каждый зондирующий импульс РЛС.

Главный контроллер может регулировать величины задержки и сдвига частоты между ретрансляциями с малой дискретностью, чтобы имитировать движение целей более правдоподобно.

Имитатор также может формировать одну РЛЦ для РЛС, использующей непрерывный зондирующий сигнал. В этом случае, если имитатор обнаруживает переполнение кольцевого буфера до того, как он получил задний фронт строга наличия сигнала, он продолжает передавать первую копию сигнала, не останавливая перезапись буфера.

#### *Модуль обнаружения*

Как описано в предыдущем подразделе, чтобы оставаться универсальным по отношению к характеристикам сигнала РЛС, имитатор не должен использовать какую-либо информацию об этом сигнале. Следовательно, единственный вариант алгоритма обнаружения – это энергетическое обнаружение. Существует множество вариантов этого алгоритма, например, на основе преобразования Фурье [10]. Однако также имеются строгие требования к задержке обнаружения. Поэтому для представленного имитатора целей выбран один из простейших алгоритмов, описанных в [11]. Согласно этому алгоритму, решение о наличии сигнала принимается, если решающая статистика  $Z = \sum_{i=0}^{N-1} s_i^2$  превосходит заранее заданный порог  $Z_{th}$  (где  $N$  – число анализируемых отсчетов сигнала, а  $s_i$  – сами анализируемые отсчеты).

Разумно выбирать  $N$  кратным числу отсчетов в кадре данных АЦП. Значение порога  $Z_{th}$  можно вычислить как

$$Z_{th} = \sigma^2 h, \quad (1)$$

где  $\sigma^2$  – дисперсия шума в приемном канале, а  $h$  – нормированный порог, вычисляемый как

$$h = 2 * Q^{-1}(P_{fa}, N/2), \quad (2)$$

где  $P_{fa}$  – требуемая вероятность ложной тревоги модуля обнаружения,  $Q^{-1}(y, a)$  – функция, обратная к верхней неполной Гамма-функции, определенная так, что  $x = Q^{-1}(y, a)$ , где  $y = Q(x, a) = (1/\Gamma(a)) \int_x^\infty t^{a-1} e^{-t} dt$ , а  $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$  – Гамма-функция. Величина функции  $Q^{-1}(y, a)$  может быть получена из функции *gammaincv* в MATLAB.

Как видно из (1) и (2), рассматривая имитатор как систему с CFAR (англ. Constant False Alarm Rate – постоянная вероятность ложных тревог), значение нормированного порога  $h$  может быть рассчитано на этапе проектирования имитатора и зафиксировано в его ПО.

Также можно выбрать порог, исходя из текущего значения дисперсии шума в канале. Однако, для этого требуется периодическая калибровка имитатора. При этом система более высокого уровня переключает вход имитатора на согласованную нагрузку, чтобы имитатор мог оценить дисперсию собственного шума. Соответственно, в процессе калибровки

имитатор не функционирует. Частота калибровки зависит от изменения сигнально-помеховой обстановки и в лабораторных условиях может быть невысокой.

*Модуль задержки сигнала*

Чтобы имитировать необходимое расстояние между несколькими РЛЦ, главный контроллер задерживает отправку каждой копии отсчетов сигнала из памяти в модуль сдвига частоты. Поскольку в имитаторе есть только один канал передачи, он не может передать еще одну копию записанного сигнала, пока не закончит передачу предыдущей. Следовательно, фактическая задержка между сигналами РЛЦ не может быть меньше длительности импульса зондирующего сигнала РЛС. При этом верхнего предела для значения задержки практически нет. Однако, если имитатору дана команда немедленно обработать следующий импульс с прерыванием текущей передачи, как только он обнаружит на своем входе новый импульс, максимальное значение задержки ограничено временем между зондирующими импульсами РЛС.

*Модуль сдвига частоты*

Используемый в предлагаемом имитаторе метод сдвига частоты принимаемого сигнала  $s(t)$  основан на умножении аналитического представления сигнала на комплексный гармонический сигнал сдвига  $\exp[j\Delta\varphi]$  [12]:

$$s'(t) = \operatorname{Re}\{A(t)\exp[j\Phi(t)] \cdot \exp[j\Delta\varphi]\} = \operatorname{Re}\{A(t)\exp[j(\Phi(t) + \Delta\varphi)]\} = \\ = A(t)\cos[\Phi(t) + \Delta\varphi],$$

где  $A(t) = \sqrt{s^2(t) + s_{\perp}^2(t)}$ , такая, что  $s_{\perp}(t)$  – трансформанта Гильберта  $s(t)$ ,  $\Phi(t)$  – такая функция, что  $\tan(\Phi(t)) = s_{\perp}(t)/s(t)$  и  $\Delta\varphi$  – полная фаза сигнала сдвига, определенная как  $\Delta\varphi = \Delta\omega t + \varphi_0$ , где  $\Delta\omega$  – желаемый сдвиг частоты, а  $\varphi_0$  – начальная фаза сигнала сдвига. Так как частотная и фазовая модуляция связаны и могут быть преобразованы друг в друга,  $\Phi(t)$  можно переписать как  $\Phi(t) = \omega_0 t + \varphi(t)$ , где  $\omega_0$  – несущая частота входного сигнала, а  $\varphi(t)$  – закон фазовой модуляции этого сигнала. Выходной сигнал модуля сдвига частоты, таким образом, можно записать в виде

$$s'(t) = A(t)\cos[(\omega_0 + \Delta\omega)t + \varphi(t) + \varphi_0].$$

Таким образом,  $s'(t)$  отличается от  $s(t)$  только наличием сдвигов несущей частоты  $\Delta\omega$  и начальной фазы  $\varphi_0$ . Первый сдвиг является необходимым. Последний, в то же время, может вызвать проблемы, если имитатор работает с РЛС, которая использует когерентное накопление сигнала. Тем не менее, это можно считать приемлемой ценой за универсальность имитатора и отказ от добавления дополнительных радиочастотных компонентов на его печатную плату для имитации доплеровского сдвига частоты.

Те же операции можно переписать и в алгебраической форме:

$$s'(t) = \operatorname{Re}\{[s(t) + js_{\perp}(t)][\cos(\Delta\varphi) + j\sin(\Delta\varphi)]\} = \\ = \operatorname{Re}\{[s(t)\cos(\Delta\varphi) - s_{\perp}(t)\sin(\Delta\varphi)]\} + \\ + \operatorname{Re}\{j[s(t)\sin(\Delta\varphi) + s_{\perp}(t)\cos(\Delta\varphi)]\}$$

Таким образом,

$$s'(t) = s(t)\cos(\Delta\varphi) - s_{\perp}(t)\sin(\Delta\varphi) \tag{3}$$

Соответственно, чтобы выполнить данные операции над отсчетами  $s(t)$ , ПЛИС должна:

1. Вычислить преобразование Гильберта от входного сигнала. Это можно сделать, например, с помощью цифрового фильтра с конечной импульсной характеристикой (КИХ).

2. Сгенерировать отсчеты функций синуса и косинуса, имеющие частоту, равную желаемому значению сдвига, например, с помощью DDS-генераторов (англ. Direct Digital Synthesis – прямой цифровой синтез).

3. Выполнить умножение и вычитание отсчетов в соответствии с (3).

Здесь также стоит упомянуть несколько дополнительных деталей реализации. Поскольку ПЛИС выполняет обработку входного сигнала покaдрово, все вышеописанные операции должны выполняться для нескольких отсчетов сигнала на каждом такте. Дизайн таких КИХ-фильтров описан в [13]. Для генерации  $N$  комплексных отсчетов на каждом такте можно использовать  $N$  генераторов DDS, работающих синхронно с фазовыми сдвигами  $i(360^\circ/N)$ , где  $i = 0, (N-1)$  – номер генератора. Наконец, для выполнения необходимых умножений целесообразно использовать блоки DSP ПЛИС.

#### *Программное конфигурирование имитатора*

Важно отметить, что предлагаемая архитектура ПО имитатора делает его программно-определяемым. Так, например, ПО ПЛИС может содержать несколько различных вариантов модуля обнаружения, оптимальных для работы с сигналами различных типов. Современные ПЛИС имеют возможность частичной динамической реконфигурации [14]. При такой реконфигурации функциональность отдельных блоков микросхемы можно изменять, не затрагивая другие блоки и не повторяя длительной процедуры инициализации ПО. Благодаря этому алгоритм обнаружения можно быстро адаптировать под меняющуюся сигнально-помеховую обстановку, практически не прерывая работу имитатора. Кроме того, неактивные модули обнаружения хранятся в постоянной памяти и не используют логические ресурсы ПЛИС.

Благодаря возможности реализовать в ПЛИС программный микропроцессор упрощается изменение отдельных параметров встроенного ПО из ПО верхнего уровня. ПО для такого процессора разрабатывать значительно проще, чем ПО, реализуемое непосредственно в логических ресурсах ПЛИС. Это позволяет сделать управление имитатором более гибким и предоставить ПО верхнего уровня возможность изменения практически любых параметров имитатора без остановки его работы. Так, например, ПО верхнего уровня через главный контроллер может управлять алгоритмом генерации задержек и частотных сдвигов.

Программное определение алгоритмов имитации РЛЦ, таким образом, значительно упрощает адаптацию имитатора к различным типам задач, возникающих при автоматизированной проверке РЛС. Кроме того, оно упрощает и применение имитатора в полевых условиях.

### **Реализация модуля сдвига частоты**

Модуль частотного сдвига – одна из самых сложных и требовательных к ресурсам частей ПО имитатора. Поэтому было решено реализовать этот модуль двумя способами – автоматизировано с использованием Xilinx System Generator и в виде модуля, написанного на языке VHDL. В следующих подразделах сравнивается количество ресурсов, требуемое для данных реализаций.

#### *Реализация в System Generator*

В качестве инструментов HLS (англ. High Level Synthesis – высокоуровневый синтез) для первой реализации модуля сдвига частоты использовались Xilinx System Generator 2018.2 и MATLAB R2017b. Модель модуля в Simulink представлена на **рис. 3**. Она реализована в соответствии с (3) и поэтому не требует подробного описания. Модуль разработан для обработки восьми отсчетов в одном входном кадре за каждый такт. Преобразование

Гильберта входного сигнала выполняется с помощью КИХ-фильтра с покадровой обработкой данных, синтезированного в MATLAB HDL Coder.

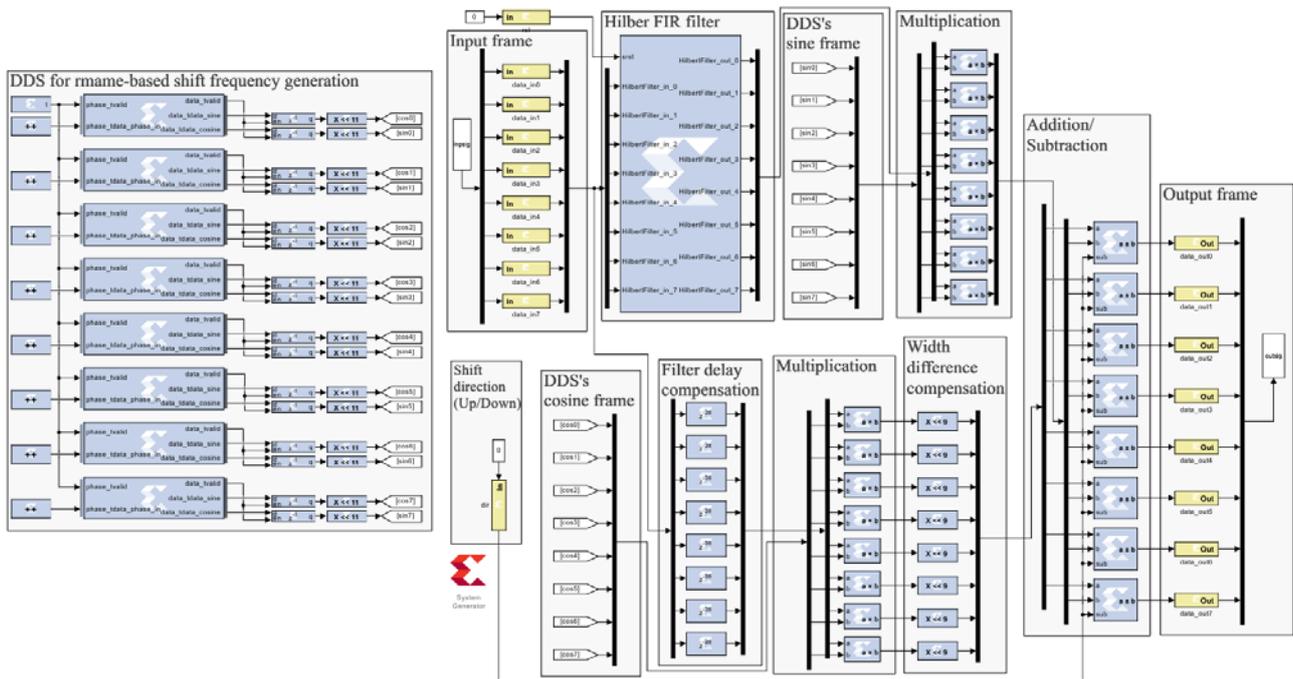


Рис. 3. Модуль сдвига частоты, реализованный в Xilinx System Generator

*Реализация на языке VHDL*

Этот же модуль реализован вручную на языке VHDL. Нет никаких структурных отличий между этой реализацией и реализацией в System Generator, о которых стоит упомянуть. Для реализации DDS-генераторов и умножителей используются IP-ядра Vivado. Для выполнения преобразования Гильберта входного сигнала используется тот же КИХ-фильтр, что и в первом варианте реализации.

*Сравнение ресурсоемкости*

Для проведения описанного сравнения с использованием Xilinx Vivado 2018.2 была выполнена имплементация обоих модулей в микросхеме Xilinx XC7K325TFFG900-2, используемой в отладочной плате Xilinx KC705. Оба модуля были проверены на этой плате с использованием IP ядра Internal Logic Analyzer и показали корректную работу на тактовых частотах до 200 МГц. Таким образом, они поддерживают частоту дискретизации до  $200 \times 8 = 1600$  МГц.

Результаты сравнения ресурсоемкости приведены в табл. 1.

Таблица 1. Результаты сравнения ресурсоемкости

Тип ресурса	Реализация	
	HLS в System Generator	Вручную написанный VHDL код
Logic Slices	7693	9003
LUTs	40098	47772
Block RAM Tiles	4	4
DSP48	52	36

Из таблицы видно, что System Generator сгенерировал более эффективный код, чем написанный вручную. Так, он использует меньше блоков LUT и размещает больше функций в блоки DSP. Следовательно, потенциально эта реализация может работать на более высокой тактовой частоте. Более того, из-за общей простоты Simulink и гораздо более удобных инструментов отладки и проверки, предоставляемых этим программным обеспечением, разработка модели модуля сдвига частоты в Simulink занимает значительно меньше времени, чем ее написание вручную на VHDL.

### Заключение

Предлагаемый программно-определяемый имитатор может быть использован для тестирования разрабатываемых РЛС и оценки способности таких РЛС работать в условиях сложной сигнально-помеховой обстановки.

Кроме того, было проведено сравнение объемов ресурсов, необходимых для реализации модуля сдвига частоты в ПЛИС. Сравнение показывает, что HLS с использованием Xilinx System Generator производит код с достаточным качеством для использования в высокопроизводительных системах цифровой обработки сигналов.

#### Список литературы:

1. **Подстригаев, А. С.** Исследование спектра сигнала при двукратном частотном преобразовании / А. С. Подстригаев, А. В. Смоляков // Труды МАИ. – 2020. – № 114. – 25 с. – DOI: 10.34759/trd-2020-114-11
2. **Купряшкин, И. Ф.** Космическая радиолокационная съемка земной поверхности в условиях помех / И. Ф. Купряшкин, В. П. Лихачев. – Воронеж: Научная книга, 2014. – 460 с.
3. **Li, H.** DRFM system based on the principle of radar deception/ H. Li, F. Zhao, F. Chen, J. Wang // International Journal of Simulation Systems, Science & Technology. – 2016. – № 17(37). – P.17.1–17.5. – DOI 10.5013/IJSSST.a.17.37.17
4. **Aseeri, M. A. S.** Design of DRFM system based on FPGA with high resources / M. A. S. Aseeri, A. A. Alasows, M. R. Ahmad // 2016 11th International Design & Test Symposium (IDT). – Hammamet, 2016. – P. 177-180. – doi: 10.1109/IDT.2016.7843036
5. **Olivier, K.** Modern wideband DRFM architecture and real-time DSP capabilities for radar test and evaluation / K. Olivier, M. Gouws // 2013 Saudi International Electronics, Communications and Photonics Conference. – Fira, 2013. – PP. 1-4. – doi: 10.1109/SIECPC.2013.6551019
6. **Strydom, J. J.** Hardware in the loop radar environment simulation on wideband DRFM platforms / J. J. Strydom, J. E. Cilliers, M. Gouws [et al.] // IET International Conference on Radar Systems (Radar 2012). – Glasgow, UK, 2012. – PP. 1-5. – doi: 10.1049/cp.2012.1687
7. **Li, C. -z.** Improved Interrupted Sampling Repeater Jamming based on DRFM / C. -z. Li, W. -m. Su, H. Gu [et al.] // 2014 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). – Guilin, 2014. – PP. 254-257. – doi: 10.1109/ICSPCC.2014.6986193
8. **Xie, M.** Design and realization of DRFM system

#### References:

1. Podstrigaev A.S. Signal spectrum studying at double frequency conversion / A. S. Podstrigaev, A. V. Smolyakov // Trudy MAI. – 2020. – № 114. – 25 P. – DOI: 10.34759/trd-2020-114-11
2. Kupryashkin I. F. Spaceborne radar imagery of the Earth's surface under jamming conditions / I. F. Kupryashkin, V. P. Likhachev. – Voronezh: Nauchnaya kniga, 2014. – 460 c.
3. Li H. DRFM system based on the principle of radar deception/ H. Li, F. Zhao, F. Chen, J. Wang // International Journal of Simulation Systems, Science & Technology. – 2016. – № 17(37). – P.17.1–17.5. – DOI 10.5013/IJSSST.a.17.37.17
4. Aseeri M. A. S. Design of DRFM system based on FPGA with high resources / M. A. S. Aseeri, A. A. Alasows, M. R. Ahmad // 2016 11th International Design & Test Symposium (IDT). – Hammamet, 2016. – P. 177-180. – doi: 10.1109/IDT.2016.7843036
5. Olivier K. Modern wideband DRFM architecture and real-time DSP capabilities for radar test and evaluation / K. Olivier, M. Gouws // 2013 Saudi International Electronics, Communications and Photonics Conference. – Fira, 2013. – PP. 1-4. – doi: 10.1109/SIECPC.2013.6551019
6. Strydom J. J. Hardware in the loop radar environment simulation on wideband DRFM platforms / J. J. Strydom, J. E. Cilliers, M. Gouws [et al.] // IET International Conference on Radar Systems (Radar 2012). – Glasgow, UK, 2012. – PP. 1-5. – doi: 10.1049/cp.2012.1687
7. Li. C. -z. Improved Interrupted Sampling Repeater Jamming based on DRFM / C. -z. Li, W. -m. Su, H. Gu [et al.] // 2014 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). – Guilin, 2014. – PP. 254-257. – doi: 10.1109/ICSPCC.2014.6986193
8. Xie M. Design and realization of DRFM system

- based on FPGA and DSP / M. Xie, J. Huang, Y. Jiang, X. Fu // IET International Radar Conference 2015. – Hangzhou, 2015. – PP. 1-6. – doi: 10.1049/cp.2015.1418.
9. **Peng, Z.** Realization of DRFM radar target simulator based on general instruments / Z. Peng // IET International Radar Conference 2015. – Hangzhou, 2015. – PP. 1-8, doi: 10.1049/cp.2015.1470.
10. **Arjoune, Y.** Spectrum sensing: Enhanced energy detection technique based on noise measurement / Y. Arjoune, Z. E. Mrabet, H. E. Ghazi, A. Tamtaoui // 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). – Las Vegas, NV, 2018. – PP. 828-834. – doi: 10.1109/CCWC.2018.8301619.
11. **Atapattu, S.** Energy Detection for Spectrum Sensing in Cognitive Radio / S. Atapattu, C. Tellambura, H. Jiang. – New York, NY: Springer New York, 2014. – 83 p.
12. **Lyons, R. G.** Understanding digital signal processing. / R. G. Lyons. – Upper Saddle River, N.J: Prentice Hall, 2011. – 954 pp.
13. **Hou, B.** Design and FPGA Implementation of High-speed Parallel FIR Filters / B. Hou, Y. Yao, M. Qin // 3rd International Conference on Mechatronics, Robotics and Automation. – Shanghai, 2015. – PP. 975–979. – doi: 10.2991/icmra-15.2015.189
14. **Vivado Design Suite User Guide: Partial Reconfiguration.** – Text : electronic // Xilinx Support. – 2018. – URL: [https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2018\\_1/ug909-vivado-partial-reconfiguration.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_1/ug909-vivado-partial-reconfiguration.pdf) (accessed: 15.10.2021).
- based on FPGA and DSP / M. Xie, J. Huang, Y. Jiang, X. Fu // IET International Radar Conference 2015. – Hangzhou, 2015. – PP. 1-6. – doi: 10.1049/cp.2015.1418
9. Peng Z. Realization of DRFM radar target simulator based on general instruments / Z. Peng // IET International Radar Conference 2015. – Hangzhou, 2015. – PP. 1-8, doi: 10.1049/cp.2015.1470
10. Arjoune Y. Spectrum sensing: Enhanced energy detection technique based on noise measurement / Y. Arjoune, Z. E. Mrabet, H. E. Ghazi, A. Tamtaoui // 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). – Las Vegas, NV, 2018. – PP. 828-834. – doi: 10.1109/CCWC.2018.8301619.
11. Atapattu S. Energy Detection for Spectrum Sensing in Cognitive Radio / S. Atapattu, C. Tellambura, H. Jiang. – New York, NY: Springer New York, 2014. – 83 p.
12. Lyons R. G. Understanding digital signal processing. / R. G. Lyons. – Upper Saddle River, N.J: Prentice Hall, 2011. – 954 pp.
13. Hou B. Design and FPGA Implementation of High-speed Parallel FIR Filters / B. Hou, Y. Yao, M. Qin // 3rd International Conference on Mechatronics, Robotics and Automation. – Shanghai, 2015. – PP. 975–979. – doi: 10.2991/icmra-15.2015.189
14. Vivado Design Suite User Guide: Partial Reconfiguration. – Text : electronic // Xilinx Support. – 2018. – URL: [https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2018\\_1/ug909-vivado-partial-reconfiguration.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_1/ug909-vivado-partial-reconfiguration.pdf) (accessed: 15.10.2021).

*Статья поступила в редколлегию 15.10.2021*

*Рецензент: канд. техн. наук, доц.*

*Брянский государственный технический университет*

*Горбунов А.Н.*

*Статья принята к публикации 22.10.2021*

#### Сведения об авторах

##### **Смоляков Андрей Владимирович**

студент второго курса магистратуры Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина)

E-mail: [andreismolyakow@gmail.com](mailto:andreismolyakow@gmail.com)

##### **Подстригаев Алексей Сергеевич**

кандидат технических наук, докторант, доцент кафедры радиоэлектронных средств Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина)

E-mail: [ap0d@ya.ru](mailto:ap0d@ya.ru)

#### Information about authors:

##### **Smolyakov A.V.**

Second year master's student, SPbETU «LETI»

E-mail: [andreismolyakow@gmail.com](mailto:andreismolyakow@gmail.com)

##### **Podstrigaev A.S.**

Candidate of Technical Sciences, doctoral student, Associate Professor, Department of Radioelectronic Facilities of SPbETU «LETI»

E-mail: [ap0d@ya.ru](mailto:ap0d@ya.ru)