

М. Н. Скрипниченко<sup>1, 2</sup>, И. А. Липатов<sup>1, 2</sup>

<sup>1</sup> Национальный исследовательский университет «МИЭТ», <sup>2</sup> АО «Научно-производственный центр «ЭЛВИС»»

## МАРШРУТ РАЗРАБОТКИ И ВЕРИФИКАЦИИ ЦИФРОВОЙ ЧАСТИ МНОГОСТАДИЙНОГО СИГМА-ДЕЛЬТА АЦП

Для решения ряда задач радиолокации требуются аналого-цифровые преобразователи (АЦП), обладающие одновременно и высоким отношением сигнала к шуму, и большой полосой пропускания сигнала. Задача проектирования подобного АЦП является настоящим вызовом при разработке аналоговой части, цифровой части и ее верификации. Маршрут разработки цифровой части должен учитывать возможность изменения спецификации аналоговой части на всех этапах проектирования, предоставлять возможность быстро получить синтезируемый RTL-код устройства и провести его функциональную верификацию. Для уменьшения временных затрат на разработку и верификацию в маршруте были использованы средства автоматизации. В данной статье описан разработанный программный комплекс, генерирующий синтезируемый RTL (Register Transfer Level) код и конфигурации верификационного окружения для каждого этапа разработки аналоговой части многостадийного сигма-дельта АЦП.

**Ключевые слова:** сигма-дельта АЦП, ЦОС, СнК, аппаратный СФ-блок, функциональная верификация

### Введение

Существует классическая схема (рис. 1) одностадийного сигма-дельта АЦП первого порядка, состоящая из вычитателя, интегратора, компаратора и однобитного цифро-аналогового преобразователя (ЦАП) в цепи обратной связи. Полученный сигнал проходит через децимирующий фильтр нижних частот, преобразуясь в многоразрядный.

В основе сигма-дельта преобразователей лежат два принципа – избыточной дискретизации и формирования шума. Формирование шума (noise shaping) приводит к тому, что основная мощность шума квантования лежит вне полосы полезного сигнала, а благодаря избыточной дискретизации возможно оперировать одноразрядными сигналами в аналоговой части, получая многоразрядный сигнал на выходе цифровой части.

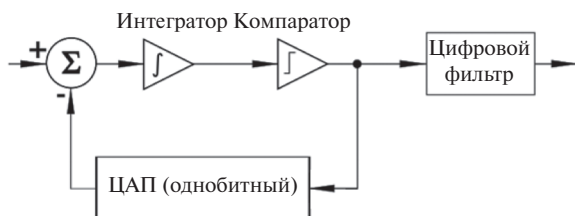


Рисунок 1. Классическая схема сигма-дельта АЦП первого порядка

При необходимости увеличить полосу сигналов, не уменьшая количество эффективных разрядов, используется схема с сигма-дельта АЦП второго или третьего порядка. Данная структура позволяет перенести мощность шума вне полосы сигнала, повышая отношение сигнал/шум и эффективное количество разрядов. Вместе с этим можно использовать многостадийную архитектуру, как показано на рис. 2, где каждая следующая стадия занимается оцифровкой ошибки квантования предыдущей стадии, уменьшая уровень шума в полосе сигнала и не влияя при этом на формирование шума.

### Разработка цифровой части АЦП

Аналоговая часть сигма-дельта преобразователя представляет собой аналоговый фильтр с определенной АЧХ. Для восстановления сигнала, пропущенного через этот фильтр, необходимо рассчитать обратный фильтр и произвести свертку сигнала с этим фильтром [1].

Цифровая часть классического одностадийного преобразователя сигма-дельта АЦП первого порядка представляет собой цифровой фильтр нижних частот с конечной импульсной характеристикой. По мере увеличения порядка сигма-дельта АЦП и количества стадий увеличивается и соответствующее количество фильтров, в общем случае – по одному на каждый выход.

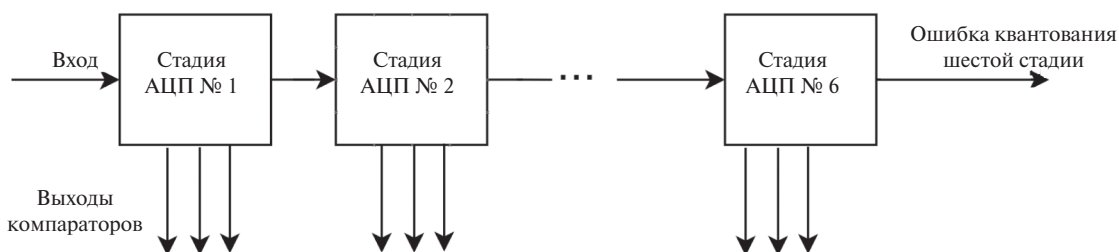


Рисунок 2. Сигма-дельта АЦП третьего порядка с шестью стадиями

Основная проблема состоит в том, что все эти фильтры довольно сильно чувствительны к разбросу, и необходимо уметь корректировать амплитудно-фазовую частотную характеристику этих фильтров непосредственно на кристалле.

Данная проблема была решена путем получения набора фиксированных фильтров для каждого из выходов аналого-цифрового преобразователя. Структурная схема фильтров приведена на рис. 3.

Задача расчета подобных фиксированных фильтров и написания соответствующего синтезируемого RTL-кода является весьма сложной. Более того, имеется явная зависимость от этапа аналогового проектирования – по мере замены идеальных компонент аналоговой части выполненными на транзисторах АФЧХ обратных фильтров постоянно меняются, а финальный результат после экстракции топологии довольно серьезно отличается от всех промежуточных. При этом необходимо на каждом этапе аналогового проектирования проверять, действительно ли можно подобрать такой набор фильтров, описывающий все имеющиеся АФЧХ обратных фильтров, адекватными ли

получаются коэффициенты перед фильтрами, а также написать RTL-код и верифицировать его.

Процесс написания RTL-кода, содержащего коэффициенты фильтров, и его верификация являются нетривиальными, и их переписывание занимает много времени.

Поэтому при построении модели цифровой части АЦП было предложено сделать автоматическую генерацию RTL-кода цифровой части. Тогда он получает следующие преимущества:

- можно легко переделать;
- низкая вероятность ошибки;
- можно получить предварительную оценку по площади, таймингу и потреблению.

Таким образом, на начальных этапах разработки цифровой части АЦП основной упор был сделан не на написание RTL-кода, а на создание программного комплекса, позволяющего по набору входных данных генерировать требуемый RTL-код.

Глобально весь код можно разделить на две части – постоянный и настраиваемый. К постоянному

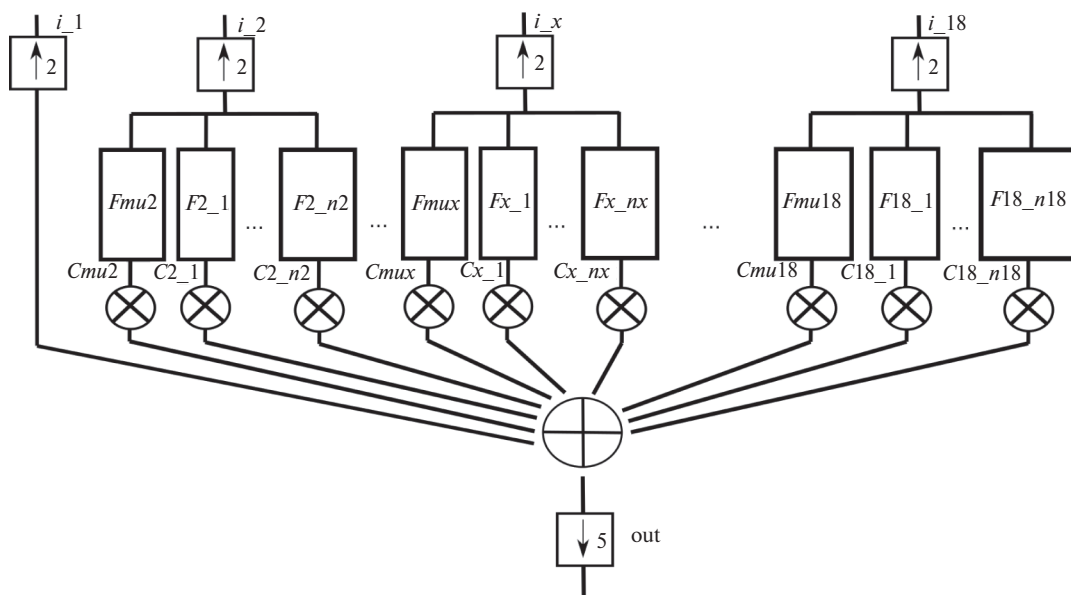


Рисунок 3. Восстановление сигнала обратными фильтрами

коду относятся часть регистрового файла, управляющие автоматы памяти и заготовки блоков настраиваемого кода. К настраиваемому коду относятся непосредственно цифровые фильтры и часть регистрового файла.

### Структурная схема цифровой части

На рис. 4 представлена структурная схема цифровой части АЦП, состоящая из блока перехода между двумя частотными доменами, блока сдвиговых регистров, непосредственно фильтров и регистрового файла с весовыми коэффициентами фильтров.

Изначально аналоговый блок выдает 25 отсчетов с частотой дискретизации 2000 МГц. Для перехода в частотный домен 400 МГц используется схема из двух параллельных сдвиговых регистров. Первый регистр состоит из пяти триггеров, работает на частоте 2000 МГц и за каждый такт сдвигает один отсчет. Второй регистр аналогично состоит из пяти триггеров, работает на частоте 400 МГц и за каждый такт копирует содержимое первого регистра.

Благодаря такой схеме становится возможно работать с отсчетами, имеющими частоту дискретизации 2 ГГц, в доменной зоне 400 МГц.

Эти отсчеты являются однобитными, что упрощает разработку фильтров. Вместо классической схемы с умножителями и сумматорами возможно перейти к использованию перекодировочной таблицы (LookUp Table, LUT). Благодаря заранее рассчитанным значениям для всех входных вариантов можно получить солидный выигрыш по задержкам, площади и потреблению. Исследования показали, что наиболее эффективный вариант по площади, удовлетворяющий требованиям по задержке, получается при использовании LUT с шестью входами.

Задача расчета обратных фильтров с подобной АЧХ является весьма нетривиальной, и классические способы построения фильтров с заранее заданной АЧХ не могут продемонстрировать удовлетворительных результатов. В связи с этим задача построения обратных фильтров разделилась на две

подзадачи – построение дециматора, удовлетворяющего перечисленным выше требованиям, и построение обратных фильтров, реализующих АФЧХ в полосе до 300 МГц.

Рассчитав и дециматор, и обратные фильтры, путем их свертки можно получить набор результирующих фильтров, одновременно и осуществляющий децимацию сигнала, и восстанавливающий исходный сигнал.

Часть программного комплекса, отвечающая за генерацию АЧХ, работает с.mat-файлом, содержащим структуру со следующими полями:

- коэффициенты фильтра;
- задержка фильтра;
- порядок фильтра.

Все фильтры сворачиваются с рассчитанным ранее дециматором. Далее на основе коэффициентов фильтра генерируется код на языке описания аппаратуры Verilog, представляющий собой модули, состоящие из наборов LUT, по два модуля для каждого фильтра. В соответствии с выбранным методом передискретизации первый субфильтр представляет собой только четные коэффициенты фильтра, а второй – только нечетные. Кроме того, на второй фильтр отсчеты поступают с задержкой в два такта для согласования выходных отсчетов – при децимации из каждых пяти отсчетов оставляется только первый.

В табл. 1 приведен принцип децимации – серым цветом выделен каждый пятый отсчет. Кроме того, из этой таблицы видно, почему необходимо задерживать отсчеты для второго фильтра.

Генератор LUT-модулей написан на языке C++ и требует в качестве входных данных коэффициенты фильтра, количество входов в одной LUT и количество разрядов, отведенных на представление коэффициентов фильтра. Эти данные генерируются в виде исходного кода на языке C++, затем выполняется его компиляция и выполнение полученного исполняемого файла.

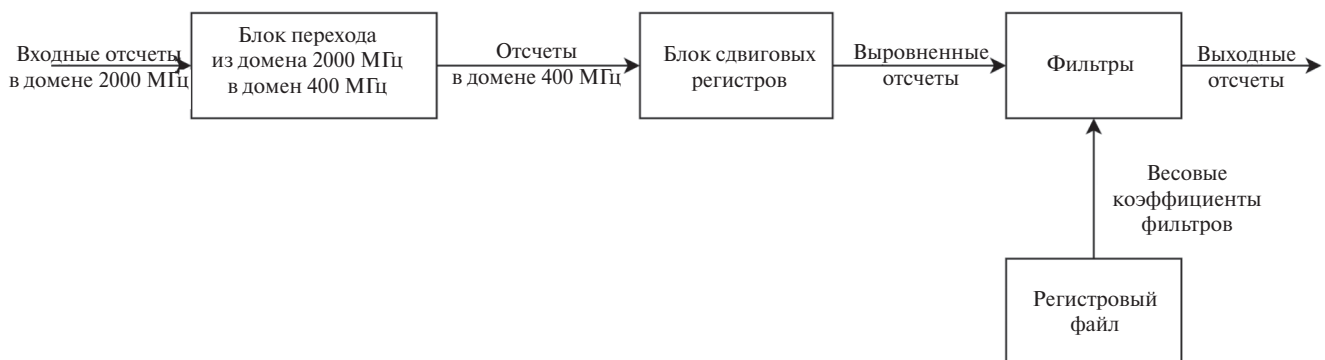


Рисунок 4. Структурная схема цифровой части АЦП

Таблица 1. Децимация отсчетов

Номер фильтра	Выходные отсчеты							
	14	12	10	8	6	4	2	0
Выход первого фильтра	14	12	10	8	6	4	2	0
Выход второго фильтра	15	13	11	9	7	5	3	1

В случае, если порядок фильтра не делится на количество входов в одной LUT нацело, коэффициенты фильтра дополняются нулями вплоть до количества, когда это условие будет соблюдаться.

Отдельно выполняется генерация RTL-кода на языке SystemVerilog, содержащая имплементацию модулей фильтров. Расчет коэффициентов фильтров выполняется для нескольких вариантов количества коэффициентов и задержек фильтра, так что в итоге каждый фильтр обладает своей задержкой, которую необходимо компенсировать. Первый фильтр в силу единичного коэффициента обладает нулевой задержкой.

Для компенсации задержки необходимо иметь сдвиговой регистр, превышающий по длине порядок фильтра. Следует помнить, что для каждого выхода существует несколько фильтров, у каждого из которых может быть свой порядок и своя задержка.

Итоговая формула для расчета длины сдвигового регистра  $i$ -го выхода имеет вид

$$sh\_len = \max(flts\_ord_i + 2 + \max(flts\_dly) - flts\_dly_i), \quad (1)$$

где  $flts\_ord_i$  – набор значений порядков фильтров для  $i$ -го выхода;  $flts\_dly$  – набор значений задержки фильтров для всех выходов;  $flts\_dly_i$  – набор значений задержки фильтров для  $i$ -го выхода.

Для каждого фильтра ищется необходимая для него длина регистра, состоящая из порядка фильтра и разницы между наибольшей задержкой и задержкой самого фильтра. Затем среди найденных величин ищется наибольшее значение.

После имплементации сдвиговых регистров идет процесс генерации кода, отвечающего за фильтры. К этому относится генерация модулей-оберток вокруг LUT, обеспечивающих задержку на два такта, и код с имплементацией этих модулей. Выход каждого фильтра умножается на соответствующий весовой коэффициент, сдвигается на несколько разрядов и суммируется с взвешенными выходами других фильтров. По мере увеличения количества фильтров усложняется и процесс логического и физического синтеза, поэтому был предусмотрен вариант с имплементацией конвейерных умножителей как самых больших блоков. Стоит отметить, что в определенный момент задержка на умножителях действительно стала больше допустимой, и генерация альтернативного кода с конвейерными умножителями исправила эту проблему, не вызвав никаких трудозатрат [2].

Важной частью сгенерированного кода является также регистровый файл. Часть регистрового файла заранее жестко определена и не изменяется вне зависимости от используемых фильтров. Другая же часть, например регистры с весовыми коэффициентами фильтров, каждый раз генерируется заново.

Помимо самого регистрового файла, на основе этих данных генерируется часть тестового окружения – заголовочные файлы с названием регистров, адресами, масками и начальными значениями. Эти файлы используются при тесте адресного пространства АЦП и для записи весовых коэффициентов при восстановлении сигнала.

### Методология функциональной верификации

Процесс функциональной верификации цифровой части проектируемого АЦП был построен с использованием принципов верификации, основанной на ограниченной рандомизации конфигурации устройства (Constrained-Random Verification). В рамках стратегии верификации при написании тестового плана использовались также принципы прямого тестирования (Directed Verification Methodology) целевого функционала, корректность реализации которого необходимо было проверить. Но такое направленное тестирование является трудозатратным процессом [3]. Если же ориентироваться на полностью случайную генерацию конфигураций и входных воздействий, это может привести в конечном итоге к трате впустую множества циклов моделирования без повышения функционального покрытия.

Для проверки корректности доступа к адресному пространству цифровой части АЦП был разработан тестовый сценарий, основанный на упомянутой выше методологии ограниченной рандомизации. Для записи в адресное пространство формировался случайный битовый вектор необходимой разрядности, производилась запись, а потом – чтение, после которого исходный битовый вектор маскировался специфицированным значением и сверялся со считанным. Если обнаруживались расхождения, то верификационная среда свидетельствовала о наличии ошибки доступа к конкретному элементу адресного пространства цифровой части.

Построение тестового сценария проверки конкретного функционала разрабатываемого устройства зависит от выбранной стратегии верификации.

Для анализа корректности восстановления сигнала цифровой частью АЦП можно использовать ее математическую модель и, подавая одни и те же входные данные на модель и устройство, сравнивать выходные данные с учетом неидеальностей реального устройства. Альтернативной стратегией может выступать анализ ключевых характеристик выходных данных, снимаемых с выхода блока цифровой части. Далее приведены определения параметров выходного сигнала цифровой части АЦП, выбранных в качестве ключевых, при разработке тестового сценария проверки восстановления сигнала.

Отношение сигнал/шум Signal-to-Noise Ratio (SNR):

$$SNR_{dB} = 10 \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right) = 20 \log_{10} \left( \frac{RMS_{\text{signal}}}{RMS_{\text{noise}}} \right), \quad (2)$$

где  $RMS_{\text{signal}}$  – среднеквадратичная амплитуда полезного сигнала; а  $RMS_{\text{noise}}$  – среднеквадратичная амплитуда шумовой составляющей сигнала [4].

Отношение сигнал/шум и коэффициент искажения (Signal-to-Noise and Distortion Ratio, SINAD). SINAD – это параметр, характеризующий деградацию полезного сигнала ввиду наличия посторонних сигналов, в частности шума и искажений. SINAD можно определить как отношение общего уровня мощности сигнала (Signal + Noise + Distortion) к нежелательной мощности сигнала (Noise + Distortion):

$$SINAD_{dB} = 10 \log_{10} \left( \frac{P_{\text{signal}} + P_{\text{noise}} + P_{\text{distortion}}}{P_{\text{noise}} + P_{\text{distortion}}} \right). \quad (3)$$

Динамический диапазон, свободный от паразитных составляющих (Spurious-Free Dynamic Range, SFDR), – отношение среднеквадратичного значения полезного сигнала к среднеквадратичному значению наихудшего паразитного сигнала  $RMS_{\text{max\_spur}}$  независимо от того, где он проявляется в частотном спектре:

$$SFDR_{dBc} = 20 \log_{10} \left( \frac{RMS_{\text{signal}}}{RMS_{\text{max\_spur}}} \right). \quad (4)$$

SFDR является важным параметром в системах связи, поскольку он характеризует наименьшую величину сигнала, которую можно отличить от большого паразитного сигнала [5].

В случае использования стратегии верификации, основанной на анализе выбранных ключевых параметров, существенно упрощается структура верификационного окружения при сохранении качественных характеристик верификации: если цифровая часть работает неправильно, вносит искажения в выходной сигнал, теряет отсчеты, это непременно отразится на характеристиках выходного сигнала и будет выявлено на этапе верификации.

### Компоненты верификационного окружения

Верификационное окружение представляет собой иерархическую структуру, предоставляющую высокоуровневые интерфейсы для управления процессом взаимодействия с верифицируемым устройством. Базисом для такого окружения является библиотека классов, упрощающая написание адаптируемого и повторно используемого кода. Автору тестового сценария необходимо лишь описать желаемые воздействия в терминах вызовов соответствующих интерфейсов, реализуемых верификационным окружением. Возможность повторного использования кода заключается в том, что компоненты окружения не зависят друг от друга. Таким образом, используя библиотеку классов, можно переносить окружение от проекта к проекту [3].

Для взаимодействия с устройством было собрано верификационное окружение, состоящее из интерфейсных драйверов, генераторов конфигураций, блоков загрузки конфигураций, выходных мониторов, а также вспомогательных управляющих агентов. Упрощенная схема соединения компонентов верификационного окружения приведена на рис. 5.

Драйверы интерфейсов осуществляют передачу управляющих воздействий в соответствии с конкретным стандартом. Блок эмуляции работы аналоговой части АЦП использовался для подачи входного сигнала на порты цифровой части АЦП. С помощью выходных мониторов осуществлялся прием выходных отсчетов цифровой части АЦП для проведения дальнейшего анализа. После приема выборки отсчетов достаточного размера проводился расчет ключевых характеристик сигнала: SNR, SFDR и SINAD. Если рассчитанные для принятой выборки значения этих параметров не попадали в заранее специфицированный диапазон, то верификационное окружение сигнализировало об ошибке. На рис. 6 приведены спектр и параметры выходного сигнала АЦП для разных конфигураций моделирования цифровой части.

Блоки выходных мониторов вкупе с компараторами использовались для проверки корректности интеграции цифровой части АЦП с блоком ЦОС, в который в тестах цифровой части с помощью конфигурационного агента загружалась конфигурация сквозного пропуска отсчетов. Компараторы сравнивали отсчеты с непосредственного выхода цифровой части и с выхода блока обработки сигналов, а в случае несовпадения или отсутствия отсчетов сигнализировали об ошибке. С помощью агента адресного пространства в верифицируемое устройство загружалась необходимая конфигурация весовых коэффициентов, сформированная на этапе генерации RTL-кода. Такого рода интеграция этапа

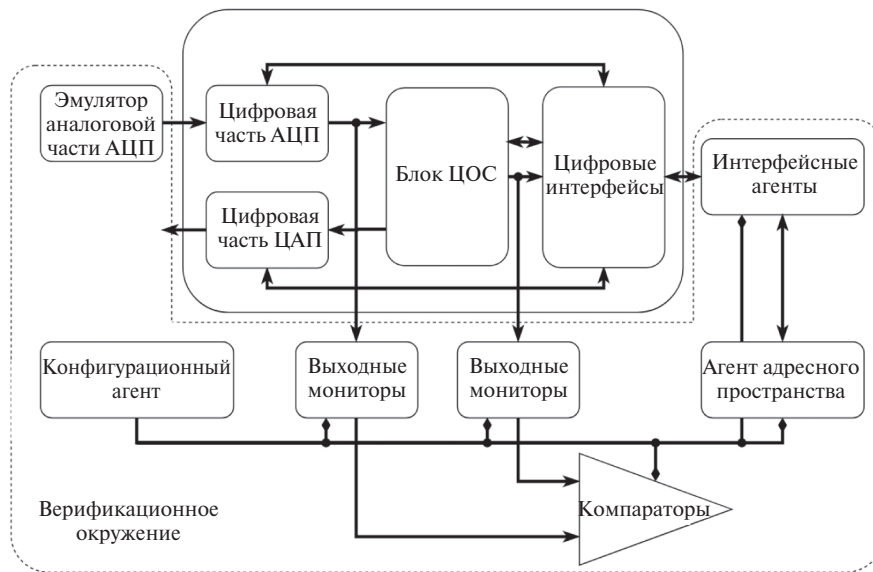


Рисунок 5. Компоненты верификационного окружения, используемые в тестах цифровой части АЦП

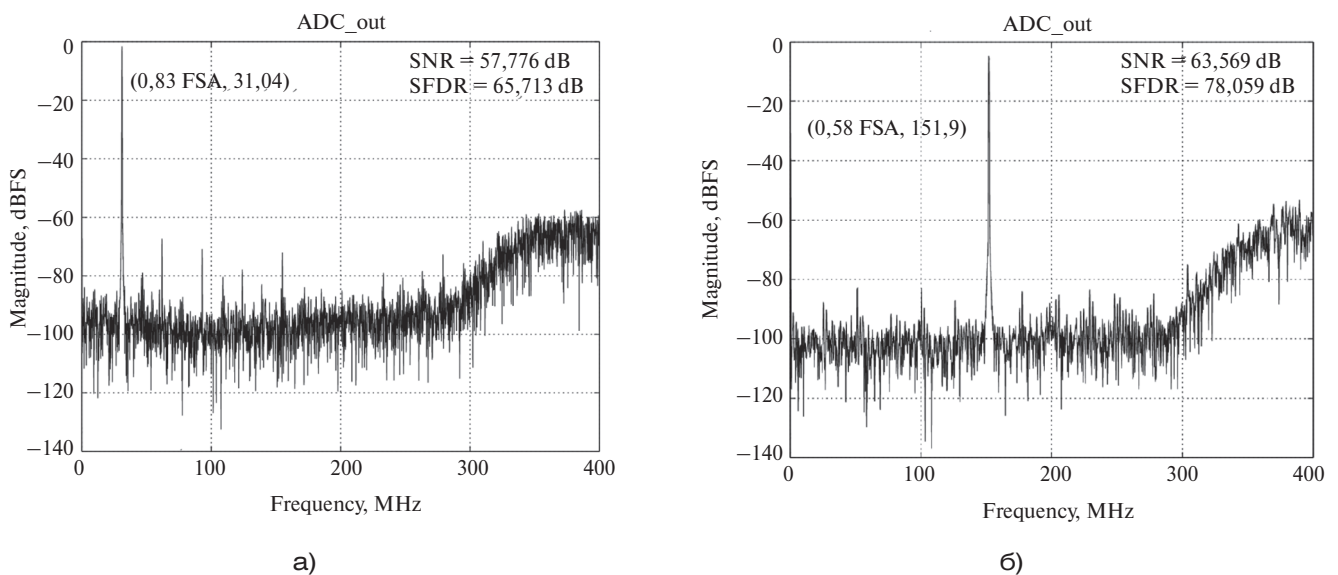


Рисунок 6. Спектр выходного сигнала цифровой части АЦП для разных конфигураций моделирования цифровой части: а – в качестве весов фильтров использовались весовые коэффициенты фильтров для номинальных АЧХ, а в качестве входных данных – отсчеты, соответствующие отсчетам канала I квадратурного АЦП с использованием гетеродирования; б – в качестве весовых коэффициентов фильтров и входных данных использовались веса и отсчеты для отдельного АЦП при моделировании параметров техпроцесса по методу Монте-Карло

генерации RTL-кода и процесса верификации обеспечивает необходимый уровень адаптивности к изменчивости спецификации устройства и, как следствие, возможность реорганизации компонентов окружения в сжатые сроки.

Для автоматизации запусков тестовых сценариев был разработан и внедрен программный пакет, осуществляющий автоматизированный запуск и генерацию отчетов о прохождении верифицируемым устройством тестовых сценариев согласно

принципам регрессионного тестирования. В генерируемый отчет включалась информация о воздействиях, поданных на устройство, а также об условиях воспроизведения данного моделирования.

### Результаты исследований

В ходе работы над цифровой частью шестистадийного сигма-дельта АЦП третьего порядка была разработана математическая модель, позволяющая генерировать синтезируемый код

Таблица 2. Ожидаемая и реальная площади синтезированных фильтров на различных стадиях аналогового проектирования

Этап проектирования	Суммарное количество коэффициентов в фильтрах	Вес одного коэффициента, мкм <sup>2</sup>	Ожидаемая площадь, мкм <sup>2</sup>	Реальная площадь, мкм <sup>2</sup>
I	9840	113	–	1 119 433
II	10944	119	1 236 672	1 296 490
III	11 520	118	1 301 760	1 361 130

цифровой части на языке описания аппаратуры SystemVerilog и соответствующее этому коду верификационное окружение. Это позволило на ранних этапах аналогового проектирования получить начальную оценку площади, задержек и потребления и вывести вес одного коэффициента фильтра в единицах измерения площади для получения предварительной оценки без процедуры синтеза. В табл. 2 приведены результаты синтеза на каждой стадии, выведен вес одного коэффициента фильтра и дана предварительная оценка площади.

Синтез фильтров проводился с использованием САПР Synopsys Design Compiler и библиотеки TSMC90 nm Standard Cell Library.

В табл. 2 использованы следующие обозначения этапа проектирования:

- I – фильтр на транзисторном уровне + идеальные элементы;
- II – полная аналоговая часть на транзисторном уровне;

- III – полностью аналоговая часть на транзисторном уровне + результаты экстракции топологии.

Как можно видеть, разница между предсказанной площадью и полученной составляет не более 5%, что является весьма хорошим результатом.

### Выводы

В статье описан маршрут разработки и верификации цифровой части многостадийного сигма-дельта АЦП. Был предложен и разработан программный комплекс, генерирующий синтезируемый RTL-код на языке SystemVerilog для семейства многостадийных сигма-дельта АЦП с произвольным количеством порядков и стадий и элементы верификационного окружения. Разработанный программный комплекс обеспечивает адаптивность к изменениям в спецификации устройства и возможность быстрого перестроения процесса верификации. Апробация данного маршрута проводилась при разработке и верификации двух АЦП в составе микросхемы АФАР 1288ТК015.

### СПИСОК ЛИТЕРАТУРЫ

1. Айфичер Э., Джервис Б. Цифровая обработка сигналов. Практический подход. 2004. С. 230–235.
2. Bhatnagar H. Advanced ASIC Chip Synthesis Using Synopsys® Design Compiler™ Physical Compiler™ and PrimeTime®. Springer Science & Business Media, 2012, 284 p.
3. Mehta A. B. ASIC/SoC Functional Design Verification A Comprehensive Guide to Technologies and Methodologies. Springer, 2018, pp. 65–70.
4. Carusone T., Johns D., Martin K. Analog Integrated Circuit Design, 2nd edition. Wiley, 2011, 365 p.
5. Kester W. Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor. Analog Devices Inc., 2009, pp. 1–8.

### ИНФОРМАЦИЯ ОБ АВТОРАХ

**Скрипниченко Максим Николаевич**, аспирант кафедры ПКИМС, Национальный исследовательский университет «МИЭТ», 124498, Москва, Зеленоград, площадь Шокина, д. 1, стр. 7, тел.: 8 (965) 126-15-53; инженер, АО «Научно-производственный центр «ЭЛВИС»», 124498, Москва, Зеленоград, проезд № 4922, д. 4, стр. 2, тел.: 8 (495) 926-79-57, e-mail: mskripnichenko@elvees.com.

**Липатов Иван Алексеевич**, аспирант кафедры ПКИМС, Национальный исследовательский университет «МИЭТ», 124498, Москва, Зеленоград, площадь Шокина, д. 1, стр. 7, тел.: 8 (965) 126-15-53; инженер, АО «Научно-производственный центр «ЭЛВИС»», 124498, Москва, Зеленоград, проезд № 4922, д. 4, стр. 2, тел.: 8 (495) 926-79-57, e-mail: ilipatov@elvees.com.

*For citation: Skripnichenko M.N., Lipatov I.A. Design and verification flow of multi-stage sigma-delta ADC digital core. Voprosy radioelektroniki, 2018, no. 8, pp. 56–63. DOI 10.21778/2218-5453-2018-8-56-63*

**M. N. Skripnichenko, I. A. Lipatov**

## **DESIGN AND VERIFICATION FLOW OF MULTI-STAGE SIGMA-DELTA ADC DIGITAL CORE**

There is a need for analog-to-digital converters with high signal-to-noise ratio and large signal bandwidth to solve a number of radiolocation problems. Developing such ADC is a challenge in the analog core, digital core and verification. The design flow of the digital core must take into account the possibility of changing the analog core specification at any design stage, provide the ability to quickly obtain the synthesizable RTL code of the device and conduct its functional verification. Automation tools were used to reduce the time spent on development and verification. This article describes the developed software package that generates the synthesizable RTL code and the verification environment configurations for each stage of development of the analog core of the multi-stage sigma-delta ADC.

**Keywords:** sigma-delta ADC, DSP, SoC, hardware IP, functional verification.

### **REFERENCES**

1. Ifeachor E., Jervis B. Digital Signal Processing: A Practical Approach, 2nd edition, Prentice Hall, 2001, pp. 230–235.
2. Bhatnagar H. Advanced ASIC Chip Synthesis Using Synopsys® Design Compiler™ Physical Compiler™ and PrimeTime®. Springer Science & Business Media, 2012, 284 p.
3. Mehta A. B. ASIC/SoC Functional Design Verification A Comprehensive Guide to Technologies and Methodologies. Springer, 2018, pp. 65–70.
4. Carusone T., Johns D., Martin K. Analog Integrated Circuit Design, 2 edition. Wiley, 2011, 365 p.
5. Kester W. Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor. Analog Devices Inc., 2009, pp. 1–8.

### **AUTHORS**

**Skripnichenko Maksim**, postgraduate student, National Research University of Electronic Technology, 1/7, ploshchad Shokina, Zelenograd, Moscow, Russian Federation, 124498, tel.: +7 (965) 126-15-53; engineer, Joint-Stock Company Research and Development Center ELVEES, 4/2, proezd № 4922, Zelenograd, Moscow, 124498, Russian Federation, tel.: +7 (495) 926-79-57, e-mail: mskripnichenko@elvees.com.

**Lipatov Ivan**, postgraduate student, National Research University of Electronic Technology, 1/7, ploshchad Shokina, Zelenograd, Moscow, Russian Federation, 124498, tel.: +7 (965) 126-15-53; engineer, Joint-Stock Company Research and Development Center ELVEES, 4/2, proezd № 4922, Zelenograd, Moscow, 124498, Russian Federation, tel.: +7 (495) 926-79-57, e-mail: ilipatov@elvees.com.