



DOI: 10.15514/ISPRAS-2025-37(2)-3

Методы схемотехнического и топологического синтеза аналоговых интегральных схем по спецификации с использованием методов машинного обучения и дифференцируемого программирования

¹ Д.С. Щепетов, ORCID: 0009-0005-3562-6374 <dshchepetov@synth-cad.ru>

² А.Г. Тимошенко, ORCID: 0000-0002-2018-165X <timoshenko@org.miet.ru>

¹ В.А. Гаранжа, ORCID: 0000-0002-8376-756X, <garan@ccas.ru>

¹ И.Е. Капорин, ORCID: 0000-0002-1451-1632 <igorkaporin@mail.ru>

³ Д.Н. Каргин, ORCID: 0009-0002-7394-0370 <dmitry.n.kargin@synth-cad.ru>

¹ Федеральный исследовательский центр "Информатика и управление" РАН, Россия, 119333, г. Москва, Вавилова, д.44, кор.2.

² Национальный исследовательский университет «МИЭТ», Россия, 124498, г. Москва, г. Зеленоград, площадь Шокина, дом 1.

³ Московский политехнический университет, Россия, 107023, г. Москва, ул. Б. Семёновская, д. 38.

Аннотация. Задача разработки принципиальной электрической схемы, возникающая при создании аналоговых интегральных схем, сформулирована как задача оптимизации дифференцируемой гладкой функции с помощью комбинации методов дифференцируемого программирования и методов машинного обучения. Показано, что такой подход позволяет достичь требований спецификации и предложить оптимальную комбинацию схемотехнических шаблонов, составляющих аналоговую интегральную схему, без привлечения методов комбинаторной оптимизации и обучения с подкреплением. Показано, что такой подход дает значительные преимущества по скорости по сравнению с традиционными методами, основанными на обучении с подкреплением. Исследована возможность полностью автоматического синтеза аналоговой ИС от спецификации до топологии без участия эксперта с помощью программных средств с открытым исходным кодом. Показаны преимущества и недостатки такого подхода.

Ключевые слова: проектирование аналоговых интегральных схем; дифференцируемое программирование; машинное обучение; схемотехнический синтез.

Для цитирования: Щепетов Д.С., Тимошенко А.Г., Гаранжа В.А., Капорин И.Е., Каргин Д.Н. Методы схемотехнического и топологического синтеза аналоговых интегральных схем по спецификации с использованием методов машинного обучения и дифференцируемого программирования. Труды ИСП РАН, том 37, вып. 2, 2025 г., стр. 33–48. DOI: 10.15514/ISPRAS–2025–37(2)–3.

Methods of Circuit and Topological Synthesis of Analog Integrated Circuits According to Specification Using Machine Learning and Differentiable Programming Methods

¹ D.S. Shchepetov, ORCID: 0009-0005-3562-6374 <dshchepetov@synth-cad.ru>

² A.G. Timoshenko, ORCID: 0000-0002-2018-165X <timoshenko@org.miet.ru>

¹ V.A. Garanzha, ORCID: 0000-0002-8376-756X <garan@ccas.ru>

¹ I.E. Kaporin, ORCID: 0000-0002-1451-1632 <igorkaporin@mail.ru>

³ D.N. Kargin, ORCID: 0009-0002-7394-0370 <dmitry.n.kargin@synth-cad.ru>

¹ Federal Research Center "Informatics and Control" of the Russian Academy of Sciences",
Russia, 119333, Moscow, Vavilova st., 44, bldg. 2.

² National Research University "MIET",
Russia, 124498, Moscow, Zelenograd, Shokina square, bldg. 1.

³ Moscow Polytechnic University,
Russia, 107023, Moscow, B. Semenovskaya st., 38.

Abstract. The problem of schematic development (netlist generation), which arises in the development of analog integrated circuits, is formulated as an optimization problem for a differentiable smooth function using a combination of differentiable programming and machine learning methods. It is shown that this approach allows one to achieve the specification requirements and propose an optimal combination of circuit templates that make up an analog integrated circuit, without involving combinatorial optimization and reinforcement learning methods. It is shown that this approach provides significant speed advantages compared to traditional methods based on reinforcement learning. The possibility of fully automatic synthesis of an analog IC from specification to topology without expert participation using open-source software is investigated. The advantages and disadvantages of this approach are shown.

Keywords: analog integrated circuits design; machine learning; differentiable programming; schematic synthesis.

For citation: Shchepetov D.S., Timoshenko A.G., Garanzha V.A., Kaporin I.E., Kargin D.N. Methods of circuit and topological synthesis of analog integrated circuits according to specification using machine learning and differentiable programming methods. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 2, 2025. pp. 33-48 (in Russian). DOI: 10.15514/ISPRAS-2025-37(2)-3.

1. Введение

Проектирование аналоговых интегральных схем (ИС), как схемотехническое, так и топологическое, исторически сопряжено с применением большого количества ручного, экспертного труда и обладает меньшей степенью автоматизации, чем проектирование цифровых ИС. Это обуславливается следующими причинами: вариативностью размеров транзисторов в аналоговом режиме, широкой номенклатурой режимов их работы, существенно большей чувствительностью к шумам по сравнению с цифровыми ИС и, соответственно, большей требовательностью к взаимному расположению транзисторов в топологическом проектировании. В связи с этим проектирование аналоговых ИС, например, топологическое, рассматривалось как "искусство", главным образом в том смысле, что автоматизация этого процесса невозможна и неэффективна («проверка алгеброй гармонии»), и эта точка зрения даже отразилась в названии книги по топологическому проектированию [1]. Способ («маршрут») проектирования аналоговых ИС имеет 4 стадии:

- выбор схемотехнического решения (разработка схемы электрической принципиальной – СЭП) на основе спецификации ИС. Стадия осуществляется полностью экспертом;

- определение размеров транзисторов и параметров пассивных элементов (т.н. «сайзинг» от англ. sizing), составляющих ИС. Стадия осуществляется либо экспертом на основании упрощенных моделей транзисторов, либо методами оптимизации типа «черный ящик». Валидация результатов осуществляется с помощью средства схемотехнического моделирования (CCM);
- топологическое размещение транзисторов и пассивных элементов с учетом их параметров. Стадия осуществляется экспертом-топологом;
- верификация – проверка на соответствие правилам проектирования (Design Rules Check, DRC), проверка на соответствие топологии схеме (Layout-versus-Schematic, LvS), определение значений паразитных элементов (Parasitic Extraction, PEX). Стадия осуществляется специализированными решателями, первая «прикидка» исходит из интуиции эксперта-тополога и эксперта-схемотехника.

Этот маршрут редко проходится один раз, возвраты на предыдущие стадии из-за невозможности выполнить требования спецификации возможны и вероятны.

Исследования последних лет в области методов машинного обучения (МО) позволили сделать важные шаги к разработке нового почти автоматического маршрута. В области схемотехнического синтеза (выбора схемотехнического решения) был представлен ряд работ, основанных на обучении с подкреплением (RL – англ. Reinforcement Learning) [2, 3]. В области сайзинга аналоговых ИС интересна работа [4], в которой авторы отказались от «золотого стандарта» сайзинга – методов байесовской оптимизации – и перешли к автокодировщикам, которые учились сопоставлять характеристикам ИС параметры транзисторов непосредственно в процессе оптимизации. В области топологического проектирования были разработаны два программных пакета [5, 6], которые позволяли проводить топологическое проектирование в автоматическом режиме в два шага: на первом с помощью графовых искусственных нейронных сетей (ИНС) выделялись условия на симметричность и компактность расположения, на втором – решалась задача оптимизации с учетом детектированных условий методами оптимизации с ограничениями.

Таким образом, с использованием упомянутых работ, появилась возможность построить контур проектирования аналоговых ИС без участия человека-эксперта. В более реальном сценарии: роль эксперта сводится к высокоуровневому дизайну и заданию ограничений, рутинные операции берет на себя автоматическая система. Однако, современные методы МО требовательны к объему и составу обучающей выборки, что делает их обучение трудоемким (во всех отношениях) процессом, а также требуют специальных знаний для выбора метода МО и для подготовки и очистки данных. Эксперт-схемотехник или тополог редко обладает такими знаниями, поэтому актуальными остаются несколько направлений, которые позволяют существенно сократить размер обучающей выборки и повысить интерпретируемость результатов, а именно:

- проектирование ИС в соответствии с определенным набором правил. Например, в работе [7] был разработан набор составных частей и связей, позволяющих описать все операционные усилители (ОУ), что позволяет синтезировать обучающую выборку в автоматическом режиме на основе алгоритмов обучения с подкреплением;
- «полуклассический» подход – «дифференцируемое программирование» (ДП) – в котором часть алгоритма записывается в строгом соответствии с правилами в виде классической программы, а часть алгоритма с помощью методов машинного обучения. При этом количество настроечных параметров у такой программы велико, а алгоритм должен быть составлен таким образом, чтобы между шагами алгоритма можно было вычислить градиент по настроечным параметрам.

Таким образом, можно сформулировать цель данного исследования:

- на основе выбранного объекта исследования – аналого-цифрового преобразователя последовательного приближения (АЦП-ПП) – исследовать возможность их автоматизированного синтеза с помощью методов дифференциального программирования и методов машинного обучения;
- проверить эффективность предложенного подхода и сравнить его с классическими методами как машинного обучения, так и обучения с подкреплением;
- проверить эффективность сквозного проектирования от спецификации до топологии с использованием средств автоматического топологического размещения.

Статья организована следующим образом. Раздел 2 содержит основные сведения о дифференцируемом программировании и его применении к рассматриваемой задаче, а также описание схемы работы АЦП-ПП и реализацию дифференцируемой программы, которая описывает поведение данного типа АЦП. Раздел 3 содержит описание используемых методов машинного обучения для решения поставленной задачи, а также краткое описание основные трудностей, с которыми столкнулся предполагаемый подход. Раздел 4 содержит описание эксперимента по применению разработанных методов и сравнение его с актуальными методами решения поставленной задачи, которые распространены в литературе. В разделе представлен анализ результатов эксперимента. Также в разделе описаны результаты автоматического создания топологии в пакете с открытым исходным кодом – Align. Раздел 5 содержит основные выводы по рассмотренной в работе проблеме.

2. Дифференцируемое программирование для синтеза ИС

2.1. Общие сведения

Согласно [8] введем определение дифференцируемой программы как композиции гладких, дифференцируемых преобразований (f_k) над состояниями программы S_k (под состоянием понимается совокупность переменных и параметров программы на i -том шаге):

$$S_k = f_k(S_{k-1}). \quad (1)$$

Финальное состояние S_n представляет собой результат работы программы. Начальное состояние S_0 представляет собой совокупность настроечных параметров и значений «входа» программы. В случае, если дана выборка начальных (S_0) состояний программы и желаемых конечных (O) состояний программы – $(S_{0,j}, O_j)$, то оптимальные значения настроечных параметров p будут определяться как аргумент, минимизирующий заданную невязку L :

$$p = \operatorname{argmin}_p \sum_j L(S_{n,j}, O_j).$$

Если все функции f_k в (1) дифференцируемые и гладкие, то возможно получить производные $\nabla_p S_n$ по параметрам алгоритма p , используя, например, методы автоматического дифференцирования [9] и проводить минимизацию невязки с помощью процедур типа «градиентный спуск».

Целесообразно задавать невязку L со следующими свойствами:

$$\begin{aligned} L(S_{n,j}, O_j) &= 0, \text{ если } S_{n,j} \equiv O_j, \\ \nabla_{f_{n,j}} L(S_{n,j}, O_j) &= 0, \text{ если } S_{n,j} \equiv O_j, \end{aligned} \quad (2)$$

где знак \equiv используется как совпадение с заданной точностью входного и выходного состояний, $\nabla_{f_{n,j}}$ – градиент невязки по переменным состояний $S_{n,j}$. Это делается по следующей причине. Выражение для невязки градиента по параметрам (p) имеет вид:

$$\nabla_{f_{n,j}} L(S_{n,j}, O_j) = \nabla_{f_{n,j}} L \cdot \prod_{i=n}^{i=1} \nabla_{f_{i-1,j}} S_{i,j} \cdot \nabla_p S_{0,j} \quad (3)$$

В случае совпадения выходов с желаемыми выходами для всех образцов выражение (3) с учетом условия (2) равно 0, что фиксирует параметры и препятствует их дальнейшему изменению в градиентном спуске.

Упрощенный алгоритм поиска оптимального решения с помощью градиентного спуска представлен в алгоритме 1.

Алгоритм 1. Алгоритм поиска оптимального решения для дифференцируемой программы.

Algorithm 1. Algorithm for finding the optimal solution of a differential program.

Вход: начальное значение параметров p_0 , функция невязки L , набор из желаемых входных и выходных состояний $(S_{0,j}, O_j)$ – обучающая выборка.

Параметры: шаг градиентного спуска τ , размер пакета N_{batch} , точность алгоритма ϵ .

Выход: критическая точка p_c : $|\nabla_p \sum_j L(S_{n,j}, O_j)| < \epsilon$.

- 1 **Пока** $|\nabla_p \sum_j L(S_{n,j}, O_j)| \geq \epsilon$:
- 2 Собрать подмножество обучающей выборки размером N_{batch} – «пакет» (S_{n,j_b}, O_{j_b}) .
- 3 Посчитать параметрический градиент $\nabla(b) = \nabla_p \sum_{j_b} L(S_{n,j_b}, O_{j_b})$ по элементам пакета.
- 4 **Если** p должно лежать на многообразии P :
- 5 Спроецировать градиент на касательную плоскость многообразия:
 $\delta = P \downarrow (\nabla(b))$.
- 6 Вычесть δ из текущей точки:
 $p_t \leftarrow p - \delta$.
- 7 Спроецировать точку обратно на многообразии:
 $p \leftarrow P \uparrow (p_t)$.
- Иначе:**
- 4 Прибавить к текущим параметрам антиградиент:
 $p \leftarrow p - \nabla(b) \cdot \tau$.
- 5 Произвести оценку обобщенного градиента. Если выполняется условие $|\nabla_p \sum_j L(S_{n,j}, O_j)| < \epsilon$, то $p_c \leftarrow p$ и выйти из цикла.
- 6 **Выход:** вернуть p_c .

Отметим, что для краткости изложения в алгоритме 1 были опущены вопросы, связанные с введением критериев останова по сходимости и достижению максимального числа итераций. Принципиальным является шаг 4. Поскольку «параметрами» программы по оптимизации составной аналоговой ИС являются характеристические функции схемотехнических блоков, таких как устройства «выборки-хранения» (УВХ), компараторов, цифро-аналоговых преобразователей (ЦАП), то эти функции принадлежат к тем или иным многообразиям по построению. Определение этих многообразий в ручном режиме трудоемко, поэтому для этих целей могут быть успешно применены методы машинного обучения. Но они также требуют определенной адаптации (подробнее – раздел 3).

2.2. Дифференцируемая программа для синтеза АЦП-ПП

АЦП-ПП и принцип его работы изображен на рис. 1.

В момент команды «Начать преобразование» АЦП-ПП УВХ переводится в режим хранения, и регистр последовательного приближения начинает формировать последовательно биты кода, начиная с самого старшего. Для этого все биты кода выставляются в 0, кроме старшего (на последующих стадиях – текущего). Код подается на ЦАП, откуда подается на компаратор. Компаратор сравнивает сигнал с УВХ и ЦАП и выдает логическую единицу (чаще всего ей служит напряжение питания – шина Vdd) в случае, если сигнал с УВХ больше сигнала с ЦАП, и логической 0 (чаще всего – шина Vss) в противном случае. Сигнал с компаратора преобразуется в бит кода и регистр переходит к выставлению следующего бита. По завершению преобразования формируется сигнал «EOC DRDY», пока идет преобразование формируется сигнал «BUSY».

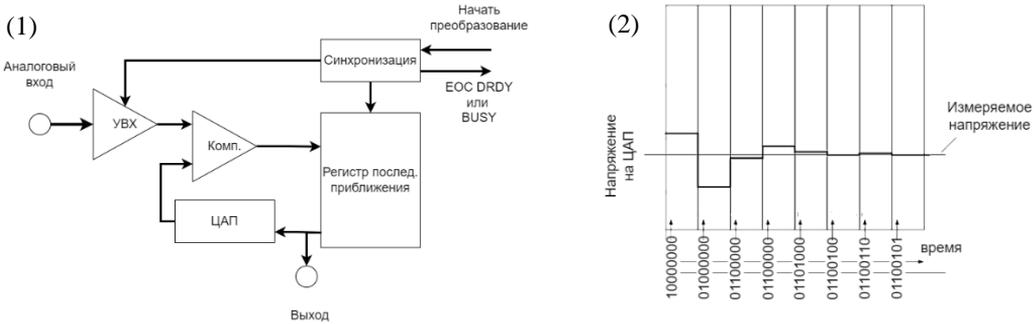


Рис. 1. Структурная схема АЦП-ПП (1) и преобразование сигнала в ней (2).
Fig. 1. SAR structural scheme (1) and signal forming in SAR (2).

Дифференцируемая программа преобразования «аналог – цифра» для подобного АЦП-ПП будет иметь следующий вид (алгоритм 2).

Характеристические функции (ХФ), упомянутые в алгоритме 2 – функции, описывающие поведение схематехнических блоков во времени в зависимости от параметров. Идеальные характеристические функции, их свойства и источники их неидеальности приведены в табл. 1.

В случае, если все ХФ близки или равны идеальным, АЦП-ПП работает с бесконечной скоростью и точностью. Очевидно, что такие функции недостижимы для реальных схематехнических решений УВХ, компаратора и ЦАП. Возможные ХФ блоков определяются схематехникой блоков и их параметрами, формирование ХФ с помощью методов МО описано в разделе 3. Общая программа для синтеза схематехнических блоков представлена в алгоритме 3.

3. Методы машинного обучения для аппроксимаций многообразий схематехнических блоков

Преобразование между пространством характеристических функций и пространством схематехнических решений наиболее естественно осуществлять с помощью методов МО. Для этого могут быть использованы нейронные сети с архитектурами автокодировщик или трансформер, в данной работе был использован вариационный автокодировщик [10]. Общий вид такой ИНС представлен на рис. 3.

Операторы проекции на касательную плоскость многообразия и репроекции на многообразии реализуются при использовании нейросетевых архитектур типа автокодировщик

естественным образом (рис. 2) по следующему алгоритму. ХФ, к которой прибавлен градиент (шаг (1) рис. 2) преобразуется в латентном пространстве автокодировщика (ВАК-ЛП (от вариационный автокодировщик-латентное пространство), шаг (2) рис. 2) в код существенной меньшей размерности, который может рассматриваться как касательное пространство к многообразию, описывающему p , затем оно декодируется в ХФ из того же многообразия. Для проведения шага 8 алгоритма 2 в архитектуре автокодировщика встроена «голова», которая осуществляет преобразование из латентного пространства, кодирующего ХФ, в СЭП соответствующего схемотехнического блока. Следует отметить, что надежность преобразования зависит, как минимум, от размера выборки и может быть недостаточной для малых объемов выборки. Поэтому для проверки надежности были проведены тесты, которые измеряют качество в зависимости от размера выборки. Их результаты приведены в табл. 2.

Было подтверждено, что при больших размерах синтезированной выборки, что схема на рис. 2 работает с достаточной надежностью. Однако, такие размеры выборки могут быть непрактичными в случае, когда надо создавать множество нейросетевых моделей схемотехнических блоков. Поэтому схема преобразования была модифицирована следующим образом (рис. 3).

Алгоритм 2. SarAdcDiff. Дифференцируемая программа оцифровки аналогового сигнала в АЦП-ПП.

Algorithm 2. SarAdcDiff. A differentiable program for analog to digital conversion.

Вход: аналоговый сигнал $A(t)$.

Выход: последовательность бит преобразования аналог-цифра. *Примечание: поскольку программа дифференцируемая – формируется последовательность вещественных чисел, близких к 0 или 1, а не строго равных им.*

Оптимизируемые в ДП параметры: «характеристическая функция» УВХ – $sh(V,t)$, «характеристическая функция» компаратора $comp(t)$, «характеристическая функция» ЦАП – $dac(C,t)$.

Константные параметры: число бит (N_{bit}), время оцифровки (T_{adc}), время шага преобразования $\Delta t = \frac{T_{adc}}{N}$, κ_0, χ_0 – параметры, характеризующие функцию бинаризации.

Изменяемые параметры алгоритма: текущий битовый код C .

- 1 Перевести УВХ в режим хранения, захватить текущий входной сигнал
 $s(t) = sh(A(t_0), t)$.
- 2 Выставить все биты кода в 0:
 $C \leftarrow 0_0 \dots 0_{N_{bit}-1}$
- 3 Для i меняющегося от $N_{bit} - 1$ до 0:
- 4 Выставить i бит в 1:
 $C \leftarrow 0_0 \dots 0_{i-1} 1_i \dots C_{N_{bit}-1}$.
- 5 Подать во время равное $t_i = t_0 + i \cdot \Delta t$ на ЦАП код следующего вида:
 $C_{out} = \sum_i 2^i \cdot C_i$ и сформировать сигнал ЦАП $dac(C_{out}, 0)$.
- 6 Подождать половину цикла и считать значение сигнала с компаратора:
 $comp(t) = comp(sh(t), dac(C_{out}, \Delta t/2))$.
- 7 Сохранить i -тый бит в компараторе как:
 $C = 0_0 \dots 0_{i-1} \Phi\left(comp\left(t_i + \frac{\Delta t}{2}\right)\right)_i \dots C_{N_{bit}-1}$
 $\Phi(\chi) = \frac{\exp((\chi - \chi_0) \cdot \kappa_0)}{\exp((\chi - \chi_0) \cdot \kappa_0) + 1}$ - гладкая бинаризирующая функция.

8 **Выход:** вернуть C

Табл. 1. Свойства ХФ.

Table.1. Properties of characteristic functions (CF).

СХТ блок	Идеальная характеристическая функция	Источники ошибок
УВХ	$sh(A,t)=A$	Недостаточность характеристик ОУ в составе УВХ, утечка заряда на конденсаторе хранения через паразитные элементы, инжекция паразитных зарядов в конденсатор хранения.
Компаратор	$comp(A, B, t) = \begin{cases} A < B, V_{ss} \\ A > B, V_{dd} \end{cases}$	Недостаточность чувствительности, большое время выставления финального значения, зависимость от величины постоянного смещения.
ЦАП	$dac(C, t) = C \cdot K$	Нелинейность преобразования, деградация со временем, недостаточность внутреннего сопротивления, невозможность представить полный диапазон значений.

Алгоритм 3. Программа схемотехнического синтеза АЦП-ПП.

Algorithm 3. SAR schematic synthesis program.

Вход: последовательность аналоговых сигналов А и эталонных битовых кодов, начальное приближение для схемотехники ИС, параметр малости невязки ϵ .

Выход: схемотехника ИС в виде СЭП.

Оптимизируемые в ДП параметры: «характеристическая функция» УВХ – $sh(V,t)$, «характеристическая функция» компаратора – $comp(t)$, «характеристическая функция» ЦАП – $dac(C,t)$.

- 1 Выполнить преобразование из имеющихся схем блоков в ХФ.
- 2 **Пока** не достигнуты параметры, удовлетворяющие спецификации:
- 3 Для каждого элемента обучающей выборки (или обучающего пакета) сформировать выходные коды с помощью алгоритма SarAdcDiff:

$$S_{n,j} = \text{SarAdcDiff}(A_j).$$
- 4 Сформировать набор невязок:

$$L = \sum_{k,j} |S_{n,j,k} - O_{j,k}| \cdot 2^k,$$
 к – номер бита.
- 5 Посчитать параметрический градиент.
- 6 Обновить ХФ.
- 7 **Если** $L < \epsilon$, преобразовать ХФ в СЭП и запустить средство схемотехнического моделирования для проверки.
- 8 **Если** схемотехническое моделирование дало удовлетворяющий спецификации результат:
- 9 **Выход**
- 10 **Иначе:**
- 11 получить коэффициенты пропорциональности между реальными и расчетным ХФ и использовать их при будущем моделировании
- 12 **Выход: вернуть С**

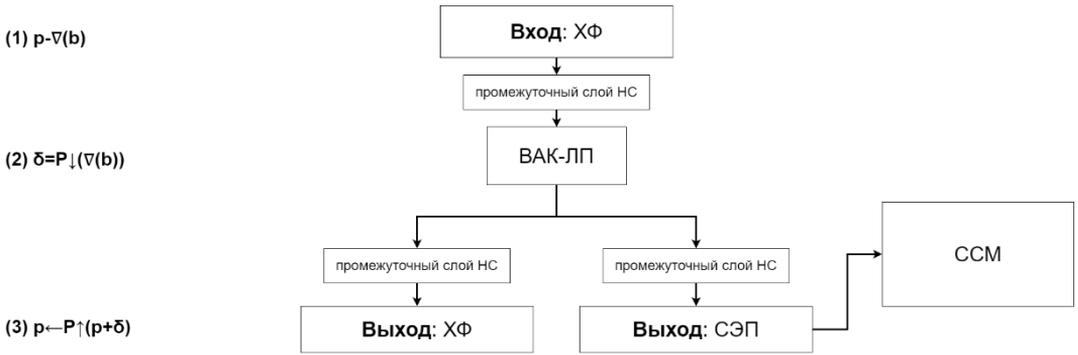


Рис. 2. Автокодировщик для представления многообразия ХФ и преобразования ХФ в СЭП.
 Fig. 2. Autoencoder for learning CF manifolds and translation CF to netlist.

Табл. 2. Зависимость ошибки предсказания ХФ от размера синтетической выборки.
 Table 2. The set of specifications for testing.

Размер выборки, кол-во элементов	Относительная ошибка предсказания ХФ с помощью автокодировщика
500	0,42
5000	0,13
50000	0,04

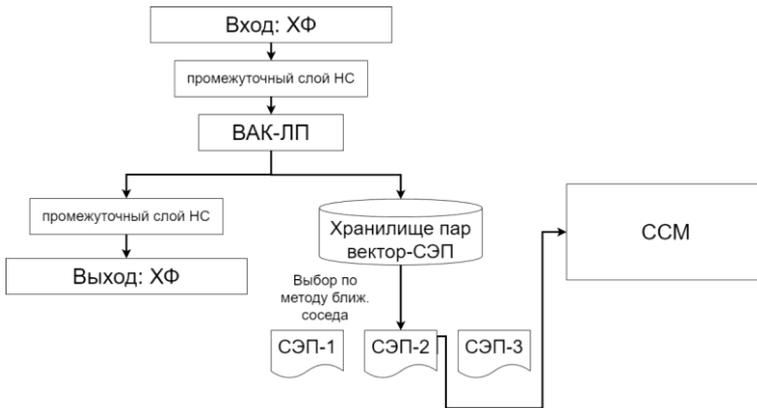


Рис. 3. Модификация автокодировщика с преобразованием ХФ-СЭП по методу ближайшего соседа.
 Fig.3 Autoencoder modification with use of CF-netlist transform by nearest neighbor search method.

Каждому СЭП соответствует вектор в латентном пространстве автокодировщика, который соответствует его ХФ. Для всех ХФ и соответствующих им СЭП из обучающей выборки формируется хранилище векторов. Преобразование ХФ в СЭП осуществляется с помощью поиска наилучшего соответствия (вектора с наименьшим расстоянием) из хранилища векторов, в качестве СЭП берется СЭП, соответствующая этому вектору из хранилища. Хранилище представляет собой реализованное авторами на основе открытых исходных кодов дерево поиска ближайших соседей [11], ускоряющее поиск ближайшего элемента. Преобразование «схема-СЭП» ограничено только теми примерами, которые уже встречались в обучающей выборке, что делает невозможным галлюцинации сети и позволяет работать даже для небольших тестовых выборок. В случае, если производство располагает большим

количеством тестовых примеров, использование схемы, изображенной на рис. 2 оправдано в большей мере, так как не ограничивает решения только ближайшими образцами.

4. Результаты экспериментов по синтезу ОУ и АЦП-ПП

Для экспериментов был собран тестовый стенд (маршрут), основанный на САПР с открытым исходным кодом, включающий ССМ Хусе [12], средство топологического размещения Align [6]. Общий маршрута представлен на рис. 4.

В качестве набора средств проектирования (Process Development Kit – PDK) использовалось PDK Google+SkyWater-130 nm [13]. В качестве условий спецификации для АЦП-ПП задавалось требуемое количество бит, эффективное соотношение сигнал-шум и частота дискретизации (табл. 3). В качестве «строительных блоков» алгоритм синтезировал из вариантов, представленных в табл. 4.

Авторами было проведено сравнение с методами обучения с подкреплением для подтверждения эффективности предложенного подхода. В качестве референсного алгоритма был выбран алгоритм [2] из-за наличия исходного кода, его простоты и заявляемой эффективности. Перед выполнением синтеза по спецификации алгоритм был пред-обучен на специальном выборе из троек – (набор (УВХ, компаратор, ЦАП) СЭП блоков и их ХФ, изменение одного элемента из данного набора, новые СЭП и ХФ). Время генерации было ограничено одними сутками (1440 минут). Состав и параметры синтезированных ИС представлен в табл. 5.



Рис. 4. Маршрут для автоматического проектирования.
Fig. 4. Proposed automatic synthesis route.

Табл. 3. Набор спецификаций для тестирования.
Table 3. The set of specifications for testing.

Спецификация	Количество бит	Соотношение сигнал шум, дБ	Частота дискретизации, МГц
АЦП-8	8	45	100
АЦП-12	12	52	25
АЦП-16	16	63	0,5

Табл. 4. Набор схемотехнических шаблонов для синтеза ИС.
Table 4. Schematics templates for IC synthesis.

Реализация компаратора	Реализация УВХ	Реализация ЦАП
Компаратор на регенеративной защелке	УВХ с одним ОУ и конденсатором с контролем заряда: <ol style="list-style-type: none"> 1. Двухкаскадный простой ОУ 2. ОУ с диодной нагрузкой (двухкаскадный) 3. ОУ на изогнутом каскаде (двухкаскадный) 	ЦАП R-2R типа с различными видами ОУ: <ol style="list-style-type: none"> 1. Двухкаскадный простой ОУ 2. ОУ с диодной нагрузкой (двухкаскадный) 3. ОУ на изогнутом каскаде (двухкаскадный)
StrongARM компаратор		

По результатам экспериментов видно (табл. 5), что предложенный подход позволяет синтезировать ИС существенно быстрее, чем подходы, основанные на обучении с подкреплением, и сравнимо по времени с подготовительной стадией проектирования экспертом (запустить САПР и выпить кофе со сменой эмоций). Также интересно, что начального УВХ (правда, с различными размерами транзисторов) «хватило» для всех трех случаев. Это связано с тем, что моделирование проводилось для одной транзисторной модели (параметры t_t – «typical typical») и при моделировании «в углах» результаты получились бы другими. Также стоит отметить более разнообразный выбор вариантов схем, полученный алгоритмом обучения с подкреплением, что имеет как свои плюсы (исследование более широкой области параметрического пространства), так и минусы (неоправданно переусложненный дизайн). Неудача с синтезом АЦП-12 имеет, по всей видимости, вероятностный характер, связанный со случайным выбором точки, являющейся локальным минимумом, из которой алгоритм не смог выбраться. При многократном повторении этот артефакт был бы исключен. Поскольку этот эксперимент не влиял на «общую картину» и был длительным повторен он не был.

Для СЭП, сгенерированных алгоритмом, запускался топологический раскладчик Align, который позволял генерировать аналоговую часть АЦП непосредственно в GDS. Результаты представлены на рис. 5.

Время, требующееся для получения топологических решений – 40 минут для АЦП-8, 56 минут для АЦП-12 и 66 минут для АЦП-16, что также существенно быстрее ручного проектирования.

Все результаты топологического проектирования свободны от DRC/LvS нарушений, но компактность топологического проектирования, по мнению авторов, недостаточна несмотря на то, что для функций потерь, с учетом распознанных условий на компактность и симметричность, были достигнуты локальные минимумы. На текущем этапе проектирование такого качества может использоваться топологом в качестве начального приближения, либо в контуре автоматического подбора параметров СЭП – сайзинге. Во втором случае, должно быть еще больше сокращено время проектирования, в первом случае – требуется повышение качество проектирования.

Табл. 5. Результаты соответствия спецификации автоматически синтезированных ИС.
 Table 5. Parameters versus schematics of syntheses schemes.

Спецификация	Способ синтеза	Состав	Время синтеза, минуты	Соотношение сигнал-шум (требования спецификации), дБ	Частота, МГц
АЦП-8	ДП	1. УВХ с простым двухкаскадным ОУ 2. Компаратор на регенеративной защелке 3. ЦАП с простым двухкаскадным ОУ	14	46,1 (45)	100
	RL	1. УВХ с простым двухкаскадным ОУ 2. Компаратор StrongArm 3. ЦАП с простым двухкаскадным ОУ	108	47,6 (45)	100
АЦП-12	ДП	1. УВХ с простым двухкаскадным ОУ 2. Компаратор StrongArm 3. ЦАП с ОУ с диодной нагрузкой	24	52,4 (52)	25
	RL	1. УВХ с ОУ на изогнутом каскаде 2. Компаратор StrongArm 3. ЦАП с ОУ на изогнутом каскаде	<i>1440 (не смог сойтись)</i>	39,8 (52)	25
АЦП-16	ДП	1. УВХ с простым двухкаскадным ОУ 2. Компаратор StrongArm 3. ЦАП с ОУ на изогнутом каскаде	40	63,2 (63)	0,5
	RL	1. УВХ с ОУ с диодной нагрузкой 2. Компаратор StrongArm 3. ЦАП с ОУ на изогнутом каскаде	237	67,4 (63)	0,5
ДП – дифференцируемое программирование RL – обучение с подкреплением					

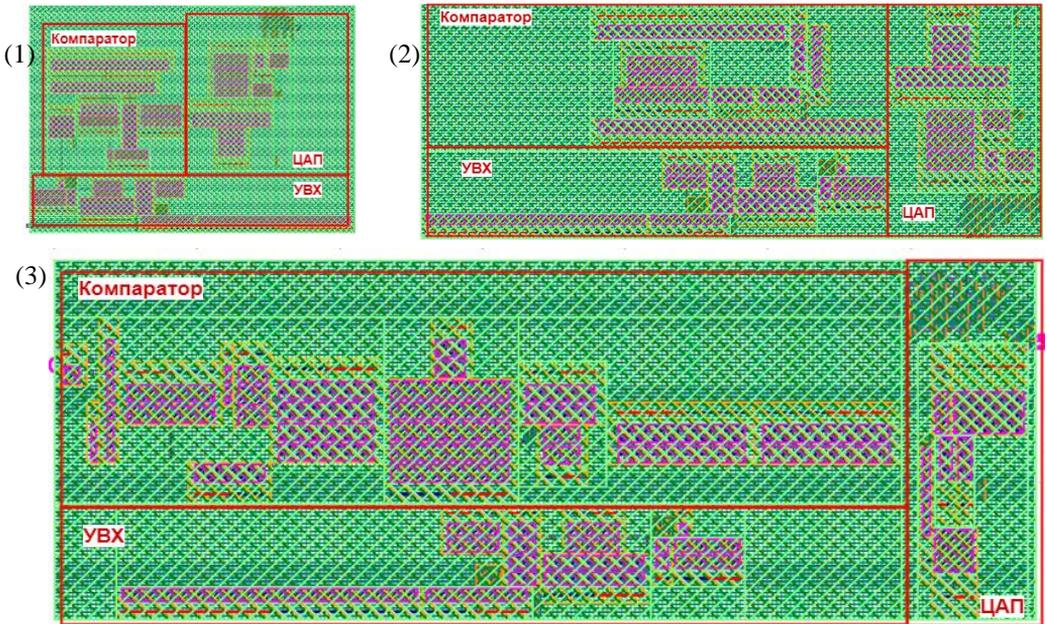


Рис. 5. Результаты топологического проектирования для синтезированных аналоговых СЭП для АЦП-8 (1) АЦП-12 (2) АЦП-16 (3).

Fig. 5. Layout for synthesized analog netlist for ADC-8 (1), ADC-12 (2), ADC-16 (3).

5. Заключение

В данной работе были исследованы возможности применения методов машинного обучения и дифференцируемого программирования к задачам синтеза аналоговых ИС и аналоговых частей аналогово-цифровых ИС. Были обоснованы причины, по которым этот подход может иметь преимущество над моделями типа RL и моделями, основанными на чистых методах машинного обучения, обученных на больших выборках. Показано, что данный подход существенно быстрее методов, основанных на обучении с подкреплением. «Под ковер» была спрятана длительная и дорогостоящая в плане вычислительных ресурсов фаза обучения ХФ схмотехнических шаблонов, но она производится однократно и не влияет на время синтеза ИС проектировщиком.

Поскольку данная работа преследовала цель проверить применимость гипотезы эффективности методов дифференцируемого программирования для синтеза ИС схмотехническое моделирование проводилось для типичного значения параметров (точка t) и не проходило верификационную часть и часть выделения паразитных элементов, и эти части еще не готовы в маршруте, разрабатываемом группой авторов исследования (рис. 4).

Было показано, что предложенные методы позволяют получить схмотехнику аналоговых ИС, удовлетворяющую спецификации, значительно быстрее, чем это получилось бы у «живого» схмотехника и алгоритма обучения с подкреплением, что делает предложенный метод применимым для решения стандартных схмотехнических задач и дает возможность эксперту сосредоточиться на сложных и требующих творческого подхода случаях.

Также было показано, что можно в течении минут получить не только схмотехническое решение, но и результат топологического проектирования, свободный от нарушений DRC/LVS, но уступающий решению, спроектированному экспертом, и содержащий ряд дефектов (с точки зрения эксперта-тополога), которые не покрываются правилами DRC /LVS.

В рамках развития данной темы можно предложить следующие направления исследования:

- одновременная оптимизация и синтез схемотехнического и топологического решения;
- поддержка моделирования в углах, а не только центральной точки (tt);
- включение в контур оптимизации методов оценки паразитных элементов и разработка решения с учетом реально получающихся условий, которые будут развиты в рамках будущих работ группы.

Список литературы / References

- [1]. Hastings A. The art of analog layout. – Prentice hall New Jersey, 2001, 576 p.
- [2]. Settaluri, K., Haj-Ali, A., Huang, Q., Hakhmaneshi, K., & Nikolic, B. (2020, March). Autockt: Deep reinforcement learning of analog circuit designs. In 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 490-495). IEEE.
- [3]. Budak, A. F., Jiang, Z., Zhu, K., Mirhoseini, A., Goldie, A., & Pan, D. Z. (2022, January). Reinforcement learning for electronic design automation: Case studies and perspectives. In 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC) (pp. 500-505). IEEE.
- [4]. Khan, S., Shi, Z., Li, M., & Xu, Q. (2024, March). DeepSeq: Deep Sequential Circuit Learning. In 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 1-2). IEEE.
- [5]. Xu, B., Zhu, K., Liu, M., Lin, Y., Li, S., Tang, X., ... & Pan, D. Z. (2019, November). MAGICAL: Toward fully automated analog IC layout leveraging human and machine intelligence. In 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 1-8). IEEE.
- [6]. Kunal, K., Madhusudan, M., Sharma, A. K., Xu, W., Burns, S. M., Harjani, R., ... & Sapatnekar, S. S. (2019, June). ALIGN: Open-source analog layout automation from the ground up. In Proceedings of the 56th Annual Design Automation Conference 2019 (pp. 1-4).
- [7]. Abel, I., Neuner, M., & Graeb, H. E. (2021). A hierarchical performance equation library for basic op-amp design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(7), 1976-1989.
- [8]. Blondel, M., & Roulet, V. (2024). The elements of differentiable programming. arXiv preprint arXiv:2403.14606.
- [9]. Baydin A. G. et al. Automatic differentiation in machine learning: a survey // *Journal of machine learning research*. – 2018. – Т. 18. – №. 153. – С. 1-43.
- [10]. Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.
- [11]. Malkov, Y. A., & Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4), 824-836.
- [12]. Dodge, H. D. (2020). Xyce Case Study (No. SAND-2020-2877R). Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).
- [13]. Rahman, M. S., Suvo, S. H., Kamal, Z., & Rahman, T. (2024, May). Standard Cell Library Characterization Flow Using Open Source Tools with Google+ SkyWater 130nm PDK. In 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) (pp. 1175-1180). IEEE.

Информация об авторах / Information about authors

Денис Сергеевич ЩЕПЕТОВ – кандидат технических наук, ведущий специалист ФИЦ ИУ РАН. Сфера научных интересов: многокритериальная оптимизация, методы машинного обучения, робототехнические системы.

Denis Sergeevich SHCHPETOV – Cand. Sci. (Tech.), Leading Specialist of the FRC ICS RAS. Research interests: multicriteria optimization, machine learning methods, robotic systems.

Александр Геннадиевич ТИМОШЕНКО – кандидат технических наук, доцент НИУ МИЭТ. Сфера научных интересов: проектирование аналоговых ИС и схем смешанного сигнала, обработка сигналов, схемотехническое проектирование.

Aleksandr Gennadyevich TIMOSHENKO – Cand. Sci. (Tech.), associate professor of the National Research University of Electronic Technology. Research interests: analog and mixed signals IC design, signal processing, circuit design.

Владимир Анатольевич ГАРАНЖА – доктор физико-математических наук, профессор РАН, главный научный сотрудник ФИЦ ИУ РАН. Сфера научных интересов: дифференциальная геометрия, многокритериальная оптимизация, параллельное программирование, построение расчетных сеток.

Vladimir Anatolyevich GARANZHA – Dr. Sci. (Phys.-Math.), Professor of the RAS, Chief Researcher FRC ICS RAS. Research interests: differential geometry, multicriterial optimization, parallel programming, construction of computational grids.

Игорь Евгеньевич КАПОРИН – доктор физико-математических наук, главный научный сотрудник ФИЦ ИУ РАН. Сфера научных интересов: итеративные алгоритмы решения СЛАУ, безградиентные методы оптимизации больших задач, параллельное программирование.

Igor Evgenievich KAPORIN – Dr. Sci. (Phys.-Math.), Chief Researcher FRC ICS RAS. Research interests: iterative algorithms for solving SLAE, gradient-free methods for optimizing large problems, parallel programming.

Дмитрий Николаевич КАРГИН – аспирант, Московский политехнический университет. Сфера научных интересов: параллельное программирование, робототехнические системы, многокритериальная оптимизация.

Dmitry Nikolaevich KARGIN – postgraduate student, Moscow Polytechnic University. Research interests: parallel programming, robotic systems, multicriterial optimization.

