



УДК 519.714.5
DOI: 10.37661/1816-0301-2025-22-1-27-39

Оригинальная статья
Original Article

Перепроектирование КМОП СБИС средствами инструмента синтеза Yosys

Д. И. Черемисинов, Л. Д. Черемисинова[✉]

*Объединенный институт проблем информатики
Национальной академии наук Беларуси,
ул. Сурганова, 6, Минск, 220012, Беларусь
✉E-mail: cld@newman.bas-net.by*

Аннотация

Цели. Рассматривается задача перепроектирования схемы транзисторного уровня, заданной в формате SPICE, в другом технологическом базисе. Целью статьи является разработка подхода к перепроектированию схем на основе использования средств программных пакетов автоматизации проектирования с открытым исходным кодом.

Методы. Предлагается метод, в основе которого лежат экстракция структуры на уровне логических элементов из плоского SPICE-описания транзисторной схемы и экспорт полученного иерархического SPICE-описания в программную среду открытого пакета синтеза Yosys. Целью экспорта являются преобразование описания логической сети в формате SPICE в описания на входных языках систем автоматизации проектирования, а также выполнение операций оптимизации и синтеза в среде Yosys.

Результаты. Для экспорта в ядро пакета Yosys логической сети, заданной в формате SPICE, была разработана программа на языке C++ с использованием классов пакета Yosys. Программа принимает и обрабатывает иерархическое SPICE-описание логической сети, переводя его в представление во внутреннем формате инструмента Yosys.

Заключение. Разработанная программа оформлена в виде программного модуля Yosys и интегрирована в его среду в качестве одной из команд. Над полученной модулем структурой логической сети могут быть выполнены все доступные в Yosys преобразования.

Ключевые слова: перепроектирование, декомпиляция транзисторных схем, КМОП-схемы, формат SPICE, язык Verilog, пакет Yosys

Для цитирования. Черемисинов, Д. И. Перепроектирование КМОП СБИС средствами инструмента синтеза Yosys / Д. И. Черемисинов, Л. Д. Черемисинова // Информатика. – 2025. – Т. 22, № 1. – С. 27–39. – DOI: 10.37661/1816-0301-2025-22-1-27-39.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 04.02.2025
Подписана в печать | Accepted 18.02.2025
Опубликована | Published 31.03.2025

Redesigning CMOS VLSI using Yosys synthesis tool

Dmitry I. Cheremisinov, Ljudmila D. Cheremisinova[✉]

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus*
[✉]E-mail: *cld@newman.bas-net.by*

Abstract

Objectives. The problem of reverse engineering of a transistor level circuit specified in the SPICE format in a different technological basis is considered. The goal of the work is to develop an approach to redesigning circuits using open source design automation software packages.

Methods. A method is proposed based on extracting the structure at the level of logical elements from a flat SPICE description of a transistor circuit and exporting the resulting hierarchical SPICE description to the software environment of the open synthesis package Yosys. The purpose of the export is to transform the description of the logical network in the SPICE format into descriptions in the input languages of design automation systems, as well as to perform optimization and synthesis operations in the Yosys environment.

Results. To export a logical network specified in the SPICE format to the core of the Yosys package, a program in C++ was developed using the classes of the Yosys package. The program accepts and processes the hierarchical SPICE description of the logical network, translating it into a representation in the internal format of the Yosys tool.

Conclusion. The developed program is designed as a Yosys program module and integrated into its environment as one of its commands. All the transformations available in Yosys can be performed on the logical network structure obtained by the module.

Keywords: reverse engineering, decompilation of transistor circuits, CMOS circuits, SPICE format, language Verilog, package Yosys

For citation. Cheremisinov D. I., Cheremisinova L. D. *Redesigning CMOS VLSI using Yosys synthesis tool*. Informatika [Informatics], 2025, vol. 22, no. 1, pp. 27–39 (In Russ.). DOI: 10.37661/1816-0301-2025-22-1-27-39.

Conflict of interest. The authors declare of no conflict of interest.

Введение. Современные сверхбольшие интегральные схемы (СБИС) содержат до миллиарда примитивных элементов на уровне транзисторов, и их сложность быстро увеличивается. Эффективным инструментом автоматизации тестирования и проектирования СБИС является обратное проектирование, или обратный инжиниринг (англ. hardware reverse engineering), схем на уровне транзисторов [1, 2]. Задача обратной инженерии инверсна задаче проектирования СБИС в смысле направления процесса преобразований. Она заключается в построении спецификации устройства путем анализа его аппаратной реализации в виде СБИС. Основным этапом обратного проектирования является декомпиляция плоской транзисторной схемы, которая состоит в извлечении описания уровня логических элементов [3, 4]. Полученное логическое описание устройства позволяет перепроектировать реализующую его транзисторную схему [2]. Перепроектирование, в отличие от оригинального проектирования, предполагает проектирование устройства на новой элементной основе для замены его существующей схемной реализации.

Процесс проектирования электронных устройств в любой системе автоматизированного проектирования (САПР) разбивается на ряд этапов преобразований от высокого уровня абстракции к низкому. Описание верхнего функционального уровня состоит из функциональных блоков, таких как триггеры, сумматоры и т. д., каждый из которых может содержать большое количество логических элементов. Логический уровень использует логические вентили как стандартные блоки, чтобы описать схемы. На транзисторном уровне схемы описываются в терминах транзисторов и их взаимосвязей.

Каждый этап проектирования требует своих специализированных инструментов. Программное обеспечение для автоматизации проектирования электронных устройств предлагают три основных мировых поставщика: Synopsys, Cadence Design Systems и Siemens EDA (ранее

Mentor Graphics, которая в 2017 г. перешла компании Siemens). Эти поставщики предлагают программные пакеты, позволяющие охватить весь спектр этапов проектирования интегральных схем: от синтеза по функционально-логическому описанию аппаратуры на языке HDL (Hardware Description Language – язык описания оборудования) до физического синтеза. Инструментальные средства САПР различаются набором не только предлагаемых проектных операций, элементных базисов и стилей проектирования, но и входных языков описания аппаратуры. Как правило, в качестве входных языков современных САПР используются языки VHDL и Verilog. Это требует наличия средств конвертации форматов представления разрабатываемых схем.

Разработка инструментов проектирования тесно связана с технологией изготовления микросхем. Свойства технологии конкретного завода полупроводников, такие как модели транзисторов, физические характеристики и правила проектирования, обычно фиксируются в описаниях, которые являются собственностью изготовителя программного обеспечения. Эти информационные материалы составляют набор PDK (Process Design Kit – комплект для процесса проектирования) для проектирования и процесса изготовления микросхем. Он разрабатывается обычно совместно изготовителем микросхем и поставщиком средств автоматизации проектирования, поэтому полупроводниковые предприятия выпускают PDK, которые совместимы с конкретным пакетом САПР.

Структура PDK, так же как и исходный код и структура промежуточных файлов данных САПР, является закрытой, а используемые технологии производства интегральных схем настолько сложны, что разработка средств автоматизации проектирования и продуктов, аналогичных существующим PDK, требует больших затрат времени квалифицированных разработчиков. Выходом из ситуации может быть использование свободно распространяемых программных пакетов САПР с открытым исходным кодом FOSS (Free and Open-Source Software – свободно доступное программное обеспечение с открытым исходным кодом). Эти пакеты в настоящее время находятся в стадии быстрой разработки благодаря проектам DARPA (Defense Advanced Research Projects Agency – агентство перспективных исследовательских проектов Министерства обороны США) и Google OpenROAD (Open Rapid Object Application Development – открытая быстрая разработка объектных приложений), которые содействуют разработке средств автоматизации проектирования микросхем с открытым исходным кодом. Проект OpenROAD [5] предлагает полный набор инструментов от высокоуровневого синтеза до генерации лейаута. Набор инструментов включает программные средства Yosys (Yosys Open Synthesis Suite – открытый пакет синтеза Yosys)¹ [6] для логического синтеза и OpenLane² [7] для синтеза топологии, которые ориентированы на выпуск заказных СБИС с технологией 130 нм (SkyWater 130 nm Open Source PDK³). В последнее время пакеты FOSS для проектирования интегральных схем совершенствуются благодаря усилиям энтузиастов из академических кругов и коммерческих организаций, которые пытаются упростить доступ и свободное использование набора инструментов для цифрового и аналогового проектирования микросхем.

Yosys представляет собой первое бесплатное программное обеспечение с открытым исходным кодом и открытой внутренней структурой представления данных, которое поддерживает подавляющее большинство синтезируемых функций языка Verilog HDL. Он создан как программный пакет с расширяемой структурой, его можно пополнять новыми программными компонентами. Текущее состояние Yosys позволяет обрабатывать, оптимизировать и синтезировать проекты на Verilog-2005, а также преобразовывать Verilog-описания в форматы BLIF (Berkeley Logic Interchange Format – формат обмена логическими данными Беркли), EDIF (Electronic Design Interchange Format – формат обмена проектами электронных устройств), RTL (Register Transfer Language – язык регистровых пересылок), Verilog, BTOR, SMT-LIB и др., доступные для использования в других САПР. Некоторые этапы процесса синтеза используют

¹Yosys Open Synthesis Suite. – URL: <https://yosyshq.net/yosys/> (date of access: 10.01.2025).

²OpenLane. The Open-Source Infrastructure Platform for Silicon Development. – URL: <https://efabless.com/openlane> (date of access: 10.01.2025).

³SkyWater SKY130 PDK. – 2020. – URL: <https://skywater-pdk.readthedocs.io/en/main> (date of access: 10.01.2025).

внешние инструменты, которые интегрированы в Yosys. Например, программные средства минимизации комбинационной логики и технологического отображения выполняются системой ABC⁴ верификации и синтеза последовательностных схем, которая разработана группой Университета Беркли (Berkeley Logic Synthesis and Verification Group).

В настоящей работе рассматривается задача из области перепроектирования КМОП СБИС. Основным этапом обратного проектирования служит декомпиляция плоского нетлиста транзисторной схемы, заданной в формате SPICE (Simulation Program with Integrated Circuit Emphasis – программа моделирования с акцентом на интегральные схемы), которая заключается в извлечении описания на уровне логических элементов [3]. С помощью декомпиляции получается логическая сеть, представленная в виде иерархического SPICE-описания. Это описание может служить исходной спецификацией для выполнения оптимизации и синтеза СБИС, если представить его на используемых в САПР языках проектирования, таких как VHDL, Verilog и др. В работе предложены метод и его программная реализация для экспорта иерархического SPICE-описания логической сети в среду пакета синтеза Yosys⁵ [6] прежде всего с целью преобразования SPICE-описания в другие форматы, а также для оптимизации и синтеза логической сети в рамках этого пакета.

Декомпиляция транзисторных схем. В статье [3] описывается программа декомпиляции КМОП-схем, которая предназначена для замены представления схемы на низком (транзисторном) уровне заданием на уровне логических элементов. Исходным объектом для программы служит плоский (одноуровневый) нетлист КМОП-схемы в формате SPICE, имя головной схемы и имена цепей питания. Результатом является иерархическое SPICE-описание, в которое включены модели всех идентифицированных КМОП-элементов. В программе распознаются логические КМОП-вентили статического стиля как самого распространенного, а также элементы на основе проходной логики – передаточные вентили и схемы на их основе, такие как мультиплексоры.

В процессе декомпиляции выполняются следующие основные этапы: выделение подсхем, представляющих передаточные вентили и схемы на их основе; распознавание подсхем, представляющих КМОП-вентили, и установление реализуемых ими функций; выделение групп функционально и топологически идентичных подсхем и формирование библиотеки элементов, из которых строится генерируемая логическая сеть; выделение подсхем с обратными связями; построение логической сети.

Полученная в результате декомпиляции логическая сеть представляется в виде иерархического описания в формате SPICE, в котором на нижних уровнях иерархии задаются модели подсхем, составляющих библиотеку логических элементов. Для каждого логического вентиля в процессе декомпиляции находится реализуемая им функция в скобочном представлении, которая приводится в текстовом виде в форме комментария в SPICE-описании соответствующей модели.

Полученная логическая сеть может быть выдана также в формате системы логических уравнений на языке SF иерархических функционально-структурных описаний дискретных устройств [8]. Описания на языке SF служат входными данными для разработанных в Объединенном институте проблем информатики НАН Беларуси систем логического проектирования функциональных блоков заказных КМОП СБИС [9]. В эти системы включены также средства [10] для конвертации SF-описаний дискретных устройств на язык VHDL. Для конвертации в другие языки проектирования (например, Verilog) предлагается применять средства пакета Yosys после преобразования SPICE-описания логической сети во внутреннее представление этого инструмента.

Задание логической сети в формате SPICE. В формате SPICE компоненты схемы описываются в виде моделей. Используются два основных типа моделей: модели устройств (device models) и модели подсхем (subcircuit models). Модели устройств, по сути, представляют собой примитивные компоненты, такие как транзисторы. Модель подсхемы определяет непримитивный элемент и обычно представляет собой схему, которая задается в виде блока в SPICE-описании, представляющем структуру соединений его элементов (примитивных и непримитив-

⁴ABC: A System for Sequential Synthesis and Verification / Berkeley Logic Synthesis and Verification Group. – URL: <https://people.eecs.berkeley.edu/~alanmi/abc/> (date of access: 10.01.2025).

⁵URL: <https://yosyshq.net/yosys/>

ных). Тип модели используемого в SPICE-описании элемента задается первой буквой его имени. Имена примитивных элементов, являющихся МОП-транзисторами, должны начинаться с латинской буквы «M» (или «m»). Имена непримитивных элементов, представляемых подсхемами, начинаются с латинской буквы «X» (или «x»).

Электрическая схема в формате SPICE задается списком соединений ее элементов. В этом списке указаны связи между элементами (а точнее, между их выводами), которые осуществляются с помощью электрических цепей (nets). Цепи представляют собой связанный набор выводов элементов, на которые подается один и тот же сигнал, и каждый вывод элемента подключен ровно к одной цепи. Для всех элементов указываются связи всех их выводов путем задания имен связанных с ними цепей.

Описание транзистора начинается с его имени и содержит перечисление меток цепей, связанных с выводами стока, затвора, истока и подложки в заранее определенной последовательности. Общая форма описания связей униполярного транзистора в формате SPICE имеет вид

$$\langle \text{name} \rangle \langle \text{nd} \rangle \langle \text{ng} \rangle \langle \text{ns} \rangle \langle \text{nb} \rangle \langle \text{model-name} \rangle,$$

где name – имя транзистора в схеме (начиная с буквы «M»); nd, ng, ns и nb – идентификаторы цепей, связанных с выводами стока (drain), затвора (gate), истока (source) и подложки (substrate) соответственно; model-name – имя модели устройства (для *n*-МОП- и *p*-МОП-транзисторов это могут быть nmos и pmos).

В качестве примера в листинге 1 приведено плоское описание транзисторной схемы (рис. 1) в формате SPICE, представляющей полный одноразрядный сумматор из обзора⁶.

Листинг 1. SPICE-описание полного одноразрядного сумматора

```
.SUBCKT adder a b cin cout sum vdd gnd
M0 vdd a 1 vdd p
M1 vdd b 1 vdd p
M2 1 b 2 vdd p
M3 2 a 3 vdd p
M4 1 cin 3 vdd p
M5 3 a 4 gnd n
M6 4 b gnd gnd n
M7 3 cin 5 gnd n
M8 5 a gnd gnd n
M9 5 b gnd gnd n
M10 cout 3 gnd gnd n
M11 vdd 3 cout vdd p
M12 vdd a 6 vdd p
M13 vdd b 6 vdd p
M14 vdd cin 6 vdd p
M15 6 a 7 vdd p
M16 7 b 8 vdd p
M17 8 cin 9 vdd p
M18 6 3 9 vdd p
M19 vdd 9 sum vdd p
M20 9 a 10 gnd n
M21 10 b 11 gnd n
M22 11 cin gnd gnd n
M23 12 a gnd gnd n
M24 12 b gnd gnd n
M25 12 cin gnd gnd n
M26 9 3 12 gnd n
M27 sum 9 gnd gnd n
.ENDS
```

Общая форма описания связей непримитивного элемента, модель которого представляется подсхемой с *n* выводами, в формате SPICE имеет вид

$$\langle \text{name} \rangle \langle \text{P1} \rangle \langle \text{P2} \rangle \dots \langle \text{Pn} \rangle \langle \text{model-name} \rangle,$$

где name – имя элемента в схеме (начиная с буквы «X»); «P1», «P2»... «Pn» – идентификаторы цепей, связанных с выводами элемента; model-name – имя модели элемента.

В формате SPICE выводы элементов не делятся на входные и выходные, но это разделение существенно для представления логических схем. В программе декомпиляции принято следующее соглашение: при описании моделей КМОП-элементов, которые являются (*n*, 1)-полюсниками, последний вывод всегда отождествляется с выходным полюсом элемента и именуется как «Y». Кроме того, распознанная при декомпиляции функция, реализуемая элементом, приводится в описании его SPICE-модели в виде текстового комментария, в котором знак инверсии опущен.

⁶CMOS Binary Full Adder. A Survey of Possible Implementations. – URL: <http://web.engr.uky.edu/~elias/projects/10.pdf> (date of access: 10.01.2025).

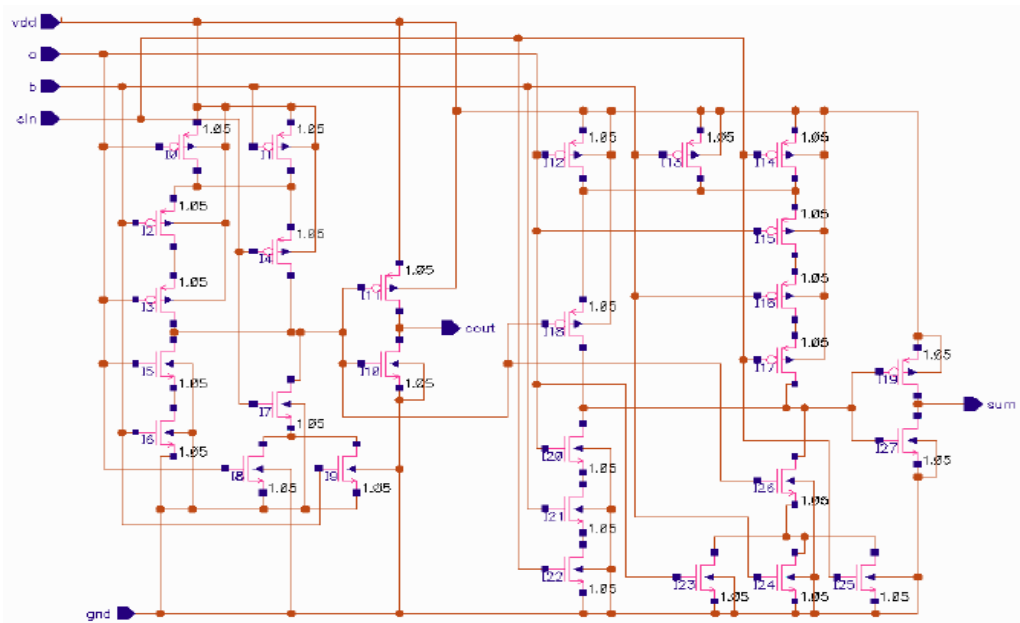


Рис. 1. Транзисторная схема полного одноразрядного сумматора
Fig. 1. Transistor circuit of a full single-bit adder

Результат декомпиляции приведенной выше транзисторной схемы одноразрядного сумматора представлен иерархическим SPICE-описанием (листинг 2), в которое включены модели трех распознанных подсхем G0, G1 и G2, представляющих КМОП-вентили. Эти модели и составляют извлеченную при декомпиляции библиотеку вентилях. Вентили XM2I1 и XM2I2 являются инверторами, они топологически изоморфны, их модель имеет имя G2. Остальные два вентиля XM0I1 и XM1I1, модели которых имеют имена G0 и G1, реализуют функции $\overline{ABC \vee G(D \vee E \vee F)}$ и $\overline{AB \vee C(D \vee E)}$.

Листинг 2. Иерархическое SPICE-описание одноразрядного сумматора, полученное после выделения логической сети

```
* SPICE deck for cell adder_gen
.GLOBAL vdd gnd
.SUBCKT G0 A B C D E F G Y
*((A AND B AND C)OR(G AND (D OR E OR F)))
M1 Y A 2 gnd n
M2 2 B 3 gnd n
M3 3 C gnd gnd n
M4 5 D gnd gnd n
M5 5 E gnd gnd n
M6 5 F gnd gnd n
M7 Y G 5 gnd n
M8 vdd A 7 vdd p
M9 vdd B 7 vdd p
M10 vdd C 7 vdd p
M11 7 A 8 vdd p
M12 8 E 9 vdd p
M13 9 F Y vdd p
M14 7 G Y vdd p
.ENDS
.SUBCKT G1 A B C D E Y
*((A AND B) OR (C AND (D OR E)))
M1 Y A 2 gnd n
M2 2 B gnd gnd n
M3 Y C 4 gnd n
M4 4 D gnd gnd n
M5 4 E gnd gnd n
M6 vdd A 6 vdd p
M7 vdd B 6 vdd p
M8 6 E 7 vdd p
M9 7 A Y vdd p
M10 6 C Y vdd p
.ENDS
.SUBCKT G2 A Y
* A
M1 Y A gnd gnd n
M2 vdd A Y vdd p
.ENDS
.SUBCKT C0 P0 P1 P2 O3 O4
XG0M0I1 P0 P1 P2 P0 P1 P2 1 2 G0
XG1M1I1 P0 P1 P2 P0 P1 P1 1 G1
XG2M2I1 1 O3 G2
XG2M2I2 2 O4 G2
.ENDS
.SUBCKT adder_gen a b cin cout sum
XC0 a b cin cout sum C0
.ENDS
```

Найденные при декомпиляции логические сети (их может быть несколько даже для одного описания [11]) выделяются как отдельные модели с именами, начинающимися с символа «С». Для отображения информации о типах портов схемы (формат SPICE не содержит средств для их указания) в списке выводов входные полюсы принято именовать, начиная с символа «Р», а выходные полюсы – с символа «О».

Для рассматриваемой схемы сумматора получилась одна логическая сеть (листинг 2, рис. 2) с именем C0, которая имеет пять выводов, три из которых (P0, P1 и P2) являются входами, два (O3 и O4) – выходами. Соответствие параметров логической сети C0 параметрам исходного SPICE-описания транзисторной схемы adder задается экземпляром XC0 этой сети в сгенерированном описании adder_gen (см. листинг 2).

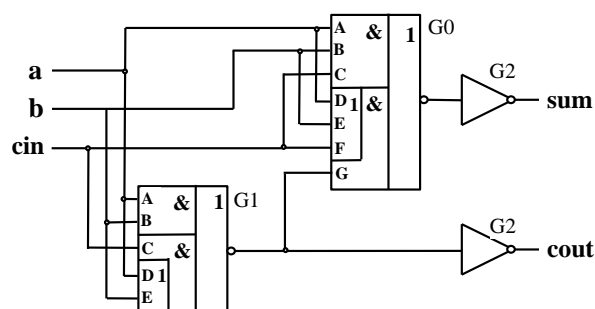


Рис. 2. Логическая сеть одноразрядного сумматора
Fig. 2. Logical network of a single-bit adder

Структура и представление данных в Yosys. Программный пакет Yosys охватывает три уровня проектирования: поведенческий, синтез на уровне регистровых передач и логический. Он принимает описание проекта на поведенческом уровне (на языке Verilog) в качестве входных данных и генерирует в качестве выходных данных описание логической сети уровня RTL логических вентилях в разных форматах. Кроме того, Yosys позволяет отображать проекты на языке Verilog в базис библиотечных ячеек заказных БИС в формате Liberty и в базис ПЛИС Xilinx седьмой серии. Для реализации логической оптимизации и отображения в указанные базы используется встроенный в Yosys программный инструмент ABC⁷.

В структурном плане Yosys представляет собой модульную программу, которая управляется с помощью скриптов. Каждая команда в скрипте относится к одному из трех классов: фронтенд (frontend), проход (pass) или бэкенд (backend). Команда типа фронтенд позволяет пользователю ввести в Yosys описание объекта проектирования, а команда типа бэкенд – получить файл с результатом. К типу проход (pass) относятся команды, которые выполняют операции над внутренним представлением объекта проектирования: анализ или преобразование (оптимизацию, синтез, перевод в другой формат). Процесс работы с новым объектом проектирования начинается с одной из команд типа фронтенд, далее команды могут выполняться в любом произвольном порядке, так как они работают с одной и той же структурой данных, которая порождена командой типа фронтенд (и может быть дополнена при выполнении команд типа pass). Команды типа бэкенд используются для записи результата преобразования описания объекта проектирования в некотором формате в выходной файл.

С функциональной точки зрения программа Yosys состоит из ядра и обрабатывающих модулей, выполнение которых запускается командами встроенного интерпретатора командной строки. Ядро Yosys представляет собой совокупность структур данных и методов, которые применяются для представления логической сети и манипуляций с ней. Обрабатывающие модули взаимодействуют через интерфейс, являющийся промежуточным объектом в формате RTLIL (Register Transfer Level Intermediate Language – промежуточный язык регистровых переключений). RTLIL-структура, представляющая проект, используется всеми проходами Yosys в качестве и входных, и выходных данных. Проходы выполняют все изменения на месте полученной на

⁷URL: <https://people.eecs.berkeley.edu/~alanmi/abc/>

входе структуры проекта, передавая на выход всю структуру целиком. Поэтому не зависящие друг от друга проходы (и команды типа backend) можно выполнять в произвольном порядке.

Представление проекта в формате RTLIL, по сути, является заданием списка соединений схемы со следующей особенностью: используется внутренняя для Yosys библиотека вентиляей, которые имеют фиксированные функции. Номенклатура вентиляей библиотеки соответствует логическим операторам языка Verilog (функции AND, OR, XOR, NOT, NAND, NOR). Однако имеется также и вентиль LUT (Lookup Table), реализуемая им булева функция задается в векторном виде. В зависимости от своей настройки вентиль LUT может реализовать любую необходимую функцию от заданного числа входных сигналов. Кроме логических вентиляей библиотека содержит также мультиплексор.

Структура данных RTLIL представляет собой класс языка C++. Корневым объектом структуры данных RTLIL является класс RTLIL::Design, который представляет собой обрабатываемую логическую сеть. В памяти Yosys всегда есть только один активный объект RTLIL::Design – текущий проект. Данные в него добавляются фронтендами (иногда и проходами), бэкенды преобразуют их в экспортируемые форматы. По умолчанию, именно текущий проект трансформируется обрабатываемыми модулями, запускаемыми командами языка-интерпретатора командной строки.

Цель последовательности трансформаций, задаваемых проходами, – преобразование вводимой логической сети в состояние, в котором ее функциональность реализуется вентилями из заданной библиотеки и проводами для связей вентиляей друг с другом. Класс RTLIL::Design является верхним слоем иерархии объектов, представляющих ядро, и может содержать несколько объектов класса RTLIL::Module (для сложных проектов), которые соответствуют модулям в Verilog. Объект RTLIL::Design служит контейнером для объектов RTLIL::Module.

Модуль RTLIL::Module, в свою очередь, включает в себя объекты трех основных типов:

- RTLIL::Cell и RTLIL::Wire, представляющие списки ячеек схемы и их связей;
- RTLIL::Process, описывающие конструкции типа if-then-else;
- RTLIL::Memory, представляющие адресуемые ячейки памяти.

При выполнении проходов синтеза объекты RTLIL::Process и RTLIL::Memory заменяются объектами RTLIL::Cell и RTLIL::Wire, которые задают списки соединений ячеек схемы.

В формате RTLIL все идентификаторы (модулей, портов, сигналов, ячеек и др.) начинаются с символа «\», если они заданы в исходном описании проекта, или с символа «\$», если они сгенерированы автоматически. При обработке объекта, заданного RTLIL, бэкендом последние идентификаторы преобразуются так, чтобы не конфликтовать с идентификаторами первого типа.

Экспорт логической сети в формате SPICE в Yosys. Yosys спроектирован как расширяемый комплекс программ. Его ядро можно дополнять командами, реализующими новые алгоритмы синтеза и обработки данных, представляемых на внутреннем языке RTLIL. Для экспорта в ядро Yosys логической сети в формате SPICE был разработан фронтенд, который принимает иерархическое SPICE-описание логической схемы и переводит его во внутреннее представление Yosys в формате RTLIL.

Фронтенд реализован программой с именем `tu_cmD` на языке C++ с использованием классов программного пакета Yosys. Разработанная программа принимает текстовое описание логической схемы в формате SPICE и генерирует соответствующий класс RTLIL::Design, состоящий из одного класса RTLIL::Module. Над полученной таким образом структурой могут быть выполнены все доступные в Yosys преобразования.

Получаемое представление схемы объектом RTLIL::Module состоит из объектов RTLIL::Cell и RTLIL::Wire, которые составляют структуру схемы со следующей особенностью, диктуемой требованиями Yosys: функциональность схемы реализуется ячейками только из заданной внутренней для Yosys библиотеки ячеек и проводами для соединения этих ячеек друг с другом.

Обработка исходного SPICE-описания разработанным фронтендом начинается с анализа его текстового представления, в процессе которого формируется внутреннее представление структуры SPICE-описания: иерархия хеш-таблиц для моделей элементов схемы (в том числе и мо-

дели самой схемы) с указанием их имен, типов, идентификаторов их входных и выходных полюсов, реализуемых функций.

В процессе анализа для модели анализируемой логической сети строится также представление, отображающее связи ее элементов, в виде помеченного двудольного орграфа $G = (V_1, V_2, E)$, $V_1 \cap V_2 = \emptyset$. В нем вершины из первой доли V_1 графа соответствуют выводам экземпляров элементов, а вершинам из V_2 ставятся в соответствие цепи и порты схемы в целом. Каждая из дуг $e \in E$ графа G связывает вершины из разных множеств V_1 и V_2 . Вершины графа из множества V_2 помечаются именами цепей и портов схемы, а вершины из V_1 имеют метки, которые идентифицируют выводы экземпляров элементов и состоят из имени экземпляра элемента и имени вывода модели этого элемента. Двудольный граф $G = (V_1, V_2, E)$, представляющий модель схемы в формате SPICE, является разреженным, и степени всех вершин в доле V_1 выводов элементов равны единице. Для задания связей такого графа логично группирование дуг, входящих и исходящих из вершин множества V_1 , которые соответствуют выводам одного и того же экземпляра элемента сети. В процессе анализа SPICE-описания для каждого экземпляра элемента сети задается список связей его выводов, приведенных в том порядке, в котором они упоминаются в описании связей модели этого элемента в формате SPICE.

Для приведенного выше SPICE-описания множества V_1 и V_2 графа G имеют 18 и 7 вершин, а связи графа задаются четырьмя списками ссылок на описания выводов элементов XG0M0I1, XG1M1I1, XG2M2I1 и XG2M2I2.

Полученная иерархия хеш-таблиц полностью отражает структуру схемы, представленной SPICE-описанием. По этому представлению генерируются списки объектов RTLIL::Cell и RTLIL::Wire, которые порождаются множествами V_1 и V_2 вершин графа $G = (V_1, V_2, E)$. При генерации объектов RTLIL::Wire (провода) для них указываются идентификаторы, ширина (равная единице в рассматриваемом случае) и направление (in/out/inout).

При генерации объектов RTLIL::Cell для каждой ячейки схемы указывается идентификатор соответствующего экземпляра элемента и его тип, идентификаторы и типы портов (входы, выходы), а также реализуемая ячейкой функция. Реализуемая функция находится по ее текстовому описанию, выбираемому из внутреннего представления SPICE-описания. Если это простая одновыходная функция (типа NAND, NOR и др.), которая реализуется одним из вентилях внутренней для Yosys библиотеки, то в RTLIL::Cell ячейка помечается типом соответствующего библиотечного элемента. В случае более сложной функции ячейка помечается типом вентиля LUT (Lookup Table), а реализуемая булева функция от n аргументов задается в векторном виде 2^n -разрядным вектором ее значений. Связи между портами ячеек и цепями задаются с помощью пар объектов RTLIL::SigSpec (или объектом типа RTLIL::SigSig).

Разработанная программа my_cmd экспорта SPICE-описания была интегрирована в среду Yosys в качестве одного из фронтов. При обращении к ней во время работы программного интерпретатора Yosys в командной строке указываются два параметра: <Par1> и <Par2>, где Par1 задает путь доступа к файлу со SPICE-описанием, а Par2 – имя модели анализируемой логической сети в этом описании.

Пример выполнения разработанного обрабатывающего модуля в среде Yosys. Работа программного модуля my_cmd в среде Yosys демонстрируется на примере импорта иерархического SPICE-описания логической сети C0 (листинг 2) из файла d:\abdata23\adder_gen_ier.sp. В Yosys это описание преобразуется с помощью команд типа write в описания на языках Verilog, BLIF и EDIF. На листинге 3 приведено состояние консольного окна программного интерпретатора Yosys при выполнении соответствующих команд.

Листинг 3. Консольное окно интерпретатора Yosys

```
yosys> my_cmd d:\2025\adder\adder_gen_ier.sp C0
Arguments to my_cmd:
  my_cmd
  d:\2025\adder\adder_gen_ier.sp
  C0
Read input file d:\2025\adder\adder_gen_ier.sp
```

```

Reading is OK
Modules in current design:
  C0 (7 wires, 4 cells)
yosys> write_verilog d:\abdata23\adder.v
1. Executing Verilog backend.
Dumping module `C0'.
yosys> write_blif d:\abdata23\adder.blif
2. Executing BLIF backend.
yosys> write_edif d:\abdata23\adder.edif
3. Executing EDIF backend.

```

На листингах 4 и 5 показаны полученные представления на языках Verilog и BLIF для импортированного в Yosys описания одноразрядного сумматора (описание в формате EDIF не приведено в силу его громоздкости). В Verilog-описании функции (вид этих функций от семи и пяти аргументов приведен в скобках в виде комментариев) первых двух ячеек заданы в виде векторов длиной 2^7 и 2^5 , две другие ячейки представляют собой инверторы. В BLIF-описании строки векторов значений аргументов функций с идентификаторами «\2» и «\1» разбиты на пять групп.

Листинг 4. Описание логической сети сумматора на языке Verilog

```

/* Generated by Yosys 0.9 (git sha1 1979e0b1, Visual Studio) */
module C0(Ca, Cb, Ccin, Ccout, Csum);
  wire \1 ;
  wire \2 ;
  input Ca;
  input Cb;
  input Ccin;
  output Ccout;
  output Csum;
  assign \2 = 128'h000055575557555755575557555755575557 >> (* src = "Y=((A * B * C) +
(G * (D + E + F)))" *) { \1 , Ccin, Cb, Ca, Ccin, Cb, Ca };
  assign \1 = 32'd2039583 >> (* src = "Y=((A * B) + (C * (D + E)))" *) { Cb, Ca, Ccin,
Cb, Ca };
  assign Ccout = ~\1 ;
  assign Csum = ~\2 ;
endmodule

```

Листинг 5. Описание логической сети сумматора в формате BLIF

```

# Generated by Yosys 0.9 (git sha1 1979e0b1, Visual Studio)
.model C0
.inputs Ca Cb Ccin
.outputs Ccout Csum
.names $false
.names $true
1
.names $undef
.names \1 Ccin Cb Ca Ccin Cb Ca \2
0000000 1      0010110 1      0101110 1      1000100 1      1011100 1
0000001 1      0011000 1      0110000 1      1000110 1      1011110 1
0000010 1      0011010 1      0110001 1      1001000 1      1100000 1
0000100 1      0011100 1      0110010 1      1001010 1      1100001 1
0000110 1      0011110 1      0110100 1      1001100 1      1100010 1
0001000 1      0100000 1      0110110 1      1001110 1      1100100 1
0001010 1      0100001 1      0111000 1      1010000 1      1100110 1
0001100 1      0100010 1      0111010 1      1010001 1      1101000 1
0001110 1      0100100 1      0111100 1      1010010 1      1101010 1

```

```
0010000 1      0100110 1      0111110 1      1010100 1      1101100 1
0010001 1      0101000 1      1000000 1      1010110 1      1101110 1
0010010 1      0101010 1      1000001 1      1011000 1
0010100 1      0101100 1      1000010 1      1011010 1
.names Cb Ca Ccin Cb Ca \1
00000 1      00011 1      01001 1      01100 1      10010 1
00001 1      00100 1      01010 1      10000 1      10011 1
00010 1      01000 1      01011 1      10001 1      10100 1
.names \1 Ccout
0 1
.names \2 Csum
0 1
.end
```

BLIF является входным форматом пакета оптимизации синтеза ABC⁸, который интегрирован в Yosys. Операции синтеза можно производить в среде Yosys для Verilog-описаний, а также в среде ABC для BLIF-описаний. Например, с помощью команд пакета ABC

```
abc 01> read_blif d:\2025\adder\adder.blif
abc 02> write_eqn d:\2025\adder\adder.eqn
```

было получено описание сумматора в виде логических уравнений (листинг 6).

Листинг 6. Описание логической сети сумматора в виде логических уравнений

```
# Equations for "C0" written by ABC on Thu Jan 30 15:59:29 2025
INORDER = Ca Cb Ccin;
OUTORDER = Ccout Csum;
new_\2_ = ((!new_\1_ + (new_\1_ * !Ccin)) * (!Ca + (!Ca * !Ccin * !Cb *
Ca))) + (new_\1_ * Ccin * !Cb * (!Ca + (!Ca * !Ccin * !Cb * Ca)));
new_\1_ = (!Cb + (Cb * !Ca)) * (!Ccin + (Ccin * !Cb * !Ca));
Ccout = !new_\1_;
Csum = !new_\2_;
```

Заключение. Разработана программа преобразования иерархического описания логической сети в формате SPICE в описание на внутреннем языке RTLIL инструмента синтеза Yosys. Программа оформлена в виде фронтенда Yosys. Расширенный пакет Yosys позволяет выполнять моделирование и перепроектирование схемы до уровня лейаута для схем, полученных путем декомпиляции СБИС на уровне транзисторов. Предлагаемая разработка полезна не только для исследовательского и академического применения, когда требуются модификации инструментов и описаний, но и для проектирования и перепроектирования СБИС промышленными САПР.

Вклад авторов. Д. И. Черемисинов разработал метод и программные средства экспорта иерархического SPICE-описания логической сети в среду пакета синтеза Yosys. Л. Д. Черемисинова разработала метод анализа иерархического SPICE-описания и подготовила текст статьи.

Список использованных источников

1. Baker, R. J. CMOS Circuit Design, Layout, and Simulation / R. J. Baker. – Third ed. – Wiley-IEEE Press, 2010. – 1214 p.
2. Hunt, V. D. Reengineering: Leveraging the Power of Integrated Product Development / V. D. Hunt. – Wiley, 1993. – 283 p.
3. Черемисинов, Д. И. Извлечение сети логических элементов из КМОП-схемы транзисторного уровня / Д. И. Черемисинов, Л. Д. Черемисинова // Микроэлектроника. – 2019. – Т. 48, № 3. – С. 224–234. – DOI: 10.1134/S0544126919030037.

⁸URL: <https://people.eecs.berkeley.edu/~alanmi/abc/>

4. Yang, L. FROSTY: A program for fast extraction of high-level structural representation from circuit description for industrial CMOS circuits / L. Yang, C.-J. R. Shi // *Integration the VLSI Journal*. – 2006. – Vol. 39, no 4. – P. 311–339.
5. Ajayi, T. Toward an open-source digital flow: First learnings from the OpenROAD project / T. Ajayi, V. A. Chhabria, M. Fogaça [et al.] // *Proc. of the 56th Annual Design Automation Conf. (DAC '19), Las Vegas, NV, USA, 02–06 June 2019*. – Las Vegas, 2019. – Article 76. – P. 1–4.
6. Wolf, C. Yosys – a free verilog synthesis suite / C. Wolf, J. Glaser, J. Kepler // *Proc. of the 21st Austrian Workshop on Microelectronics (Austrochip 2013), Linz, Austria, 10 Oct. 2013*. – Linz, 2013. – URL: <https://yosyshq.net/yosys/files/yosys-austrochip2013.pdf> (date of access: 10.01.2025).
7. Ghazy, A. A. OpenLANE: The Open-Source Digital ASIC Implementation Flow / A. A. Ghazy, M. Shalan. – URL: <https://woset-workshop.github.io/PDFs/2020/a21.pdf> (date of access: 10.01.2025).
8. Бибило, П. Н. Логическое проектирование дискретных устройств с использованием производственно-фреймовой модели представления знаний / П. Н. Бибило, В. И. Романов. – Минск : Беларус. навука, 2011. – 279 с.
9. Система логического проектирования функциональных блоков заказных КМОП СБИС с пониженным энергопотреблением / П. Н. Бибило, Н. А. Авдеев, С. Н. Кардаш [и др.] // *Микроэлектроника*. – 2018. – Т. 47, № 1. – С. 71–87.
10. Черемисинов, Д. И. Анализ и преобразование структурных описаний СБИС / Д. И. Черемисинов. – Минск : Беларус. навука, 2006. – 275 с.
11. Черемисинов, Д. И. Извлечение логических сетей при декомпиляции описаний КМОП-схем на уровне транзисторов / Д. И. Черемисинов, Л. Д. Черемисинова // *Информатика*. – 2024. – Т. 21, № 3. – С. 23–38.

References

1. Baker R. J. *CMOS Circuit Design, Layout, and Simulation*, third edition. Wiley-IEEE Press, 2010, 1214 p.
2. Hunt V. D. *Reengineering: Leveraging the Power of Integrated Product Development*. Wiley, 1993, 283 p.
3. Cheremisinov D. I., Cheremisinova L. D. *Extracting a logic gate network from a transistor-level CMOS circuit*. *Mikroelektronika [Russian Microelectronics]*, 2019, vol. 48, no. 3, pp. 224–234 (In Russ.). DOI: 10.1134/S0544126919030037.
4. Yang L., Shi C.-J. R. FROSTY: A program for fast extraction of high-level structural representation from circuit description for industrial CMOS circuits. *Integration the VLSI Journal*, 2006, vol. 39, no 4, pp. 311–339.
5. Ajayi T., Chhabria V. A., Fogaça M., Hashemi S., Hosny A., ..., Xu B. Toward an open-source digital flow: First learnings from the OpenROAD project. *Proceedings of the 56th Annual Design Automation Conference (DAC '19), Las Vegas, NV, USA, 02–06 June 2019*. Las Vegas, 2019, article 76, pp. 1–4.
6. Wolf C., Glaser J., Kepler J. Yosys – a free verilog synthesis suite. *Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip 2013), Linz, Austria, 10 October 2013*. Linz, 2013. Available at: <https://yosyshq.net/yosys/files/yosys-austrochip2013.pdf> (accessed 10.01.2025).
7. Ghazy A. A., Shalan M. *OpenLANE: The Open-Source Digital ASIC Implementation Flow*. Available at: <https://woset-workshop.github.io/PDFs/2020/a21.pdf> (accessed 10.01.2025).
8. Bibilo P. N., Romanov V. I. Logicheskoye proyektirovaniye diskretnykh ustroystv s ispol'zovaniyem produktsionno-freymovoy modeli predstavleniya znaniy. *Logical Design of Discrete Devices Using a Production-Frame Model of Knowledge Representation*. Minsk, Belaruskaja navuka, 2011, 279 p. (In Russ.).
9. Bibilo P. N., Avdeev N. A., Kardash S. N., Kirienko N. A., Lankevich Ju. Ju., ..., Cheremisinova L. D. *A system for logical design of custom CMOS VLSI functional blocks with reduced power consumption*. *Mikroelektronika [Russian Microelectronics]*, 2018, vol. 47, no. 1, pp. 65–81 (In Russ.).
10. Cheremisinov D. I. Analiz i preobrazovaniye strukturnykh opisaniy SBIS. *Analysis and Transformation of VLSI Structural Descriptions*. Minsk, Belaruskaja navuka, 2006, 275 p. (In Russ.).
11. Cheremisinov D. I., Cheremisinova L. D. *Extraction of logical networks during decompiling transistor-level CMOS circuit descriptions*. *Informatika [Informatics]*, 2024, vol. 21, no. 3, pp. 23–38 (In Russ.). DOI: 10.37661/1816-0301-2024-21-3-23-38.

Информация об авторах

Черемисинов Дмитрий Иванович, кандидат технических наук, доцент, ведущий научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.

E-mail: cher@newman.bas-net.by

Черемисинова Людмила Дмитриевна, доктор технических наук, профессор, главный научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.

E-mail: cld@newman.bas-net.by

Information about the authors

Dmitry I. Cheremisinov, Ph. D. (Eng.), Assoc. Prof., Leading Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

E-mail: cher@newman.bas-net.by

Ljudmila D. Cheremisinova, D. Sc. (Eng.), Prof., Chief Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

E-mail: cld@newman.bas-net.by