

## МЕТОДЫ И АЛГОРИТМЫ ПЕРЕПРОЕКТИРОВАНИЯ ИНТЕГРАЛЬНЫХ МИКРОСХЕМ

**Введение.** Проектирование СБИС базируется на ее иерархической декомпозиции, в основу которой положена концепция абстракции, при этом возможно разделение в двух направлениях: снизу-вверх и сверху-вниз. В зависимости от уровня абстракции, СБИС представляется на системном, регистровом, вентильном и физическом (библиотечных элементов кристалла) уровнях [1]. Переход на очередной уровень абстракции выполняется после ряда итераций (рис. 1).

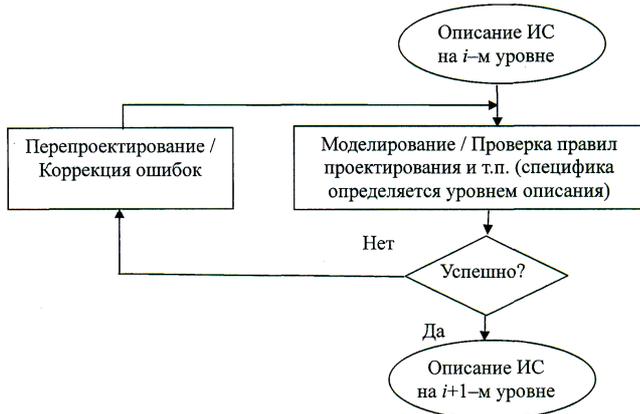


Рис. 1. Схема перепроектирования на  $i$ -м уровне абстракции

В традиционной технологии проектирования проект продвигается поэтапно от одного уровня к другому и никогда не происходит возврат

Уровни описания и  
межуровневые преобразования

Задачи перепроектирования  
одного уровня

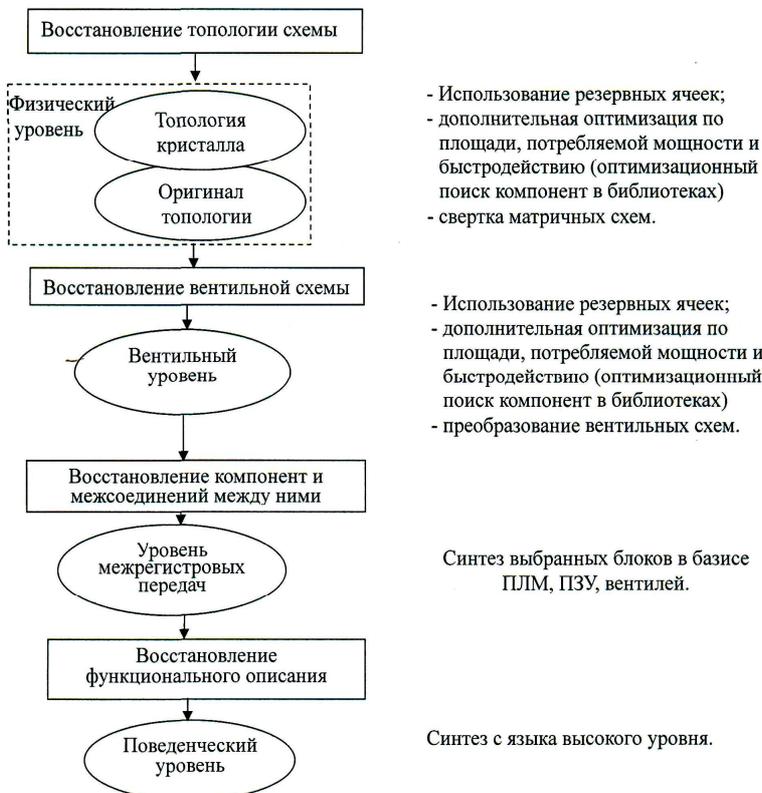


Рис. 2. Схема межуровневых переходов и задачи перепроектирования

на предыдущие уровни. Например, проектировщик уровня регистровых передач (RTL) не может прийти к системному разработчику и сказать, что его алгоритм нереализуем, или не может попросить изменить RTL-код, чтобы добиться необходимых временных параметров.

Недостаток этой методологии проектирования в том, что с увеличением сложности проекта увеличивается опасность появления ошибок и затрудняется процесс их поиска. Ошибки, обнаруженные в конце той или иной стадии проектирования, ведут к повторному ее выполнению, что в ряде случаев влечет за собой неоднократный выпуск прототипов ИС, приводит к значительному замедлению сроков выполнения проекта и резко повышает его стоимость. Более того, для схем, изготавливаемых по субмикронной технологии, такой маршрут вообще не будет работать, поскольку особенности физической реализации должны учитываться уже на логическом уровне проектирования. Поэтому в последнее время стала пользоваться популярностью методология Specify-Explore-Refine («описал – опробовал – доделал»). После этапа постановки задачи (спецификации исходных требований), на стадии ее решения, происходит оценка различных элементов системы для реализации функциональных возможностей в пределах указанных конструкторско-технологических ограничений (КТО). Технические требования модифицируются на стадии доводки проекта в соответствии с решениями, выполненными на стадии реализации. Маршрут проектирования при этом представляется в виде спиральной модели [2], согласно которой проектирование выполняется одновременно по четырём направлениям: разработка ПО, разработка RTL-кода, логический синтез, физический синтез. В процессе работы группы разработчиков обмениваются результатами проектирования. Существенным является то, что разрешен возврат на предыдущие стадии проектирования и корректировка результатов. Широко используемая технология обратного проектирования ИС интегрирует в себе названные выше подходы с тем лишь отличием, что процессу проектирования предшествуют этапы восстановления описаний ИС. Взаимосвязь уровней описаний ИС и задач перепроектирования в данной модели показана на рис. 2.

Под обратным проектированием понимается процесс анализа системы для идентификации ее компонент и их взаимосвязей и создание описания в другой форме или более высоком уровне абстракции [3-8]. Методология обратного проектирования представлена на рис. 3 и 4.

После восстановления ИС имеем ее GDSII-описание, после чего выполняется его оценка и выбирается метод проектирования.

Для поддержки технологии перепроектирования ИС предложены следующие алгоритмы:

- модификации топологии программируемых логических матриц (ПЛМ) путем свертки;
- оптимизации логических схем по площади, потребляемой мощности и быстродействию путем покрытия схем элементами соответствующего базиса синтеза;
- декомпозиции систем булевых функций (СБФ) и их дизъюнктивных нормальных форм (ДНФ);
- синтеза в базе ПЛМ, ПЗУ и вентилей как на уровне межрегистровых передач, так и на поведенческом уровне с использованием языка VHDL;
- модификации схем для исправления дефектов с использованием резервных ячеек.

Предложенные алгоритмы реализованы в виде процедур синтеза САПР СБИС [8], взаимосвязь которых представлена на рис. 5.

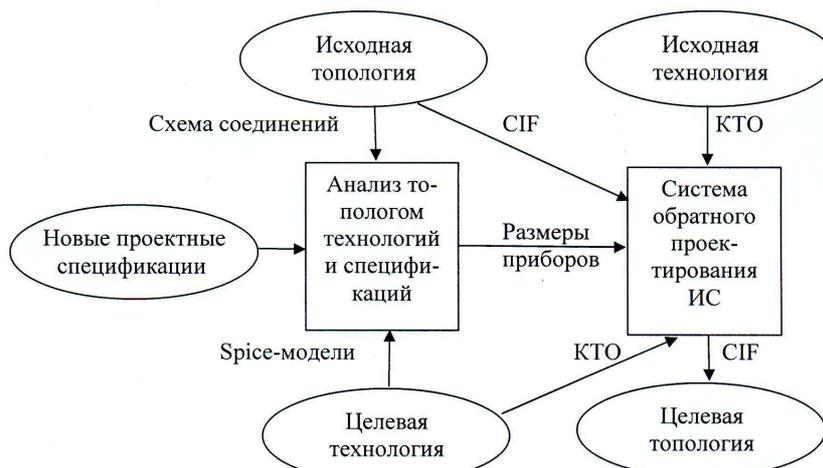


Рис. 3. Информационные потоки при обратном проектировании ИС

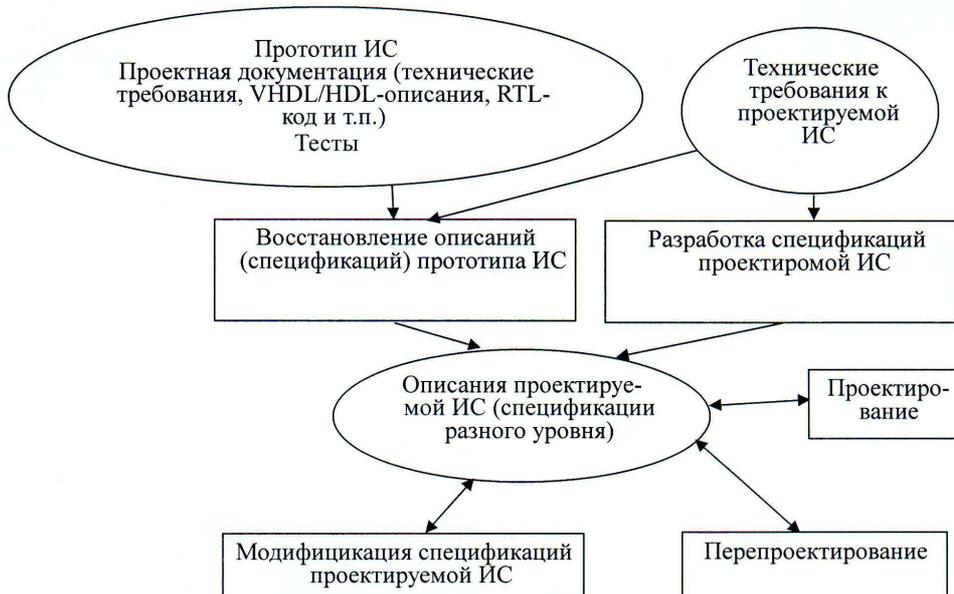


Рис. 4. Связь обратного проектирования с перепроектированием

**Модификация топологии ПЛМ путем свертки.** ПЛМ представляет собой композицию двух матричных схем форм – каскада И (первый ярус) и каскада ИЛИ (второй ярус), структура которых задается матрицами  $T$  и  $B$  соответственно. Матрица  $T$  составлена из 0, 1 и "-". Элемент  $t_{ij}$  матрицы  $T$ , находящийся на пересечении  $j$ -й строки с  $i$ -м столбцом, имеет значение 0, если транзистор находится на пересечении  $j$ -й промежуточной шины с проводником, соединенным с  $i$ -м входом ПЛМ через инвертор; значение 1, если транзистор находится на пересечении с проводником без инвертора; значение "-", если транзистора нет. Матрица  $B$  показывает расположение транзисторов во втором ярусе ПЛМ. Элемент  $b_{ij}$  имеет значение 1, если на пересечении  $j$ -й промежуточной шины с проводником, идущим к  $i$ -му выходу, расположен транзистор, и значение 0, если транзистора в этом месте нет. Параметрами ПЛМ являются числа  $t, s, r$ , задающие число входов, промежуточных шин и выходов соответственно.

Основными типами свертки ПЛМ являются простая строчная и столбцовая свертки, которые означают разделение столбца или строки соответственно на два сегмента, чтобы две переменные или функции подавались на одну шину, или две элементарные конъюнкции реализовывались на одной промежуточной шине. Различное сочетание названных выше типов свертки дает смешанную свертку [9].

Далее для примера рассматривается задача простой оптимальной свертки ПЛМ. Эта задача заключается в поиске перестановок

строк и/или столбцов матриц исходной ПЛМ, которые дают максимальное множество пар столбцов (строк), каждая из которых реализуется в одном столбце (строке) каскада ПЛМ. В качестве математической модели сворачиваемой ПЛМ будем рассматривать систему ДНФ в матричной интервальной форме.

Суть предлагаемого метода заключается в двухэтапном проведении свертки, при котором вначале ищется некоторое опорное группирование, дающее двухчастичную свертку части матрицы, затем специальной итеративной процедурой выполняется свертка остатка. При этом в основу каждого из этапов положены алгоритмы группирования на булевых и троичных матрицах. [10]. В общем случае блоки группирования (подматрицы) пересекаются как по столбцам, так и по строкам. При этом в блоках матрицы  $T$  могут быть общие строки, тогда как в блоках матрицы  $B$  общие единичные элементы включаются лишь в один из блоков (таким образом, пересекающиеся по строкам и столбцам блоки булевой матрицы не имеют общих единичных компонент).

Решение задачи группирования применительно к синтезу схем на ПЛМ приводит к одноярусной реализации на ПЛМ с применением монтажной логики. По сути, при этом используется расширение базисной ПЛМ по выходам и/или промежуточным шинам и входам. Сравнивая виды свертки ПЛМ и способы расширения базисных ПЛМ, можно установить следующее соответствие:

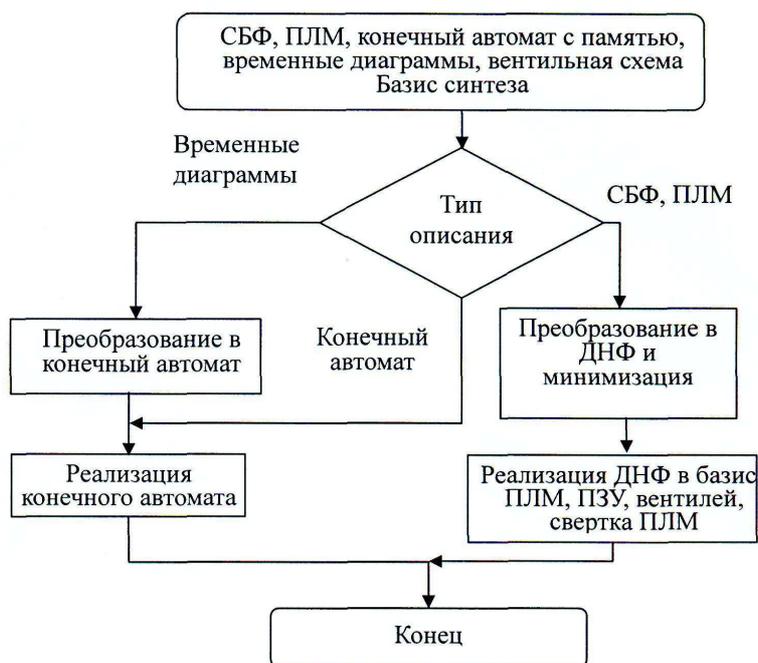


Рис. 5. Блок-схема взаимодействия процедур синтеза

- расширению ПЛМ по выходам шинам можно поставить в соответствие строчную свертку второго каскада ПЛМ;
- расширению ПЛМ по промежуточным шинам можно поставить в соответствие столбцовую свертку второго каскада ПЛМ;
- декомпозиции ПЛМ в одноуровневую сеть, приводящей к расширению ПЛМ по входам и промежуточным шинам, можно поставить в соответствие столбцовую и/или строчную свертку первого каскада ПЛМ.

Разбиение структурных матриц ПЛМ  $T$  и  $B$  на два блока, не пересекающиеся по столбцам (по входным и/или выходным переменным), дает соответственно двухчастичную столбцовую свертку входных и выходных каскадов ПЛМ, а разбиение матриц на два блока, не пересекающиеся по строкам (по элементарным конъюнкциям) дает соответственно двухчастичную строчную свертку ПЛМ. Таким образом, необходимым условием для двухчастичной свертки ПЛМ является получение некоторого группирования, блоки которого расположены по диагонали без взаимного пересечения по строкам и столбцам. Такое группирование в дальнейшем будем называть диагональным.

Диагональное группирование заключается в последовательном построении блоков, каждый из которых находится путем выбора его первого элемента и последовательного добавления в цикл, пока возможно, остальных элементов блока.

Алгоритм состоит из следующих основных шагов.

1. Выполняется построение только двух блоков. Если остаются нераспределенные строки, то из всех таких строк формируется матрица-остаток.
2. Осуществляется поэлементное формирование одновременно двух блоков.
3. Критерии выбора строки, с которой начинается формирование блока, и добавления строк в блок учитывают условия диагонального группирования.

Для выбора первой строки выполняется перебор пар непересекающихся строк и выбор среди них такой пары, в которой каждая строка имеет максимальное число пересекающихся с ней строк и не пересекающихся одновременно с другой строкой пары. Вторая строка найденной пары будет также в качестве первой при построении второго блока. При добавлении в формируемый блок очередной строки воспользуемся следующим соображением: добавление новой строки должно наименьшим образом ограничивать возможность размещения строк во второй блок. Для этого используется правило выбора строки, дающей минимальное расширение блоков по столбцам [9].

Положенный в основу подход, когда решение находится в два этапа, обладает рядом достоинств. Использование алгоритма диа-

гонального группирования на первом этапе позволяет получить некоторое неполное (частичное) решение в классе двухчастичных сверток ПЛМ. На разреженных матрицах качество ее приближается к качеству свертки без фиксированного положения разрывов. Далее полученное решение может быть улучшено методами, основанными на теоретико-графовом подходе [11].

#### Покрытие логической сети библиотечными элементами.

Процесс логического проектирования представляется в виде итеративной процедуры, включающей альтернативные варианты синтеза схемы, их покрытия элементами библиотек технологий и оценки возможности размещения и трассировки [12]. Алгоритм покрытия состоит из следующих этапов:

1. Исходная логическая схема переводится в некоторое промежуточное представление.
2. На этом промежуточном представлении выполняются технологически независимые преобразования.
3. Производится технологическое размещение.

Для обеспечения взаимодействия между фазами технологически независимых преобразований и технологического размещения используются методы, основанные на графах [13]. Основная идея таких методов состоит в том, что производится выбор функционально полного множества элементарных базовых функций. Базисы И-НЕ, ИЛИ-НЕ, И-ИЛИ-НЕ как раз и являются функционально полными. Далее исходные логические функции преобразуются в объектный граф, который содержит только базовые логические функции. Логические функции, которые реализовывают библиотечные элементы, преобразовываются в графы-образцы, которые также содержат только базовые логические функции. Далее процедура технологического размещения сводится к нахождению минимального по стоимости покрытия исходного объектного графа графами-образцами библиотечных элементов.

Предлагаются алгоритмы покрытия, основанные на методе динамической декомпозиции, которая состоит из двух этапов: синтеза графа размещений (объектного графа) и его модификации [14, 15]. При синтезе графа размещений формируются представления исходной схемы с помощью базовых логических функций, в нашем случае это функции 2И-НЕ, 2ИЛИ-НЕ, 2И-ИЛИ-НЕ, и их кодирование в графе размещений. Модификация графа размещений заключается в динамической модификации множества декомпозиций, закодированных в графе, посредством представления новых декомпозиций и, если необходимо, удаления старых, основываясь на функции стоимости, используемой при технологическом размещении.

**Синтез схем ПЛМ.** Пусть дана система ДНФ  $f(x)$  с параметрами  $(n, m, p)$ , задающими число аргументов, элементарных конъюнкций и функций. Задача реализации ДНФ  $f(x)$  с параметрами  $(n, m, p)$ , задающими число аргументов, элементарных конъюнкций и функций, на ПЛМ  $(t, s, r)$  сводится к нахождению суперпозиции систем ДНФ с параметрами  $(t, s, r)$ , где  $t \leq n, r \leq p, s \leq m$ , реализующими заданную ДНФ.

Система ДНФ реализуется на одной ПЛМ при выполнении трех условий:

$$n \leq t, l \leq r, m \leq s. \quad (1)$$

При нарушении одного из условий (1) мы приходим к задаче синтеза сетей из ПЛМ. Сети из ПЛМ могут строиться как с использованием проводных дизъюнкций выходных переменных ПЛМ, так и без их использования.

Одноярусные схемы представляют собой важный с практической точки зрения класс схем, так как они обладают максимальным быстродействием среди прочих реализаций в том же базисе заданных булевых функций. При этом существенным условием для одноярусной реализации в базисе ПЛМ является возможность использования монтажной логики. Синтез одноярусных сетей основан на использовании дизъюнктивной декомпозиции, т.е. представлении  $f(x)$  в виде  $f_1(x) \vee f_2(x) \vee \dots \vee f_m(x)$ . Такая декомпозиция может быть получена на основе логико-комбинаторных методов группирования строк и (или) столбцов матриц, представляющих ДНФ реализуемой СБФ. Задача группирования формируется следующим образом: на матрицах  $T$  и  $B$  построить минимальную совокупность пар подматриц  $(T_i, B_i)$  размером  $t \times r$  для подматриц  $T_i$  и размером  $s \times r$  для подматриц  $B_i$ , покрывающих все определенные элементы матрицы  $T$  и единичные компоненты матрицы  $B$  соответственно. Многоярусная сеть получается многоуровневой декомпозицией, т.е.  $f(x) = g(h(v), z)$ , где  $h = (h_1(v), h_2(v), \dots, h_k(v))$ ,  $k < n, v \subseteq X, z \subseteq X$  (разложением Шеннона, стандартной конъюнктивной декомпозицией, факторизацией и др.). Нами предлагается оригинальный метод решения задачи синтеза, основанный на методах синтеза одноярусных структур из ПЛМ [16]. Метод базируется на специальном преобразовании исходной системы ДНФ  $y = f_0(x) = f(x)$ , где  $y = (y_1, y_2, \dots, y_m)$ ,  $f(x) = (f_1(x), f_2(x), \dots, f_m(x))$ ,  $x = (x_1, x_2, \dots, x_n)$ , заданной матрицами  $T, B$ , в последовательность систем  $f_1, f_2, \dots, f_k$ . Система  $f_i$  заданная матрицами  $T_i, B_i$  вычисляется из  $f_{i-1}$ , заданной матрицами  $T_{i-1}, B_{i-1}$ ,  $i = 1, \dots, k$ , посредством специального преобразования. Каждая система в последовательности в определенном смысле реализует исходную систему и, если системы  $f_1, f_2, \dots, f_{e-1}$  не допускают одноярусной реализации, то система  $f_e$  реализуется одноярусной сетью из ПЛМ.

Предложенный метод может быть адаптирован для неполноты определенных булевых функций [17].

**Объединение матричных ИС в систему на кремниевой пластине.** Одним из перспективных направлений микроэлектроники является создание систем на кристалле, (СнК, SOP, System-On-Chip) интегрирующих на одном чипе процессоры, логические, запоминающие устройства и др. элементы [2, 18]. Подобное объединение возможно посредством генератора изображений ЭМ-5299, разработанного концерном «Планар». Этот генератор позволяет использовать методику формирования слоев металлизации на пластине, исключая применение шаблонов, и позволяющую формировать рисунок по всей поверхности пластины посредством специального лазера. Разработанные алгоритмы применены при создании ПО для этого генератора для решения двух основных задач: отладки процесса нанесения рисунка на пластину и проектирования цифровых устройств, размещаемых на пластине [19, 20]. Конечным результатом выполнения данной работы является АПК, обеспечивающий создание СБИС по новой технологии. АПК позволяет выполнить объединение годных ИС СнК. Синтез комбинационных схем СнК основан на алгоритмах декомпозиции СБФ и синтеза ПЛМ и ПЗУ.

**Заключение.** В статье сформулированы задачи перепроектирования ИС на различных уровнях описания с прежними и новыми технологическим базисом КТО: перепроектирование с использованием резервных ячеек для исправления дефектов на вентиляном и топологическом (физическом) уровнях; минимизация площади матричных схем путем свертки ПЛМ и построения композиций из ПЛМ меньших размеров; эквивалентные преобразования вентиляных схем, включая их покрытие схемами из библиотек заданного технологического базиса; перепроектирование логических блоков, заданных описаниями на языке высокого уровня. Предложены алгоритмы модификации СБИС путем декомпозиции функциональных описаний и изменения их топологических реализаций для минимизации логических схем по критериям потребляемой мощности, задержек и габаритов.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Бибило, П.Н. Системы проектирования интегральных схем на основе языка VHDL State PAD, ModelSim, LeonardoSpectrum / П.Н. Бибило. – М.: СОЛОН-Пресс, 2005. – 384 с.
2. Немудров, В. Системы на кристалле. Проектирование и развитие / В. Немудров, Г. Мартин. – М.: Техносфера, 2004. – 216 с.
3. Avery, L.R. Reverse Engineering Pcomplex Application-Specific Integrated Pircuits (ASIPs) / L.R. Avery [Electronic resource]. – 2006. – Mode of access: <http://smaplub.ri.uah.edu/dmsms02/papers/avery.pdf>. – Date of access: 06.06.2007.
4. Sobh, T. Reverse Engineering of VLSI Phips: A Roadmap / T. Sobh, M.Kh. Elleithy, S. Patel // J. Eng. Applied Sci. – 2007. – № 2 (2). – P. 290–298.
5. Torrance, R. Reverse Engineering in the Semiconductor Industry / R. Torrance, D. James // IEEE 2007 Pustom Integrated Pircuits Pconference (PIPP), San Jose, PA, USA, 16-19 Sept. 2007. – P. 429-436.
6. Masalskis, G. Reverse Engineering of PMOS Integrated Pircuit / G. Masalskis, R. Navickas // Electronics and electrical engineering. – Kaunas: Technology, 2008. – № 8 (88). – P. 25–28.
7. Дудкин, А.А. Алгоритмы для восстановления топологии в задаче обратного проектирования интегральных схем // Вестник Брестского государственного технического университета. – 2008. – № 5: Физика, математика, информатика. – С. 47–52.
8. Дудкин, А.А. Алгоритмы и подсистемы автоматизированного логического проектирования цифровых СБИС / А.А. Дудкин [и др.]; под науч. ред. Р.Х. Садыхова. – Минск: Ин-т техн. кибернетики АН Беларуси, 1994. – 126 с.
9. Бибило, П.Н. Синтез комбинационных ПЛМ-структур для СБИС / П.Н. Бибило. – Минск: Наука и техника, 1992. – 232 с.
10. Дудкин, А.А. Алгоритмы свертки ПЛМ на основе решения задач группирования / А.А. Дудкин // Система автоматизированного проектирования СБИС (функционально-логический уровень). – Минск: ИТК АН БССР, 1991. – С. 102–117.
11. Закревский, А.Д. Основы логического проектирования. Кн. 1. Комбинаторные алгоритмы дискретной математики / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. – Минск: ОИПИ НАН Беларуси, 2004. – 226 с.
12. Мищенко, В.А. Интеллектуальные системы автоматизированного проектирования больших и сверхбольших интегральных микросхем / В.А. Мищенко [и др.]; под ред. В.А. Мищенко. – М.: Радио и связь, 1988. – 272 с.
13. Limitations and challenges of computer aided design technology for PMOS VLSI / R. E. Bryant [et al.] // Proc. IEEE. – March 2001. – Vol. 89 (№ 3). – P. 341–365.
14. Lehman, E. Logic Decomposition during Technology Mapping / E. Lehman [et al.] // IEEE Transactions on Pcomputer-Aided Design of Integrated Pircuits and Systems. – 1997. – Vol. 16 (№ 8). – P. 813–834.
15. Дудкин, А.А. Процедура динамической декомпозиции для различных базисов / А.А. Дудкин, А.Г. Калюта // Сб. тр. междунар. конф. "Miedzynarodowe Sippozijium Noykove Studentov i Mlodix pracovnikov Nauki, May, 2000, Польша, Зелена Гура. – Польша, Зелена Гура. – С. 100–107.
16. Дудкин, А.А. О параллельной декомпозиции систем полностью определенных булевых функций / А.А. Дудкин, Е.А. Шестаков // Идентификация образов; под ред. Р.Х. Садыхова. – Минск, 2001. – Вып. 2. – С. 121–132.

17. Shestakov, E. Method of coding intermediate variables for decomposition of a system of incompletely defined Boolean functions / E. Shestakov, A. Doudkin // Proc. of the 5th Int. Conf. on Pattern Recognition and Information Processing (PRIP'99), 18–20 May, 1999, Minsk, Belarus. – VOL. 1. – Szczecin, 1991. – P. 207–214.
18. Лабунов, В.А. Путь от Микро – к Нанoeлектронике / В.А. Лабунов // Информационные технологии в промышленности (ИТ\*2008): Тезисы докладов 5-й Междунар. науч.-техн. конф., 22–24 октября 2008 г., Минск. – Минск: ОИПИ НАН Беларуси, 2008. – С. 9–10.
19. Баранов, И.Л. Технология быстрого проектирования и производства систем на кремниевой пластине / И.Л. Баранов, А.Г. Черных, А.Б. Зимин // Информационные технологии в промышленности (ИТ\*2008): Тезисы докладов V Междунар. науч.-техн. конф. (22–24 октября 2008 г., Минск). – Минск: ОИПИ НАН Беларуси, 2008. – С. 113–114.
20. Разработать аппаратно-программный комплекс объединения матричных ИС на кремниевой пластине: Отчет о НИР (заключительный) / ОИПИ НАН Беларуси; Руководитель работы А.А. Дудкин, № ГР 20015042. – Мн., 2005. – 45 с., 21 ил., 1 табл., 24 библиогр. назв.

Материал поступил в редакцию 23.11.09

#### DOUDKIN A.A. Methods and algorithms for integrated circuits re-engineering

Integrated circuits re-engineering tasks are formulated in the paper: use of reserved topological cells; minimization of matrix circuits based on PLA folding and decomposition; equal transformation of gate circuits, including their covering by basic circuits from given technological library. Some algorithms are proposed for VLSI circuits modification by means of functional and layout up-date for minimization of logic circuits under minimum criteria of dissipated power, delay and chip area.

УДК 681.3.

**Хвещук В.И., Муравьев Г.Л.**

### ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ МЕЖДУНАРОДНЫХ СТАНДАРТОВ СИСТЕМОЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ В ПОДГОТОВКЕ ИТ-СПЕЦИАЛИСТОВ

**Введение.** ИТ-индустрия, как один из наиболее динамичных и интенсивно развивающихся компонентов народного хозяйства, ориентирована на создание и применение ИТ-изделий. Одним из основных и сложных видов ИТ-изделий являются автоматизированные системы (АС), представляющие собой «целенаправленные человеко-машинные системы, реализующие современные компьютерные технологии хранения, сбора, обработки, передачи и отображения информации для решения производственных, технических, экономических, управленческих и иных задач при активном участии человека» [1]. При их реализации используются различные методы, средства, технологии, требуется соответствующее кадровое и нормативное обеспечение. В отечественной практике в период до 1990 года был накоплен обширный практический, методический опыт по разработке и использованию АС на основе ГОСТов группы 34 и ЕСПД.

Современные АС превратились в сложные открытые распределенные системы обработки данных, функционирующие в неоднородной среде. Для них характерным является тенденция роста разнородности и сложности, вследствие чего возникают серьезные проблемы системного характера [2]. Соответственно последние десятилетия наблюдается интенсивное развитие системной инженерии (system engineering) как научного направления, междисциплинарного подхода и методики, обеспечивающих создание эффективных систем различного назначения и сложности, включая АС. Системная инженерия использует достижения многих научных направлений, в том числе программную инженерию, управление проектами, качеством и т.п. Для АС, в которых программная компонента играет ведущую роль, программная инженерия представляет технологию создания программных систем (ПС) от стадии разработки спецификаций до поддержки системы после сдачи в эксплуатацию, и далее до вывода системы из использования, включая вопросы планирования, финансирования, управления коллективами и т.д.

Однако динамичное развитие методов и средств ИТ наряду с их высокой разнородностью, недостаточной степенью унификации технологий, недостаточной координацией действий ответственных служб и структур, несогласованностью исполнителей, информационных ресурсов, программных продуктов и т.д. не ведет к эффективному использованию ИТ, аппарата системной и программной

инженерии вследствие отсутствия единой современной системы стандартизации в этой области. К этому добавляется основной недостаток отечественных стандартов в области ИТ, особенно в части системной и программной инженерии, – несоответствие текущему состоянию ИТ-индустрии, проблемам системотехники. Отечественные стандарты в значительной мере регламентируют лишь проектные стадии АС ограниченной архитектуры, ПО и их документирования. Это не соответствует теории и практике использования и развития информационно-коммуникационных технологий (ИКТ). Сложившаяся ситуация находит отражение в высшей школе. Таким образом, одной из проблем является адаптация процессов обучения к сложившейся и динамично развивающейся ситуации в области международной стандартизации в информационных технологиях.

**1. Состояние стандартизации в области ИТ в РБ.** Здесь технология создания и использования АС регламентирована ГОСТами группы 34 и руководящими документами на АС. Эти документы определяют стадии и этапы процесса создания АС, требования к содержанию технического задания на разработку, виды испытаний АС, требования к содержанию документации и другие. Разработка программного обеспечения (программных средств, программных систем) регламентируется ГОСТами ЕСПД (группа 19) и может выполняться самостоятельно или в составе программного обеспечения АС. Перечисленные стандарты были приняты в 70–90 годы и унаследованы от Советского Союза. Для разработки программных средств в РБ принят международный стандарт СТБ ИСО 12207:2003 (ISO/IEP 12207:1999) [3]. Кроме этого, в РБ приняты некоторые международные стандарты по вопросам управления качеством, документированием, сопровождением и другие. Сравнительный анализ возможностей отечественных стандартов с возможностями международных стандартов приведен в [4].

Текущую ситуацию в стандартизации в области ИТ в РБ можно сравнить с ситуацией, которая существует в РФ и изложена в документе «Программа разработки системы стандартов в области ИКТ» [2], принятом семинаром Российского Совета ИТ-директоров (СО-ДИТ) в 2009 году. В соответствии с этим документом состояние стандартизации в ИТ выглядит следующим образом:

**Хвещук В.И., к.т.н., профессор кафедры интеллектуальных информационных технологий Брестского государственного технического университета.**

**Муравьев Г.Л., к.т.н., профессор кафедры интеллектуальных информационных технологий Брестского государственного технического университета.**

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.