

## ВЫЯВЛЕНИЕ ВЫСОКОУРОВНЕВЫХ ИЕРАРХИЧЕСКИХ СТРУКТУР СВЕРХБОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ ЧЕРЕЗ СИЛЬНО СВЯЗАННЫЕ ЛОГИЧЕСКИЕ ГРУППЫ

В.И. Кузовлев<sup>1</sup>  
Н.А. Иванова<sup>1,2</sup>

KUZVI.bmstu@mail.ru  
ineldin@gmail.com

<sup>1</sup> МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

<sup>2</sup> Институт нанотехнологий микроэлектроники РАН, Москва, Российская Федерация

---

### Аннотация

Приведен метод автоматического анализа сверхбольших интегральных схем. В группах с нерегулярной структурой ячейки сильно связаны друг с другом, группы имеют больше внутренних связей, чем внешних. Выявлены связанные логические структуры, что позволяет выделить из транзисторной схемы потенциальный функциональный фрагмент, чтобы затем установить его назначение. На первом этапе формирования логически связанной группы происходит линейное упорядочение, на втором — выбор кандидата на добавление в группу, на третьем — улучшение и сокращение группы

### Ключевые слова

*Сверхбольшие интегральные схемы, связанные логические группы, функциональный анализ схемы*

---

Поступила в редакцию 02.12.2015  
© МГТУ им. Н.Э. Баумана, 2016

В процессе синтеза топологии сверхбольших интегральных схем (СБИС) логические структуры высокого уровня воплощаются в виде низкоуровневых логических транзисторных ячеек [1]. Схема, состоящая из низкоуровневых логических ячеек, передается в программное обеспечение, которое выполняет разводку топологических соединений и размещает ячейки на кристалле. На этом шаге данные об исходной логической структуре теряются. Большинство статей и литературы о размещении ячеек на кристалле СБИС (например [2, 3]) рассматривают только схему уровня логических ячеек, не принимая во внимание структурные и иерархические данные о схеме.

Большие группы ячеек с высокой плотностью межсоединений могут образовывать потенциальные функциональные элементы. Поиск функциональных элементов схож с обычной кластеризацией схем [1, 4], но есть два важных отличия. Во-первых, цель поиска функционального элемента — это определение больших (несколько сотен элементов) групп ячеек, которые представляют собой такие логические структуры, как сумматоры или дешифраторы, в то время как кластеризация имеет дело лишь с десятками ячеек. Во-вторых, при поиске функционального элемента необходимо получить группы ячеек, объединенных одной логической функцией, вместо того, чтобы приписывать каждую ячейку к кластеру, с целью снижения сложности.

Сильно логически связанные группы (ЛСГ) — это группы сильно связанных ячеек, которые с высокой вероятностью участвуют в формировании определенной логической функции. Помимо выделения крупных логических блоков, называемых макроэлементами (дешифраторы, автоматы и т. п.), ЛСГ можно использовать при решении проблем прямого проектирования, таких как [5]:

- *трассируемость*; поскольку ЛСГ имеют повышенную связность, то механизмы размещения будут располагать ячейки близко друг к другу, что часто приводит к возникновению мест перегрева. Карта топологических скоплений приведена на рис. 1. На ней красным отмечены места перегрева. В нижней левой части рисунка перегрев вызван плотным размещением элементов, которые входят в ЛСГ;

- *разработка общей топологии СБИС*; так как ЛСГ будут размещаться неразделимо, разработчик может сформировать «мягкий» блок (мягкие режимы групповой распайки блоков элементов на печатной плате — это режимы, при которых температура блоков изменяется плавно, а максимальная температура разогрева блока лишь на 10...30 К превышает температуру плавления припоя), который определяет ограничения для размещения ячеек;

- *перестройка по логике*; при автоматической трассировке топологии схемы основным критерием расстановки логических ячеек является компактность, потому в результате трассировка схемы получается спутанной.

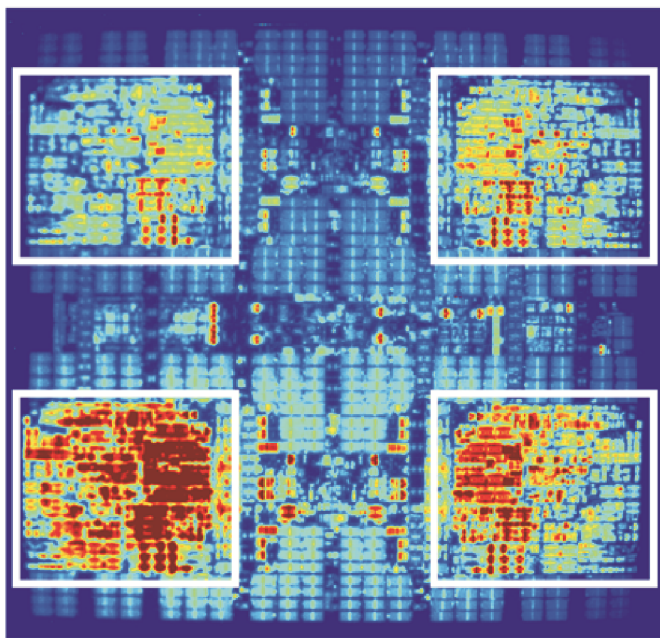


Рис. 1. Пример мест перегрева топологии

В отличие от обычного размещения ЛСГ может быть перераспределена таким образом, чтобы покрыть большее пространство и снизить число возможных точек перегрева. Если применять этот подход к небольшим областям топологии, то ее размеры увеличатся незначительно.

Существующие метрики кластеризации не позволяют сравнивать группы ячеек различных размеров. Необходимо так разграничивать структуры, чтобы компоновщик и трассировщик смогли правильно сформировать топологию схемы. Такие метрики и алгоритмы позволяют выделять как несколько небольших ЛСГ, так и одну крупную ЛСГ, включающую в себя более мелкие. Метрики масштабированы таким образом, что приблизительный коэффициент связности одного обычного кластера равен единице, а которые меньше обычного, оцениваются меньшими значениями (например, менее 0,1), что соответствует ЛСГ.

Далее будет показано, как использовать метрику, чтобы найти набор ЛСГ. Алгоритм начинает работу со случайно выбранного зерна и разрастается в ЛСГ путем итеративного добавления ячеек. Для одновременного разрастания нескольких зерен и исключения тех кандидатов, которые не подходят или накладываются друг на друга, используется распараллеливание потоков. Результатом является независимый набор обнаруженных ЛСГ.

**Существующие решения.** Логически связанная группа, как и кластер, является фрагментом схемы логических ячеек, поэтому складывается впечатление, что к ЛСГ можно применить традиционные кластерные метрики. Однако существуют некоторые явные различия в формулировках проблемы поиска сильно связанных логических структур и проблемы кластеризации ячеек.

1. Общепринятая кластеризация приводит к снижению размеров площади размещения. Такие кластеры, как правило, малы (от двух до десяти ячеек) для того, чтобы не потерять много данных при решении задачи уменьшения места расстановки. Кластеризация такого типа имеет локальную природу [6]. Однако функциональные элементы являются большими особыми логическими структурами, состоящими из сотен и тысяч ячеек, что требует более глобального подхода, учитывающего как внешние, так и внутренние связи групп.

2. При обычной кластеризации необходимо, чтобы каждая ячейка принадлежала кластеру, которые составляют всю схему СБИС. При выделении ЛСГ, напротив, рассматриваются особые подгруппы ячеек, которые будут размещаться другим приоритетным способом.

Пусть входная электрическая принципиальная схема СБИС будет гиперграфом  $G = (V, E)$ , где  $V$  (вершины графа) — это набор ячеек, а  $E$  (ребра графа) — набор связей таких, что каждая  $e \in E$  соединена с подмножеством  $V$ . Результатом кластеризации является набор отдельных подмножеств ячеек  $C_1, C_2, \dots, C_k \subset V$  таких, что  $V = C_1 \cup C_2 \cup \dots \cup C_k$ . Пусть дан кластер  $C$ , тогда разрез графа определяется как размер множества  $T(C) = \{e \in E | C \cap e \neq \emptyset \& (V - C) \cap e \neq \emptyset\}$ .

**Метрики кластеризации** могут получать разрез графа двумя различными способами, но в основном разрез не зависит от размера кластера. Это более уместно при делении или размещении сверху вниз, где размеры областей ограничены.

**Объединение** [7] — это метрика, считающая число внутренних соединений, которое растет с размером кластера, что плохо подходит для сравнения двух кластеров, как претендентов на ЛСГ. Кластер, больший по размеру, всегда будет обладать большим весом и, как следствие, большей способностью к объединению.

**Метрики разреза отношения и масштабированного разреза** [8] используют при расчете веса кластера по формуле  $\frac{T(C)}{|C|}$ . Поскольку  $T(C)$  растет намного медленнее, чем размер кластера, то больший кластер будет всегда иметь меньший вес, что не позволяет такой метрике сравнивать кластеры разных размеров.

В [9] предложено использовать экспоненту Рента для измерения параметров кластера, при этом вес кластера  $C$  пропорционален  $\frac{\ln T(C)}{\ln |C|}$ . Несмотря на то, что это более верно, чем при разрезе отношения, вес кластера по-прежнему монотонно убывает с его ростом.

В [10] представлена концепция метрики РС (разделение степени). Степень — среднее число связей, инцидентных каждой вершине в кластере, разделение — средняя длина самого короткого пути между двумя вершинами. Проход по случайному пути в этом методе помогает получить хорошую глобальную кластеризацию схемы. Однако недостаток этой метрики в том, что она не рассматривает внешних соединений кластера. Более того, среднее значение здесь используется для оценки качества всей кластеризации, а не отдельного кластера.

В [11, 12] приведены несколько сложных метрик: связность, раздельность граней и слипание, которые могут быть полезны при решении поставленной задачи. Однако эти метрики не учитывают разный размер кластеров и требуют значительных временных затрат при вычислении.

В итоге ни одна из рассмотренных методик не сравнивает кластеры разных размеров, не смещаясь в сторону больших или меньших. Предлагаемая метрика будет учитывать размер кластера при подсчете коэффициента его связности.

**Метрики для ЛСГ.** Такая метрика должна, во-первых, сравнивать кластеры различных размеров, во-вторых, измерять связность групп ячеек. Начнем с разреза отношения ( $RC$ ) и метрики Рента ( $Rent$ ) для каждого кластера  $C$ . Как обсуждалось ранее:

$$RC(C) = \frac{T(C)}{|C|} \quad Rent(C) \propto \frac{\ln T(C)}{\ln |C|},$$

где  $T(C)$  — число цепей в разрезе, а  $|C|$  — размер кластера,  $\propto$  — знак пропорциональности. Недостаток обеих метрик в том, что числитель (относящийся к разрезу) и знаменатель (относящийся к размеру кластера) не масштабируются относительно друг друга. Однако из правила Рента известно, что  $T(C)$  должен расти пропорционально  $|C|^p$ , где  $p$  — экспонента Рента,  $p < 1$ . Таким образом, можно задать вес ЛСГ ( $GTL-S$ ) как

$$GTL-S(C) = \frac{T(C)}{|C|^p}.$$

Можно предположить, что эта метрика будет постоянной для средней структуры. Здесь не принимаются во внимания кластеры малого размера, состоящие из десятка ячеек или области, поглощающие большую часть схемы СБИС.

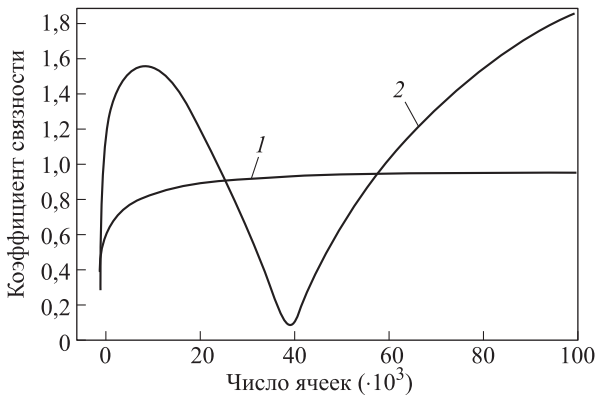
Пусть  $A(G)$  будет общим число пинов (входы, выходы ячеек группы) в  $G$ ; деленное на  $|V|$   $A(G)$  будет средним числом пинов ячейки. Согласно правилу Рента,  $A(G)$  будет являться предположительным значением  $GTL-S(C)$ . Тогда метрика развивается в нормализованную ЛСГ:

$$GTL-S(C) = \frac{T(C)}{A_G \cdot |C|^p}.$$

Такая нормализация позволяет не учитывать разность между схемами с низкой и высокой нагрузочными способностями ячеек. Подобное масштабирование определяет вес средней группы значительно больший, чем предполагаемый вес ЛСГ.

Чтобы проиллюстрировать поведение метрики на практике, учитывая степень интеграции ячеек, при которой выбирается случайная ячейка, вокруг которой разрастаются группы, постепенно присоединяя соседей с большим соединительным весом, был сгенерирован случайный граф из 250 000 ячеек. В этом графе 40 000 ячеек связаны между собой сильнее, чем с другими вершинами графа, таким образом, в графе есть одна ЛСГ размером в 40 000 ячеек.

На рис. 2 приведена функция  $GTL-S$  — это функция размера группы из двух массивов ячеек. Одна из кривых — функция для ячеек, которые находятся вне ЛСГ (1). Эта кривая начинается со значения 0,3 около нулевого размера группы и затем быстро разрастается, асимптотически приближаясь к 0,9, в то время как функция для второй группы ячеек, которые находятся внутри



**Рис. 2.** Пример  $GTL-S$  метрики (по оси абсцисс — число ячеек, по оси ординат — коэффициент связности)

ЛСГ (2), растет до значения 1,5, после чего резко снижается, достигая локального минимума в 0,1 как раз там, где все 40 000 ячеек обнаружены. Добавление тех ячеек, которые не принадлежат к ЛСГ, приводит к росту функции. Как только все ячейки ЛСГ будут включены в группу  $T(C)$ ,  $|C|^p$  достигнет минимума. После того, как к группе будут добавляться ячейки, которые ей не принадлежат,  $T(C)$  будет расти асимптотически к  $|C|^p$  по правилу Рента.

Чтобы логическая структура была связанной, число ее внутренних связей должно превышать число внешних. Чаще всего мультиплексоры представляют с помощью сложных ячеек, например из ячеек, выполняющих функции логического сложения *AND* или умножения *OR*, они обычно воплощают большее число функций на единицу площади. У таких ячеек, как правило, больше пинов (4 и более), чем у обычных ячеек, таких как инверторы, усилители, *2AND/2OR* (3 пина). Для того чтобы учесть плотность пинов и нормализовать *GTL*-метрику необходимо привести уравнение к виду

$$GTL-SD(C) = \frac{T(C)}{A_G |C|^{p_{AC}/A_G}},$$

где  $A_C$  — отношение числа пинов  $C$  к  $|C|$ , другими словами среднее число пинов ячейки в группе.

Отношение  $A_C / A_G$  стремится к единице, когда число пинов внутри  $C$  такое же, как и в остальной схеме. Однако, если  $C$  содержит несколько сложных ячеек, это отношение будет больше единицы, что делает его более похожим на ЛСГ. В результате умножения этого значения на экспоненту Рента весовая функция смещается в сторону групп с большим числом пинов, и, следовательно, с более связанной логикой. Метрика, в которой учитывается плотность пинов, обеспечивает проверку как маленьких, так и больших групп ячеек и найдет среди них ЛСГ, если они будут иметь высокую плотность.

На рис. 3 показана та же кривая, что и на рис. 2, только с учетом изменений, внесенных в последнее уравнение. Сравнивая два графика, выявили, что в обоих случаях ЛСГ в 40 000 ячеек была обнаружена, но во втором случае минимум был выражен ярче.

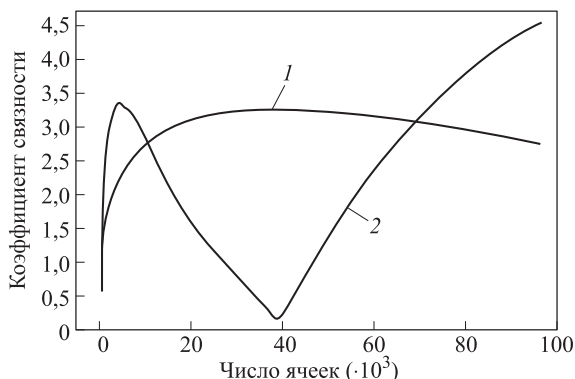


Рис. 3. Пример графика плотности ЛСГ

**Метод нахождения ЛСГ.** Далее предлагается прямой алгоритм нахождения ЛСГ, основанный на рассмотренных метриках. Он состоит из трех фаз.

Фаза 1: линейное упорядочение.

Фаза 2: получение начальных кандидатов ЛСГ.

Фаза 3: улучшение и сокращение ЛСГ.

Фазы 2 и 3 могут сочетаться с разными методами линейного упорядочивания [13].

**Фаза 1. Линейное упорядочение** инициализирует группу ячеек, зерновая ячейка которой выбрана случайно. Затем к группе добавляется по одной ячейке. Кандидатами на добавление в группу являются ячейки, которые не входят в группу, но соединены с ней напрямую. Среди этих кандидатов выбирается один с самым большим весом присоединения. Если у ячейки-кандидата  $v_i$  есть связь, не ведущая к группе, и у этой связи есть  $\lambda(e)$  пинов снаружи группы, то ее вес будет равен  $\frac{1}{\lambda(e)+1}$ . Чем больше пинов внутри группы, тем больше вес. Связь

между  $v_i$  и группой определяются как  $\sum_{e|v_i \in e, e \cap C \neq \emptyset} \frac{1}{\lambda(e)+1}$ . Для разрушения связей в качестве второго критерия используют минимальный разрез.

Таким образом, строятся группы связанных ячеек для получения возможного линейного порядка. По мере последовательного присоединения ячеек, функция весов будет влиять на увеличение степени связности.

В начале объединения ячеек важно уделять внимание связности  $\sum_{e|v_i \in e, e \cap C \neq \emptyset} \frac{1}{\lambda(e)+1}$ , а не только минимальному разрезу. Если ячейка-кандидат находится вне ЛСГ, как правило, у нее меньше связей с соседями, чем у ячейки внутри ЛСГ. Если использовать минимальный разрез как основной критерий объединения, скорее всего такая ячейка будет включена в группу. Аналогично, если ячейка-кандидат находится внутри ЛСГ, как правило, она имеет необходимое число связей с соседними ячейками, а критерий минимального разреза может исключить ее из ЛСГ. Порядок, по которому ячейки добавляются, задает линейный порядок. Чем большее предпочтение отдается соединениям, а не разрезу, тем плотнее будет группа и тем меньше будет внешних связей, тем больше к ЛСГ будет добавлено ячеек, которые и должны ей принадлежать.

**Фаза 2. Выбор начального кандидата ЛСГ.** Группа ячеек может быть получена из линейного порядка согласно метрикам. Пусть группа  $C$ , размера  $k = |C|$ , составлена из первых  $k$  ячеек линейного порядка, тогда функции  $GTL-SD(C)$  и  $GTL-S(C)$  с учетом  $k$  будут выглядеть как на рис. 2 и 3. В случае, если на графике функции присутствует ярко выраженный минимум, соответствующая группа ячеек будет кандидатом  $GTL$  «В». В процессе вычисления весовой функции  $nGTL$  необходимо принять значение экспоненты Рента  $p$  как среднее значение экспоненты Рента для всех групп, полученных в результате

линейного упорядочивания. Экспонента Рента для группы  $C$  может быть получена как  $\frac{\ln T(C) - \ln A_C}{\ln |C|}$ , где  $A_C$  — среднее число пинов ячейки  $C$ .

Этот метод описан для нахождения одной ЛСГ в схеме. Если начальное зерно находится за пределами ЛСГ, то дальнейшие действия не приведут к нахождению такой группы и на функции не будет минимумов. Для того чтобы найти ЛСГ, необходимо провести несколько поисков с различными начальными зернами. Если число поисков будет достаточным, то все ЛСГ будут найдены.

**Фаза 3. Улучшение и сокращение ЛСГ.** Кандидат ЛСГ, выращенный из случайного зерна, может не входить в ЛСГ. Например, если кандидат находится на границе ЛСГ, то могут быть включены некоторые ячейки, которые группе не принадлежат. Для того чтобы этого не происходило, необходимо найти для каждого кандидата  $V_i$ , полученного на 2-м этапе, набор других кандидатов  $V_{i,1}, V_{i,2}, \dots, V_{i,l}$ , используя зерна внутри  $V_i$  и те же действия, что на этапах 1 и 2. Эти кандидаты обычно близки, но немного отличаются от  $V_i$ . Затем к  $\{\hat{V}_{i,1}, \hat{V}_{i,2}, \dots, \hat{V}_{i,l}\}$  применяют операции объединения и пересечения, похожие на генетический алгоритм. Наконец, кандидат  $\hat{V}_i$  с лучшим весом относительно предложенных метрик выбирается как улучшенный кандидат относительно  $V_i$ . Эта процедура проводится над всеми начальными кандидатами в  $V$  для получения набора улучшенных кандидатов  $\{\hat{V}_{i,1}, \hat{V}_{i,2}, \dots, \hat{V}_{i,l}\}$ . Затем улучшенные кандидаты сравниваются друг с другом. Если один из кандидатов пересекается с другим и ухудшает вес ЛСГ, то он отсеивается. Отделенные кандидаты, оставшиеся в конце, являются результирующим набором ЛСГ предложенного метода.

Следует отметить, что все три фазы, описанные ранее, могут быть выполнены для  $t$  начальных зерен параллельно. Единственная часть алгоритма, которую необходимо выполнять последовательно, это сравнение полученных кандидатов в фазе 3.

**Эксперимент на реальных данных.** В результате применения полученных метрик и алгоритмов к фрагменту таблицы соединений СБИС Atmega 328 были получены минимумы на графике функции ЛСГ, которые соответствовали функциональным элементам схемы (рис. 4, а).

Такие минимумы носили локальный характер и затрудняли автоматизацию принятия решения о принадлежности группы ячеек к функциональному элементу. В результате анализа входных данных реальной таблицы соединений элементов СБИС было установлено, что сигналы сброса и тактирования, которые являются общими у большого числа ячеек — членов функциональных фрагментов, мешают четкому выделению структур по их функции, несмотря на то, что удовлетворяют всем требованиям предложенных метрик.

Выделить такие «общие» цепи в автоматическом или полуавтоматическом режимах можно по тому, как число элементов, подключенных к этим цепям, в несколько раз превышает число элементов, подключенных к остальным цепям схемы. Для того чтобы избавиться от влияния общих цепей на поиск ЛСГ, носящих одну



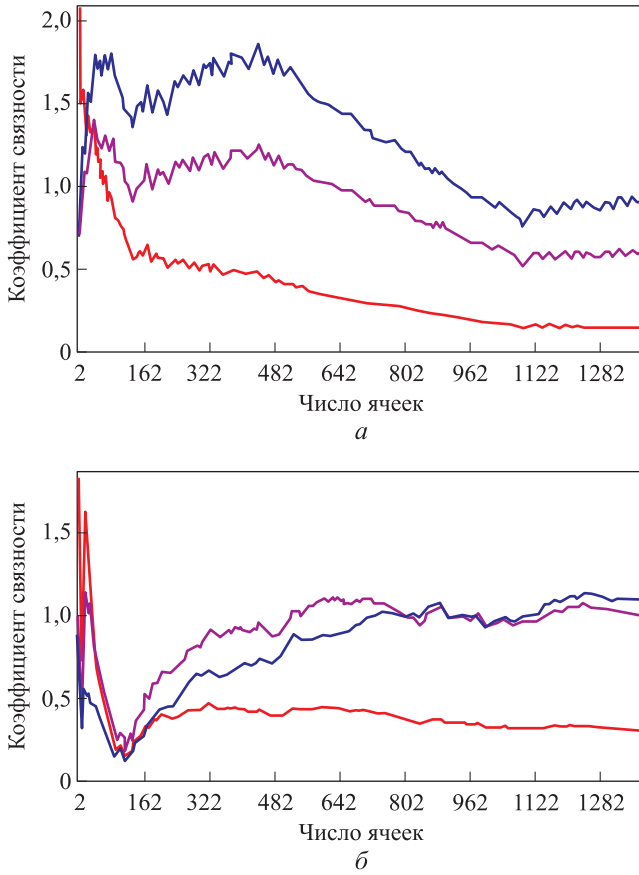


Рис. 4. Поиск ЛСГ с использованием метрик  $GTL-SD(C)$  (а) и  $GTL-SDW(C)$  (б)

функциональную нагрузку, необходимо ввести вес пинов, что позволит не учитывать их при подсчете плотности ЛСГ, задавая общим цепям меньший вес.

Такая система является гибкой, потому как позволяет распределять вес пинов в зависимости от загруженности подключенной цепи. Распределение веса не должно изменяться прямо пропорционально числу подключенных элементов. Вес должен резко уменьшаться с приближением к максимальному значению нагрузки цепи элементами. Тогда новая метрика, учитывающая общие цепи функциональных блоков, будет иметь вид

$$GTL-SDW(C) = \frac{T(C)}{A_G |C|^{p_{A,C}(w)/A_G}},$$

где  $A_{*C}(w)$  — среднее число пинов ячейки с учетом веса цепи, подключенной к пину.

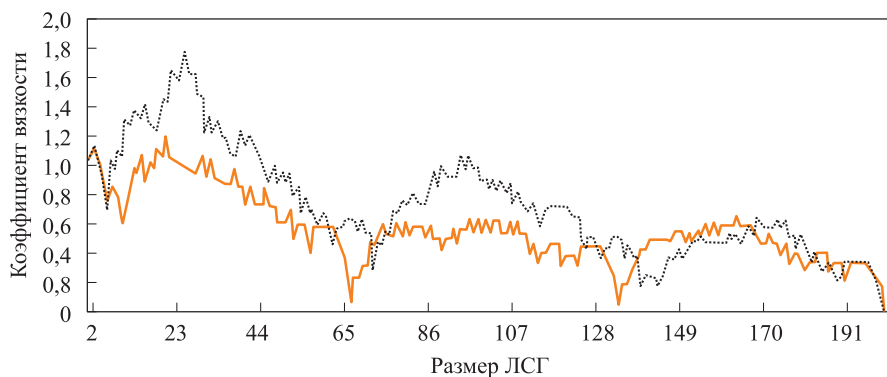
На рис. 4, б приведен график с использованием  $GTL-SDW(C)$ , на котором можно выделить глобальный минимум. Из него следует, что средний размер

ЛСГ составляет 80–120 элементов. Такие результаты можно получить, если зерно будет находиться внутри ЛСГ.

Вес пинов может распределяться не только исходя из загруженности цепи. Вводить различные атрибуты пинов, которые могут влиять на его вес, можно, исходя из характеристик элемента, которому принадлежит пин, и заведомо известных характеристик функции ЛСГ, которую необходимо найти. В случае, когда пин не содержит атрибутов, влияющих на его вес, можно заполнить это значение средним арифметическим соответствующих значений атрибутов ближайших объектов [14].

Этот алгоритм можно улучшить, изменив выбор кандидатов на первом шаге. Выбирая кандидата с заведомо более высоким значением  $GTL$ -функции можно избежать добавления в группу менее подходящих элементов.

На рис. 5 приведены сравнение работы алгоритма с простым линейным упорядочиванием элементов-кандидатов в ЛСГ (прерывистая линия) и выбор кандидатов с учетом их  $GTL$ -значения на примере трех последовательно соеди-



**Рис. 5.** Метрика  $nGTL$  с учетом  $GTL$ -значения для каждого кандидата (сплошная кривая) и без учета (штриховая кривая)

ненных одинаковых триггеров, представленных схемой транзисторного уровня. Из графиков следует, что минимумы стали более точными и число элементов в ЛСГ уменьшилось. Это обусловлено тем, что в первом случае в группу принимались лишние элементы, но минимум все равно был достигнут за счет сильной связности других элементов между собой. После изменения порядка подбора кандидатов в группу сложность алгоритма возросла не критично, но минимумы стали стремиться к нулю и размер группы теперь соответствует реальному размеру группы в поставленном эксперименте.

На рис. 6 приведена ЛСГ с большим числом входов и выходов, что представляет собой сложный случай для анализа, потому что число внешних связей близко к числу внутренних.

В результате поиска ЛСГ с учетом числа элементов в цепях (прерывистая линия, см. рис. 7) и улучшенным подбором кандидатов (сплошная линия, см. рис. 7) было получено, что оба подхода позволяют точно выявить элементы,

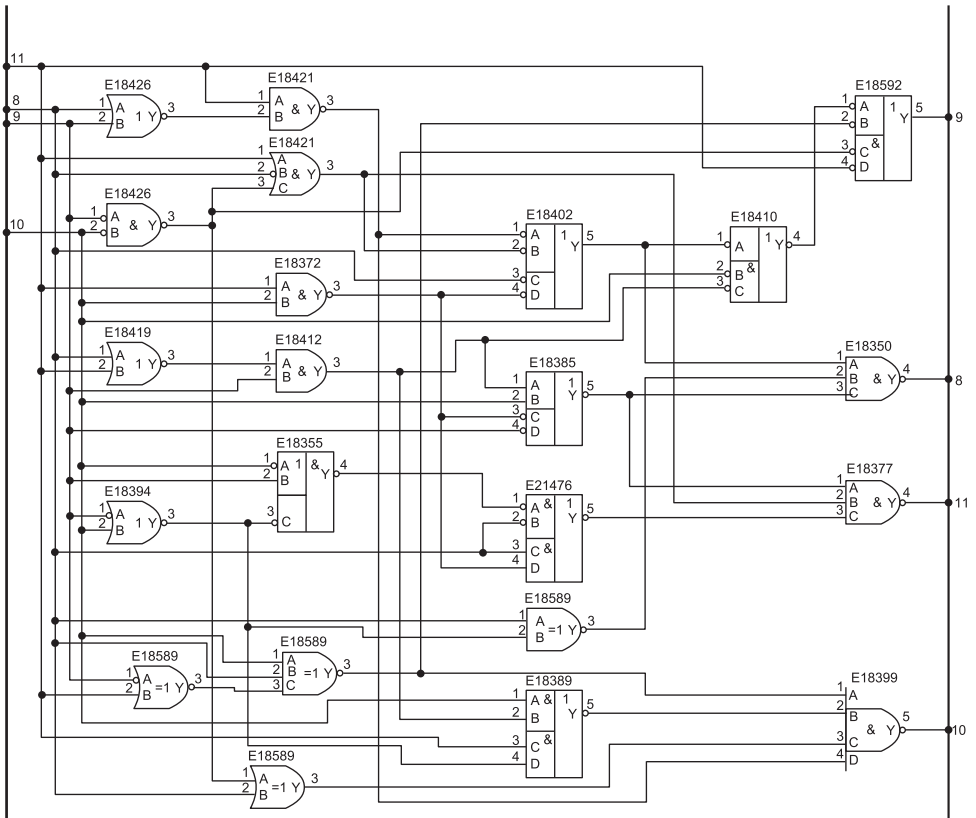


Рис. 6. ЛСГ с большим числом внешних связей

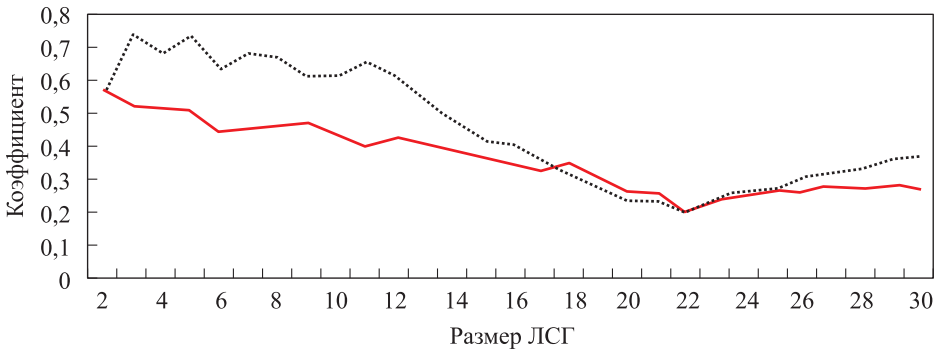


Рис. 7. Метрика *nGTLS* для ЛСГ с большим числом внешних связей

входящие в группу, но при подборе кандидатов минимум более ярко выражен при использовании метрики *nGTLS* (см. рис. 7).

Следовательно, в ЛСГ, в которых число внутренних связей значительно превышает число внешних, улучшенный алгоритм подбора кандидатов позволяет точнее выявлять элементы, принадлежащие ЛСГ. В группах, где число внутренних цепей близко к числу внешних, минимумы выражены сильнее.

Экспериментальные данные подтверждают работоспособность предложенных метрик. В отличие от структурного анализа и поиска по шаблонам, поиск ЛСГ не дает ответа на вопрос, какую функцию выполняет найденная группа элементов. С другой стороны, он не требует никаких знаний о возможных логических структурах СБИС и значительно облегчает ручной анализ схемы, благодаря выделению потенциальных функциональных элементов. Этот метод может быть использован вместе с шаблонным анализом с целью повысить быстродействие последнего. При этом сначала находят ЛСГ, затем среди них проводится поиск на совпадение с существующим шаблоном. Анализ схемы СБИС, основанный на поиске ЛСГ и сопоставлении с шаблонами при масштабах СБИС, превышающих несколько десятков тысяч базовых функциональных элементов, лучше всего проводить параллельно несколькими экспертами, через одну базу данных [15, 16]. Предложенные метрики позволяют эффективно определять сильные ЛСГ даже в случаях, когда число их связей внутри логической группы незначительно превышает число связей ячеек, не входящих в группу ячеек, выполняющих одну логическую функцию. В исходном алгоритме в группу принимались лишние элементы, но минимум все равно был достигнут за счет сильной связности между собой других элементов. После изменения порядка подбора кандидатов минимумы стали стремиться к нулю, а размер группы стал соответствовать реальному размеру. Предложенные весовые коэффициенты цепей позволяют объединять в ЛСГ ячейки, число внешних связей которых превышает число внутренних, так как благодаря весовым коэффициентам на формирование группы меньше влияют не только цепи с большим числом элементов, но и цепи, выделенные экспертом по другим функциональным признакам.

## ЛИТЕРАТУРА

1. Григорьян С.Г. Конструирование электронных устройств систем автоматизации и вычислительной техники. Ростов н/Д: Феникс, 2007. 303 с.
2. Phillip E. Allen, Douglas R. Holberg. CMOS Analog circuit design third edition. Oxford University Press. 2012.
3. Chen T., Hsu T., Jiang Z., Chang Y. NTUPlace: a ratio partitioning based placement algorithm for large-scale mixed-size designs // Proceedings of the International Symposium on Physical Design. 2005. P. 236–238.
4. Князев Б.А., Черненький В.М. Методика и модель кластеризации паттернов двигательной активности лица как преобразований метаграфов // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2014. № 4. С. 34–54.
5. mPL6: Enhanced multilevel mixed-size placement / T. Chan, J. Cong, J. Shinnerl, K. Sze, M. Xie // Proceedings of the 2006 International Symposium on Physical Design. 2006. P. 212–214.
6. Winn detecting tangled logic structures in VLSI netlists / Tanuj Jindal, Charles J. Alpert, Jiang Hu, Zhuo Li, Gi-Joon Nam, Charles B. Winn // Conference: Design Automation Conference'10. 13–18 June 2010.
7. Andrew B. Kahng, Jens Lienig, Igor L. Markov, Jin Hu. Netlist and system partitioning. Springer netherlands VLSI physical design: Graph Partitioning to Timing Closure. 2011. P. 31–54.

8. *Tae-Young Choe, Chan-Ik Park*. A k-way graph partitioning algorithm based on clustering by eigenvector. Springer Berlin Heidelberg Computational Science // ICCS. 2004. P. 598–601.
9. *Adya Saurabh N., Markov Igor L., Villarrubia Paul G*. Improving min-cut placement for VLSI using analytical techniques proc.: IBM ACAS Conference. IBM ARL, 2003.
10. *Ting-Chi Wang, Wang L.-C*. Multilevel circuit clustering for delay minimization. Computer-Aided Design of Integrated Circuits and Systems // IEEE Transactions on Volume 23, Issue 7, July 2004. P. 1073–1085.
11. *Cong J., Lim S. K*. Edge separability-based circuit clustering with application to multilevel circuit partitioning // IEEE Transactions on Computer-Aided Design. 2004. No. 23 (3). P. 346–357.
12. *Kudva P., Sullivan A., Dougherty W*. Metrics for structural logic synthesis // Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, 2002. P. 551–556.
13. *Jason Cong Fellow and Sung Kyu Lim*. Edge separability-based circuit clustering with application to multilevel circuit partitioning // Transactions on computer-aided design of integrated circuits and systems. 2004. Vol. 23. No. 3. P. 36–357.
14. *Коротаев А.И., Кузовлев В.И.* Система определения геометрических параметров трехмерных объектов // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2011. № 12. URL: <http://technomag.neicon.ru/doc/280112.html>
15. *Галкин В.А., Миляев Н.А.* Подходы к построению имитационных моделей сети хранения данных // Инженерный вестник. Электронный научно-технический журнал (МГТУ им. Н.Э. Баумана). 2013. № 8. С. 4. URL: <http://engsi.ru/doc/638235.html>
16. *Галкин В.А., Костенко Д.В.* Алгоритм выбора аппаратных средств сервера СУБД ERP-систем // Известия высших учебных заведений. Машиностроение. 2010. № 2. С. 50–59.

**Кузовлев Вячеслав Иванович** — канд. техн. наук, доцент кафедры «Системы обработки информации и управления» МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5).

**Иванова Наталья Андреевна** — младший научный сотрудник ИНМЭ РАН (Российская Федерация, 119991, Москва, Ленинский пр-т, д. 32А), аспирантка МГТУ им. Н.Э. Баумана (Российская Федерация, 105005, Москва, 2-я Бауманская ул., д. 5).

**Просьба ссылаться на эту статью следующим образом:**

Кузовлев В.И., Иванова Н.А. Выявление высокоуровневых иерархических структур сверх-больших интегральных схем через сильно связанные логические группы // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2016. № 4. С. 4–18.

DOI: 10.18698/0236-3933-2016-4-18

## VLSI CIRCUIT DETECTION THROUGH TANGLED LOGIC STRUCTURES

V.I. Kuzovlev<sup>1</sup>

KUZVI.bmstu@mail.ru

N.A. Ivanova<sup>1,2</sup>

ineldin@gmail.com

<sup>1</sup> Bauman Moscow State Technical University, Moscow, Russian Federation

<sup>2</sup> Institute of Nanotechnology and Microelectronics, Russian Academy of Sciences, Moscow, Russian Federation

## Abstract

This work proposes a method of automatic very-large-scale integration (VLSI) circuit analysis. Groups with irregular structure have highly interconnected cells; groups have more internal than external connections. Detecting Tangled Logic Structures (TLS) with a linear ordering allows to identify the functional structure of the circuit and the gate-level VLSI circuit. High-level functional blocks in circuit description consist of gate-level cells groups, which are also highly interconnected. TLS-blocks are smaller, they represent a cell of high-level circuit, and are thus more suitable for further functional circuit analysis than a gate-level VLSI circuit

---

## Keywords

*Very-large-scale integration (VLSI), tangled logic structures (TLS), functional circuit analysis*

## REFERENCES

- [1] Grigor'yana S.G. Konstruirovaniye elektronnykh ustroystv sistem avtomatizatsii i vychislitel'noy tekhniki [Design of electronic devices for automation systems and computers]. Rostov na Donu, Feniks Publ., 2007. 303 p.
- [2] Phillip E. Allen, Douglas R. Holberg. CMOS Analog Circuit Design Third Edition. Oxford University Press. 2012.
- [3] Chen T., Hsu T., Jiang Z., Chang Y. NTUPlace: a ratio partitioning based placement algorithm for large-scale mixed-size designs. *Proc. of the International Symposium on Physical Design*, 2005, pp. 236–238.
- [4] Knyazev B.A., Chernen'kiy V.M. Method and model for clustering facial activity patterns using metagraph transformations. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborost. [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.]*, 2014, no. 4, pp. 34–54 (in Russ.).
- [5] Chan T., Cong J., Shinnerl J., Sze K., Xie M. mPL6: enhanced multilevel mixed-size placement. *Proc. of the 2006 International Symposium on Physical Design*, 2006, pp. 212–214.
- [6] Jindal Tanuj, Alpert Charles J., Hu Jiang, Li Zhuo, Nam Gi-Joon, Charles B. Winn detecting tangled logic structures in VLSI netlists. *Conference: Design Automation Conference'10*. 2010, June 13–18.
- [7] Andrew B. Kahng, Jens Lienig, Igor L. Markov, Jin Hu. Netlist and system partitioning. Springer netherlands VLSI physical design. Graph Partitioning to Timing Closure, 2011, pp. 31–54.
- [8] Tae-Young Choe, Chan-Ik Park. A k-way graph partitioning algorithm based on clustering by eigenvector. Springer Berlin Heidelberg Computational Science. ICCS, 2004, pp. 598–601.
- [9] Adya Saurabh N., Markov Igor L., Villarrubia Paul G. Improving min-cut placement for VLSI using analytical techniques. *proc. IBM ACAS Conference*. IBM ARL, 2003.
- [10] Ting-Chi Wang, Wang L.-C. Multilevel circuit clustering for delay minimization. *Computer-Aided Design of Integrated Circuits and Systems*, 2004, vol. 23, no. 7, pp. 1073–1085.
- [11] Cong J., Lim S.K. Edge separability-based circuit clustering with application to multilevel circuit partitioning. *IEEE Transactions on Computer-Aided Design*, 2004, no. 23 (3), pp. 346–357.

- [12] Kudva P., Sullivan A., Dougherty W. Metrics for structural logic synthesis. *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, 2002, pp. 551–556.
- [13] Jason Cong Fellow and Sung Kyu Lim. Edge separability-based circuit clustering with application to multilevel circuit partitioning. *Transactions on computer-aided design of integrated circuits and systems*, 2004, vol. 23, no. 3, pp. 36–357.
- [14] Korotaev A.I., Kuzovlev V.I. System of determining geometric parameters of 3D objects. *Nauka i obrazovanie. MGTU im. N.E. Baumana* [Science & Education of the Bauman MSTU. Electronic Journal], 2011, no. 12. Available at: <http://technomag.neicon.ru/en/doc/280112.html>
- [15] Galkin V.A., Milyaev N.A. Approaches to the construction of storage area network simulators. *Inzhenernyy vestnik. Jelektr. nauchno-tekhn. zhurn.* [Engineering Bulletin. El. Publ. (MGTU im. N.E. Baumana)], 2013, no. 8, p. 4. Available at: <http://engsi.ru/doc/638235.html>
- [16] Galkin V.A., Kostenko D.V. The algorithm for selecting hardware of DBMS server of ERP-systems. *Izv. Vyssh. Uchebn. Zaved., Mashinostr.* [Proc. of Higher Educational Institutions. Machine Building], 2010, no. 2, pp. 50–59 (in Russ.).

**Kuzovlev V.I.** — Cand. Sci. (Eng.), Assoc. Professor of Information Control Systems Department, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation).

**Ivanova N.A.** — Research Scientist, Institute of Nanotechnology and Microelectronics, Russian Academy of Sciences (Leninskiy prospekt 32A, Moscow, 119991 Russian Federation); post-graduate student, Bauman Moscow State Technical University (2-ya Baumanskaya ul. 5, Moscow, 105005 Russian Federation).

**Please cite this article in English as:**

Kuzovlev V.I., Ivanova N.A. VLSI Circuit Detection Through Tangled Logic Structures. *Vestn. Mosk. Gos. Tekhn. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2016, no. 4, pp. 4–18. DOI: 10.18698/0236-3933-2016-4-4-18