

Повышение эффективности проектирования интегральных схем на ПЛИС с ограниченными трассировочными ресурсами

Т.В. Гарбулина, О.В. Лялинская, В.М. Хватов

Институт проблем проектирования в микроэлектронике РАН (ИППМ РАН),

garbulina_t@ippm.ru, olgaly@ippm.ru, khvatov_v@ippm.ru

Аннотация — В настоящее время на смену зарубежным программируемым логическим интегральным схемам (ПЛИС) приходят отечественные разработки. Однако существующие системы автоматизированного проектирования (САПР) поддерживают разработку ПЛИС определенных фирм производителей и не позволяют работать с новой архитектурой. Целью выполнения данной работы является разработка маршрута проектирования, применимого к ПЛИС с любой архитектурой, и повышение эффективности проектирования интегральных схем (ИС) на отечественных ПЛИС с помощью специально разработанных лингвистических средств на основе языка Tcl. В ходе исследования разработаны и реализованы программные механизмы для оптимального размещения компонентов устройства и трассировки их межсоединений. Приведены результаты моделирования, показывающие эффективность используемых методов.

Ключевые слова — ПЛИС (программируемая логическая интегральная схема), отечественные ПЛИС, маршрут проектирования, язык Tcl, FIFO, алгоритм Pathfinder, размещение, трассировка.

1. ВВЕДЕНИЕ

В настоящее время микросхемы ПЛИС, являясь универсальным конструктором из логических элементов, успешно и широко применяются в большинстве электронных систем практически во всех областях — от научных исследований до аэрокосмического приборостроения и массового производства электронной техники. ПЛИС повсеместно заменяют микропроцессорные узлы и специализированные интерфейсные микросхемы, особенно в части параллельной обработки больших объемов данных. Программируемые в процессе эксплуатации вентильные матрицы представляют собой универсальные микросхемы гибкой логики, которые могут программироваться для выполнения желаемой функции заказчика. ПЛИС можно описать как программируемое устройство с внутренним массивом логических блоков, окруженных кольцом программируемых блоков вводов-выводов, соединенных между собой через программируемые соединения. Такая структура позволяет проектировщикам гибко управлять вычислительными

ресурсами и выводить готовый продукт на рынок за относительно короткие сроки.

Главным преимуществом ПЛИС перед другими семействами интегральных схем (СИС, БИС, СБИС) является свойство реконфигурируемости — на одной и той же интегральной схеме возможно программирование различных устройств. Также основными достоинствами являются высокая скорость изготовления схемы, низкая стоимость конечного продукта и уменьшение конечной площади аппаратуры в сравнении с аналогичными устройствами, реализованными на БИС.

В настоящее время одной из важных задач отечественной радиоэлектронной промышленности является импортозамещение высокотехнологичной продукции, в том числе микросхем ПЛИС. Российские производители стараются решить эту задачу и уже имеют широкую номенклатуру. На данный момент их разработки используются в военно-промышленном комплексе, телекоммуникациях, системах связи и вычислительных комплексах.

Существующие САПР поддерживают разработку ПЛИС определенных фирм производителей, таких как Xilinx, Altera, Actel и других, но не обеспечивают возможность проектирования ПЛИС с новой архитектурой [1, 2]. Для конкурентоспособности с зарубежными производителями и обеспечения независимого развития собственных производств необходимы САПР, позволяющие работать с отечественными ПЛИС с новой архитектурой.

Быстрый переход от зарубежных ПЛИС и систем их проектирования к отечественным продуктам требует больших финансовых вложений. Поэтому с целью экономии средств и плавного перехода к отечественным САПР целесообразно рассмотреть маршрут, который использует опыт применения зарубежных программных продуктов, но адаптирован под ПЛИС отечественного производства. Это является одним из этапов перехода на маршрут проектирования, полностью использующий только отечественное программное обеспечение [3-8].

Дальнейшее содержание работы включает следующие разделы.

В разделе II представлен маршрут проектирования с использованием лингвистических средств на базе языка Tcl.

В разделе III рассмотрена возможность перехода к Tcl-интерфейсу для оптимизации размещения и трассировки.

В разделе IV(A) приведено описание модифицированного алгоритма PathFinder для трассировки межсоединений [9].

Численные эксперименты приведены в разделе IV(B).

II. ОПИСАНИЕ МАРШРУТА ПРОЕКТИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ЛИНГВИСТИЧЕСКИХ СРЕДСТВ НА БАЗЕ ЯЗЫКА TCL

Анализ существующих САПР для ПЛИС выявил необходимость создания новых алгоритмов, методов и средств конфигурирования ПЛИС, архитектура которых отличается от существующих. В рамках маршрута проектирования интегральных схем (ИС) на ПЛИС отечественного производства в ИППМ РАН были разработаны программные средства на языке Tcl для размещения и трассировки элементов ИС [10]. В разработанном маршруте также возможно использовать возможности САПР Altera (Quartus II) в области логического синтеза.

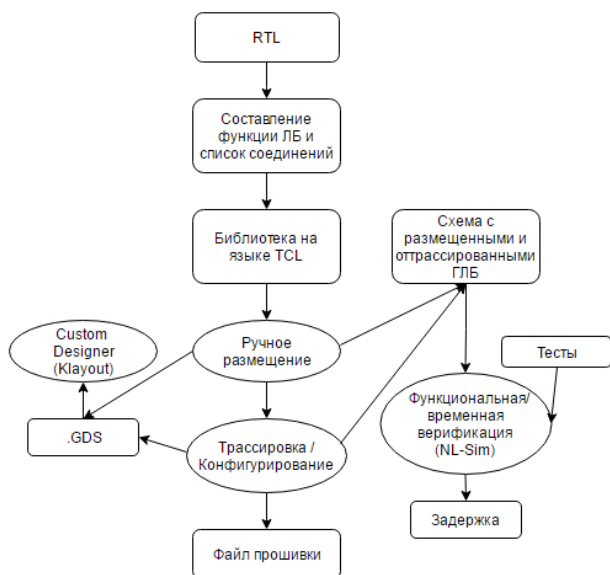


Рис. 1. Маршрут проектирования с использованием языка Tcl

Разработанный маршрут включает следующие этапы (рис. 1):

- анализ RTL-описания устройства;
- описание всех логических блоков, составляющих устройство, в базе ПЛИС и составление списка всех его соединений;
- генерация библиотеки для ПЛИС на языке Tcl;

- генерация описания всех составляющих устройство ЛБ на языке Tcl;
- ручное размещение всех ГЛБ на плате, передача результатов программе для дальнейшей трассировки;
- выполнение функциональной и временной верификации (в данном случае использовалась программа NanoLib-Sim).

NL-Sim разработана в ИППМ РАН и предназначена для моделирования и характеристики электрических схем цифровых КМОП БИС [11]. В разработанном маршруте проектирования на ПЛИС NL-Sim используется для моделирования переходных процессов в электрических схемах.

На основе RTL-описания устройство разделяется на составляющие ЛБ, которые представляются в виде логических функций в базе данной ПЛИС и далее описываются на языке Tcl, образуя самостоятельную библиотеку, состоящую из элементов LUT (look up table, произвольная таблица истинности).

Для примера того, как должно выглядеть описанное устройство можно воспользоваться логическим синтезом в САПР Altera Quartus II в базе библиотек Altera и посмотреть размещение ЛБ на плате, их логические функции и межсоединения с помощью встроенной в Quartus II утилиты Chip Planner.

САПР Quartus II – основной программный продукт компании Altera. Эта среда предназначена для проектирования проектов на ПЛИС Altera, при этом включает в себя их синтез, размещение элементов на схеме, трассировку соединений и верификацию.

Далее генерируется описание устройства на языке Tcl из имеющихся библиотечных ЛБ и списка межсоединений. После того, как все ЛБ описаны, их необходимо разделить на группы и разместить на ПЛИС.

Важно заметить, что Quartus II позволяет разместить каждую из доступных ГЛБ и заполнить все 10 ЛБ в каждой группе, что обусловлено минимальными ограничениями на трассировочные ресурсы ПЛИС Altera. В условиях жесткого ограничения таких ресурсов не получается провести размещение и трассировку автоматически и приходится использовать средства для гибкого ручного размещения (Tcl-скрипты).

В качестве примера рассмотрим автоматическое размещение и трассировку сложно-функционального логического блока синхронного буфера памяти FIFO размером 16 слов по 4 бита каждое [12, 13]. На первом этапе был проведен RTL-синтез блока FIFO в программе Synopsys Design Compiler в базе библиотеки стандартных ячеек. Далее было проведено автоматическое размещение логических блоков на ПЛИС с помощью программы на основе алгоритма моделирования отжига и их автоматическая

трассировка модифицированным алгоритмом PathFinder. Итоговое размещение (рис. 2) использовало все имеющее пространство ПЛИС, но, вследствие ограниченности трассировочных ресурсов, остались неразведенные ГЛБ.

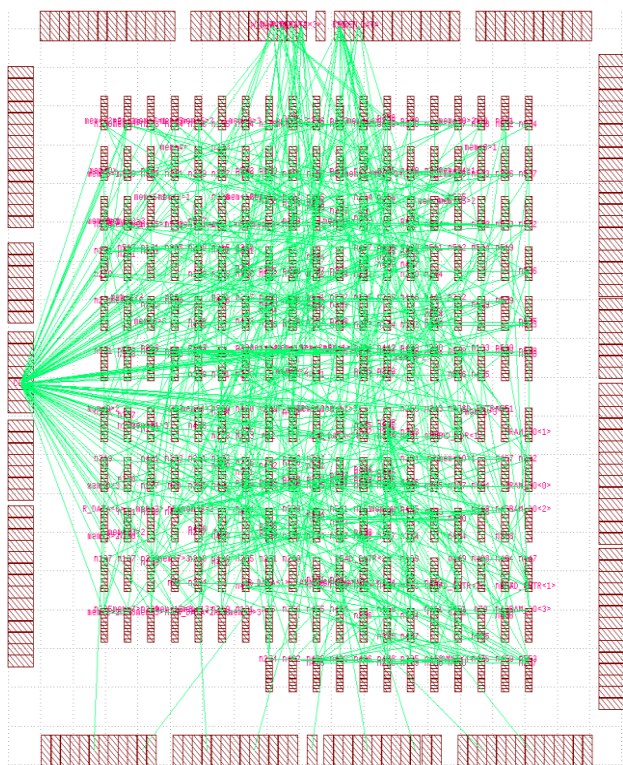


Рис. 2. Синхронный блок FIFO 16x4b. Автоматическое размещение и трассировка

В связи с этим возникает необходимость в разработке новых эффективных методов, алгоритмов и средств проектирования на ПЛИС.

III. МЕТОД ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПРОЕКТИРОВАНИЯ НА ОСНОВЕ УПРАВЛЯЕМОГО TCl ИНТЕРФЕЙСА

Для решения поставленной задачи был разработан маршрут проектирования, включающий ручное размещение и позволяющий проектировать на ПЛИС устройства, которые невозможно автоматически разместить с помощью готовых стандартных средств.

Устройство в созданном маршруте полностью описывается на языке Tcl для последующего гибкого и экономичного размещения и трассировки. Процесс размещения начинается с формирования ГЛБ таким образом, чтобы каждая группа имела определенное число входов (ограничение), используемых несколькими ЛБ. Описание блоков на языке Tcl позволяет сделать это, дав разработчику полную свободу действий. В случае неудачной трассировки нужно посмотреть результирующие GDS файлы в визуализаторе KLayout, воспользоваться методами композиции и декомпозиции ГЛБ или продублировать ячейки (в случае повторения большого количества сигналов). Также для оптимизации длины

межсоединений и площади Tcl-интерфейс дает возможность назначить ячейки входов/выходов там, где требуется в данной схеме.

В Tcl интерфейсе также предусмотрена возможность полуавтоматического размещения: часть ГЛБ вручную фиксируется (регулярная часть), оставшиеся ГЛБ размещаются программой.

Блок FIFO 16x4b был спроектирован по этому маршруту, то есть размещен вручную, в результате чего, в отличие от автоматического размещения, был полностью трассирован (рис. 3). Как видно из рисунка, итоговое размещение занимает гораздо меньше площади ПЛИС, чем автоматический вариант и затрачивает меньше каналов для трассировки.

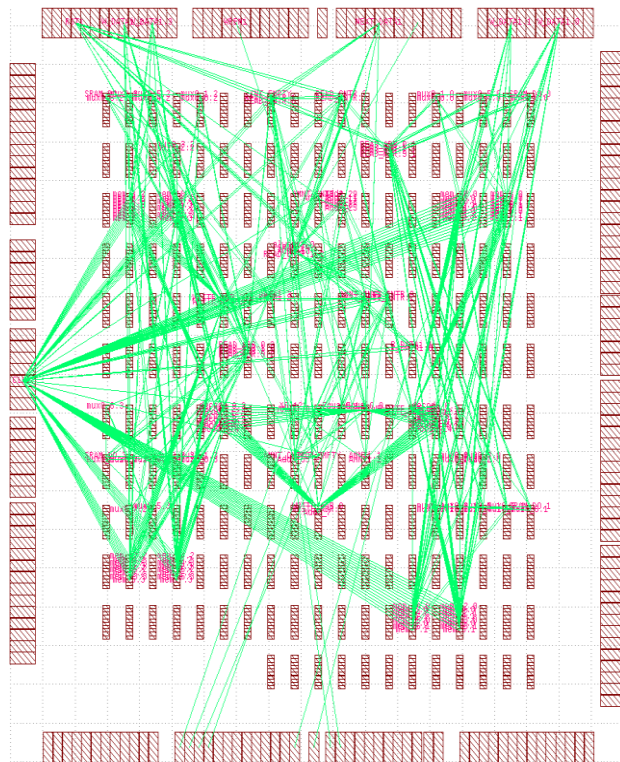


Рис. 3. Синхронный блок FIFO 16x4b. Ручное размещение и автоматическая трассировка

Разработанные программные средства на основе управляемого Tcl-интерфейса достигают эффективности проектирования схем на ПЛИС за счет более компактного представления, сокращения количества компонентов и оптимального использования ресурсов трассировки.

IV. МОДИФИЦИРОВАННЫЙ АЛГОРИТМ PATHFINDER

A. Теоретическая часть

В представленном выше маршруте для реализации размещения и трассировки используется модуль XC, разработанный в ИППМ РАН. За основу алгоритма автоматической трассировки межсоединений для ПЛИС взят алгоритм PathFinder.

В этом алгоритме не используются специфические особенности определенной архитектуры, так что он

может быть применен к ПЛИС с любой архитектурой при минимальных затратах времени и ресурсов. После того как разработчиком выбрана базовая архитектура, трассировщик настраивается под неё путём задания шаблона, который описывает ресурсы трассировки и взаимозаменяемость контактов.

В отличие от стандартного алгоритма PathFinder, где ресурсы трассировки и их соединения представлены направленным графом, модифицированный алгоритм представлен частично направленным графом $G = (V, E)$ [14, 15]. Неориентированные ребра – для мультиплексоров, ориентированные – для буферов. Набор вершин V соответствует электрическим узлам или проводам в архитектуре ПЛИС, а рёбра E – переключателям, которые соединяют эти узлы [16].

С каждым узлом n в графе ассоциирована базовая стоимость b_n , которая представляет собой относительную стоимость использования данного узла. Эта стоимость, в простейшем случае, пропорциональна длине линии, хотя могут быть учтены и другие факторы, такие как различные паразитные емкости и количество входов/выходов ячейки в реальной ПЛИС. Каждый узел также имеет задержку D_n , которая может быть учтена в алгоритме.

PathFinder уравнивает цели трассируемости и производительности работы проектируемого устройства. Трассируемость достигается путём согласования сигналов относительно ресурса трассировки, тем самым определяется какой сигнал нуждается в данном ресурсе больше всего. Более критичный сигнал имеет большие права в борьбе за ресурс.

Основная задача – распределить ресурсы так, чтобы передать все сигналы. Другая цель – свести к минимуму задержку, что требует использования минимальных маршрутов по задержке для сигналов. Таким образом, решение всей проблемы трассировки требует одновременного решения двух взаимоисключающих задач. PathFinder – итерационный алгоритм, который уравнивает эти задачи. На начальном этапе сигналам позволено использовать общие ресурсы трассировки, но впоследствии должны вестись согласования, чтобы определить какой сигнал нуждается в ресурсе больше всего. Особенность подхода заключается в постепенной корректировке стоимости ресурсов трассировки для достижения оптимального распределения ресурсов.

В течение первой итерации глобальной трассировки нет стоимости для разделения ресурсов трассировки, и они могут использоваться более чем одним сигналом. Во время последующих итераций штраф постепенно увеличивается, и сигналы согласовывают между собой ресурсы трассировки. Сигналы могут совместно использовать ресурс трассировки в том случае, если все альтернативные

маршруты используют более востребованные ресурсы.

Согласно алгоритму согласования перегруженности, стоимость использования ресурса трассировки определяется как:

$$\begin{aligned} c_n &= p_n \cdot (b_n + h_n), \\ p_n &= 1 + (N_{\text{iter}} - 1) \cdot F_p \cdot N_{\text{path}}, \\ h_n &= h_n + F_h \cdot (N_{\text{path}} - 1), \end{aligned} \quad (1)$$

где c_n – стоимость узла, b_n – базовая стоимость, определенная графом, p_n – функция числа итераций и числа других сигналов, которые в данный момент используют узел n , h_n – отвечает за историю перегруженности узла n , N_{iter} – номер итерации, N_{path} – число путей, проходящих через узел n .

Если нет других сигналов, которые используют узел n , то $p_n = 1$, иначе $p_n = \infty$. В алгоритме согласования заборов первоначально p_n устанавливается в 1, и все сигналы трассируются. Это позволяет оттрассировать каждый сигнал независимо от маршрутов других сигналов. Затем стоимость ресурсов, которые были разделены, возрастает, и все маршруты разрываются и по очереди перетрассируются. Стоимость разделения возрастает на каждой итерации, и итерационный процесс продолжается до тех пор, пока все сигналы не будут успешно оттрассированы. Идея заключается в том, что стоимость перегруженного узла будет возрастать, и те сигналы, у которых есть альтернативные пути, освободят путь для сигнала, который нуждается в этом ресурсе больше других сигналов.

Слагаемое h_n «помнит», что в течение предыдущих итераций узел n был перегружен. История обновляется после каждой итерации трассировки. Любой узел, разделённый несколькими сигналами, имеет свою историю h_n . Цель h_n состоит в том, чтобы постоянно увеличивать стоимость использования перегруженных узлов так, чтобы предпринимались маршруты через другие узлы. Без этого слагаемого, как только сигналы перестают разделять общий узел, его стоимость опускается до базовой стоимости и снова становится привлекательной. Это приводит к колебаниям, сигналы переключаются между узлами, не решая проблему перегруженности.

Параметр h_n обновляется всякий раз, когда возникает перегруженность. Размер параметра F_h , на который увеличивается h_n и то, как он зависит от количества сигналов, разделяющих общий ресурс, можно настраивать. Если F_h слишком мал, то может потребоваться много итераций для устранения перегрузки; если он слишком велик, то некоторые решения невозможно найти. Кроме того, крайне важно отношение p_n к h_n . Например, это может позволить параметру истории решить проблему перегруженности прежде, чем применять для решения параметр p_n [17].

В. Экспериментальная часть

В качестве примера представлены данные реализации модифицированного алгоритма PathFinder для схем из набора ISCAS85 на ПЛИС 5510XC3T. Проведено сравнение полученных результатов, и подобраны оптимальные параметры для трассировки цифровых схем в архитектуре данной ПЛИС.

Для достижения трассировки базовой схемы c432 изменялся параметр F_h , который определяет зависимость стоимости использования узла от числа путей, проходящих через данный узел на текущей итерации. Вместе с этим изменялся параметр $\max_c$, который отвечает за максимально возможную суммарную стоимость пути. Для всех возможных сочетаний параметров получено число итераций трассировки и число неразведённых цепей. Результаты представлены в табл. 1.

Таблица 1

Результаты трассировки для c432

F_p	$\max_c$	Число итераций	Неразведённые цепи
0.9	100	76	3
1.2	100	66	1
1.5	100	54	3
0.9	200	148	0
1.2	200	106	0
1.5	200	97	1
0.9	300	171	0
1.2	300	145	1
1.5	300	135	0

Аналогичная таблица получена для усложнённой схемы c1355 – табл. 2.

Таблица 2

Результаты трассировки для c1355

F_p	$\max_c$	Число итераций	Неразведённые цепи
0.9	100	75	3
1.2	100	64	2
1.5	100	53	2
0.9	200	149	0
1.2	200	119	0
1.5	200	96	2
0.9	300	191	1
1.2	300	145	1
1.5	300	142	2

На основе сравнения графиков для схем c432 и c1355 сделан вывод о наиболее приемлемых параметрах для реализации трассировки. Это параметры $\max_c = 200$, $F_p = 1.2$, при которых отсутствуют неразведённые цепи, и результат достигается за минимальное число итераций.

Для достижения лучших результатов трассировки также необходимо подобрать оптимальное значение параметра F_h , который отвечает за историю перегруженности узла на предыдущих итерациях.

При фиксированных ранее подобранных параметрах $\max_c = 200$, $F_p = 1.2$ изменялся параметр F_h . В качестве выходных данных в табл. 3 представлены число итераций и число неразведённых цепей.

Таблица 3

Зависимость числа итераций и неразведённых цепей от параметра F_h для схемы c432

F_h	Число итераций	Неразведённые цепи
0.1	280	0
0.2	164	0
0.3	106	0
0.4	102	1
0.5	98	1
0.6	91	1
0.7	80	1

Результаты аналогичного эксперимента для усложнённой схемы c1355 представлены в табл. 4.

Таблица 4

Зависимость числа итераций и неразведённых цепей от параметра F_h для схемы c1355

F_h	Число итераций	Неразведённые цепи
0.1	345	0
0.2	201	1
0.3	191	0
0.4	111	1
0.5	100	1
0.6	90	2
0.7	85	2

На основе результатов по двум представленным таблицам сделан вывод о том, что наиболее выгодным для достижения поставленного результата является параметр $F_h = 0.3$.

В ходе проведённых экспериментов получен набор параметров для реализации оптимального размещения и трассировки для цифровых схем в архитектуре ПЛИС 5510XC3T. Это параметры $\max_c = 200$, $F_p = 1.2$, $F_h = 0.3$.

В. ЗАКЛЮЧЕНИЕ

Целью выполнения данной работы являлась разработка маршрута проектирования и методов конфигурирования программируемых логических

интегральных схем, разработанных и изготовленных на отечественных предприятиях.

В ходе работы разработаны способы и методы решения задач размещения элементов, трассировки межсоединений для ПЛИС 5510XC1T, 5510XC2T, 5510XC3T.

Представленный маршрут, в отличие от существующих, обеспечивает возможность проектирования ИС в базисе отечественных ПЛИС с ограниченными трассировочными ресурсами.

ЛИТЕРАТУРА

- [1] MAX II Device Handbook // Altera Corporation [Электронный ресурс]. Системные требования: Adobe Acrobat Reader. Режим доступа: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/max2/max2_mii5v1.pdf (дата обращения: 20.12.2015).
- [2] Quartus II Version 7.2 Handbook // [Электронный ресурс]. Системные требования: Adobe Acrobat Reader. – Режим доступа: http://www.cs.columbia.edu/~sedwards/classes/2008/4840/qts_qii5v1.pdf (дата обращения: 10.01.2016).
- [3] Гаврилов С.В., Иванова Г.А. Методы мультиинтервального анализа быстродействия нанoeлектронных схем на логическом уровне // «Электронная техника. Серия 3. Микроэлектроника». 2015. Выпуск № 3 (159) С. 11-18.
- [4] Гаврилов С.В., Рыжова Д.И. Алгоритм оценки пикового тока на логическом уровне проектирования на основе анализа распространения логических корреляций в схеме // Вестник Рязанского государственного радиотехнического университета. 2015. №2 (Выпуск 52). С. 56-64.
- [5] Гаврилов С.В., Иванова Г.А. Анализ быстродействия сложных цифровых схем с учетом неопределенности технологических и схемных параметров // Вестник Рязанского государственного радиотехнического университета. 2015. (Выпуск 53). С. 29-35.
- [6] Korshunov A.V., Volobuev P.S. Active-mode leakage power optimization using state-preserving techniques // Design & Test Symposium (EWDTS), 2014 East-West, 2014, P. 1-4.
- [7] Гаврилов С.В., Гудкова О.Н., Пирютина Г.А. Метод анализа быстродействия вентиля с учетом одновременного переключения входов // V Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и нанoeлектронных систем – 2012»: сб. научн. тр. / под общей ред. А.Л.Стемпковского. М.: ИППМРАН, 2012. С. 119-124.
- [8] Гаврилов С.В., Иванова Г.А., Рыжова Д.И., Стемпковский А.Л. Методы повышения надежности комбинационных микроэлектронных схем на основе мультиинтервального анализа быстродействия // «Системы высокой доступности». №4. 2015. С. 69-76.
- [9] C. Ebeling, L. McMurchie, S. Hauck, S. Burns Placement and Routing Tools for the Triptych FPGA [Текст] // IEEE Transactions on VLSI Systems, Vol. 3, No. 4. – 1995. – С. 473–482.
- [10] Б. Уэлш, К. Джонс, Практическое программирование на Tcl и Tk [Текст] // 4-е издание.: Пер. с англ. – М.: Вильямс, 2004. – 1136 с
- [11] Свидетельство на программу NanoLib-Sim // Режим доступа: <http://www.ippm.ru/infoIPPM/index.php?page=vpaper&code=D606&ls=ru> (дата обращения: 01.02.2016).
- [12] Борисенко Н. Подходы к организации унифицированного ряда синтезируемых моделей буферов FIFO, реализуемых в различных семействах программируемой логики. // Компоненты и технологии 12.2012, 01.2013, С.152-159, С.6-10.
- [13] Угрюмов Е.П. Цифровая схемотехника 2-е изд. 2007, С.244-247.
- [14] R. Nair. A simple yet effective technique for global wiring [Текст] // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 6(2), – 1987. – С. 165–172.
- [15] V. Betz, J. Rose. VPR: A new packing, placement and routing tool for FPGA research. [Текст] // Proceedings of the Seventh International Workshop on Field-Programmable Logic and Applications. Springer-Verlag, – 1997. – С. 213–222.
- [16] E.W. Dijkstra. A note on two problems in connexion with graphs [Текст] // Numerische Mathematik, 1. – 1959. – С. 269–271.
- [17] L. McMurchie, C. Ebeling PathFinder: A Negotiation-Based Performance-Driven Router for FPGAs [Текст] // ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. – 1995. – С. 111–117.

Improving the Efficiency of the Integrated Circuit Design on FPGA with Limited Routing Resources

T.V. Garbulina, V. M. Khvatov, O.V. Ljalinskaja

Institute for Design Problems in Microelectronics (IPPM RAS),

garbulina_t@ippm.ru, olgaly@ippm.ru, khvatov_v@ippm.ru

Keywords – CAD, TCL, routing, placement, PathFinder.

ABSTRACT

At the present time, domestic developments are coming to replace foreign field programmable gate arrays (FPGAs). Existing computer-aided design (CAD) support the development of specific FPGA manufacturers (Xilinx, Altera, Actel) and do not provide the opportunity to design FPGA with new architectures. The purpose of this work is the development of the design flow, applicable to any FPGA architecture, and improving the efficiency of the integrated circuit design on domestic FPGA using specially developed linguistic tools on the basis of TCL language. The main result of the done work is the development and improvement of the software tools for the optimal placement of device components and routing their interconnections by modified PathFinder algorithm. Simulation results show the efficiency of the developed algorithms.

REFERENCES

- [1] MAX II Device Handbook. Altera Corporation System. requirements: Adobe Acrobat Reader. Available at: https://www.altera.com/en_US/pdfs/literature/hb/max2/max2_mii5v1.pdf (accessed: 20.12.2015).
- [2] Quartus II Version 7.2 Handbook System requirements: Adobe Acrobat Reader. Available at: http://www.cs.columbia.edu/~sedwards/classes/2008/4840/qts_qii5v1.pdf (accessed: 10.01.2016).
- [3] Gavrilov S.V., Ivanova G.A. Methods of multi-interval analysis speed of nanoelectronic circuits on a logical level. *Jelektronnaja tehnika. Serija 3. Mikroelektronika*, 2015, no. 3 (159) pp. 11-18. (in Russian)
- [4] Gavrilov S.V., Ryzhova D.I. The algorithm estimates the peak current at logic level design based on the analysis of correlations in propagation logic circuit. *Vestnik Rjazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*, 2015, no. 52. pp. 56-64. (in Russian)
- [5] Gavrilov S.V., Ivanova G.A. Analysis of the performance of complex digital circuits, taking into account the uncertainty of technology and circuit parameters. *Vestnik Rjazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*, 2015, no. 53. pp. 29-35. (in Russian)
- [6] Korshunov A.V., Volobuev P.S. Active-mode leakage power optimization using state-preserving techniques // *Design & Test Symposium (EWDTS)*, 2014 East-West, 2014, P. 1-4.
- [7] Gavrilov S.V., Gudkova O.N., Pirjutina G.A. Valve performance analysis method based on simultaneous switching inputs. *Vserossijskaja nauchno-tehnicheskaja konferencija «Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem – 2012»*: sb. nauchn. tr. / pod obshhej red. A.L. Stempkovskogo. M.: IPPM RAS, 2012. pp. 119-124. (in Russian)
- [8] Gavrilov S.V., Ivanova G.A., Ryzhova D.I., Stempkovskij A.L. Methods to improve the reliability of the combination of microelectronic circuits on the basis of a multi interval analysis speed. *Sistemy vysokoj dostupnosti*, 2015, no. 4. pp. 69-76. (in Russian)
- [9] C. Ebeling, L. McMurchie, S. Hauck, S. Burns Placement and Routing Tools for the Triptych FPGA. *IEEE Transactions on VLSI Systems*, Vol. 3, No. 4. – 1995. – pp. 473–482.
- [10] B. Ujelsh, K. Dzhons Practical Programming in Tcl and Tk. 4th ed. –M.: Vil'jams, 2004, -1136 p. (in Russian)
- [11] Certificate for the program NanoLib-Sim. Available at: <http://www.ippm.ru/infoIPPM/index.php?page=vpaper&code=D606&ls=ru>.
- [12] Borisenko N. The approaches to the organization of a unified series of synthesized models FIFO buffers implemented in different families of programmable logic. // *Components and Technologies* 12.2012, 01.2013, pp.152-159, pp.6-10.
- [13] Ugrjumov E.P. Digital circuitry 2th ed. 2007 pp. 244-247.
- [14] R. Nair. A simple yet effective technique for global wiring // *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 6(2), – 1987. pp. 165–172.
- [15] V. Betz, J. Rose. VPR: A new packing, placement and routing tool for FPGA research. // *Proceedings of the Seventh International Workshop on Field-Programmable Logic and Applications*. Springer-Verlag, – 1997. – pp. 213–222.
- [16] E.W. Dijkstra. A note on two problems in connexion with graphs // *Numerische Mathematik*, 1. –1959. – pp. 269–271.
- [17] L. McMurchie, C. Ebeling PathFinder: A Negotiation-Based Performance-Driven Router for FPGAs // *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. – 1995. – pp. 111–117.