

Разработка и сравнительный анализ методов начального размещения на ПЛИС

П.И. Фролова, Р.Ж. Чочаев

Институт проблем проектирования микроэлектроники РАН, г. Москва

frolova_p@ippm.ru, chochaev_r@ippm.ru

Аннотация — Требования к алгоритмам в автоматизированном маршруте проектирования программируемых интегральных схем постоянно растут, и вместе с этим усиливается зависимость каждого последующего этапа проектирования от результатов предшествующего ему этапа. В данной работе мы проводим сравнительный анализ различных способов генерации начального размещения для островных ПЛИС с последующей оптимизацией решения методом моделирования отжига. Результаты исследования показали, что начальное размещение может влиять на время работы метода моделирования отжига, оптимальность размещения и, как следствие, на быстродействие схемы.

Ключевые слова — ПЛИС, начальное размещение, система автоматизированного проектирования (САПР).

I. ВВЕДЕНИЕ

Программируемая логическая интегральная схема (ПЛИС) – это ИС, которая может быть перепрограммирована для конкретных целей конечного пользователя [1]. ПЛИС можно использовать для прототипирования разрабатываемых схем, а также для прикладных целей во многих отраслях промышленности.

В современном маршруте автоматизированного проектирования усиливается взаимосвязь между отдельными этапами, а сами этапы становятся сложнее и зачастую имеют иерархическую структуру [2-4]. Это, наряду с постоянным усложнением технологических процессов, вынуждает исследователей искать концептуально новые решения и переосмысливать имеющиеся методы.

Одним из важнейших этапов в маршруте проектирования на ПЛИС является этап размещения [5], на котором элементам проектируемой схемы назначаются конкретные посадочные места. Современные алгоритмы размещения включают в себя несколько этапов, среди которых можно выделить два ключевых: генерацию начального решения и его оптимизацию по некоторому критерию. Данная работа посвящена исследованию алгоритмов генерации начального размещения с последующей оптимизацией длины цепей методом моделирования отжига в ПЛИС островной архитектуры. Были реализованы пять алгоритмов начального размещения: случайный,

силовой, ранжированный от входов/выходов, построчный. После генерации начального размещения выполнялась оптимизация методом имитации отжига и трассировкой. В качестве критериев сравнения алгоритмов использовались время работы, быстродействие и длина цепей.

1) Данная работа состоит из пяти разделов. Во II разделе приведён краткий обзор существующих алгоритмов размещения и описаны основные алгоритмы начального размещения. Раздел III посвящен описанию островной архитектуры ПЛИС, используемой в данной работе. В IV разделе приводится описание исследуемых в рамках работы методов. В V разделе представлены экспериментальные результаты и подведены итоги исследования. Итоговые выводы приведены в разделе VI.

II. КРАТКИЙ ОБЗОР СУЩЕСТВУЮЩИХ АЛГОРИТМОВ РАЗМЕЩЕНИЯ И РОЛЬ НАЧАЛЬНОГО РАЗМЕЩЕНИЯ

Существующие алгоритмы размещения можно классифицировать множеством способов [6]. Одна из наиболее популярных классификаций делит всё множество алгоритмов размещения на ПЛИС на три большие группы: эвристические, аналитические и алгоритмы на основе декомпозиции.

К эвристическим алгоритмам можно отнести алгоритмы на основе метода моделирования отжига [7], генетические алгоритмы [8]. Наиболее популярными являются алгоритмы на основе метода моделирования отжига, благодаря своей простоте и высокой эффективности. Несомненным достоинством также является асимптотическая сходимость данного метода [9]. Однако, с ростом размерности задачи наблюдается резкий рост времени, которое требуется для поиска оптимального решения.

Сегодня для ПЛИС, содержащих более 500 тыс. конфигурируемых логических блоков, используются аналитические алгоритмы [10, 11], в которых задача размещения, представленная в аналитическом виде, решается численными методами. Для того, чтобы избежать стягивания элементов, используются различные методы локальной оптимизации и легализации. Главным преимуществом аналитических алгоритмов является высокое быстродействие. К недостаткам можно отнести сложность реализации и

необходимость в эффективных методах легализации и оптимизации.

Методы на основе декомпозиции [12,13] выполняют размещение путем последовательного разбиения области размещения и пользовательской схемы на части с последующим назначением каждой подсхемы на соответствующую подобласть размещения. Данные алгоритмы хорошо подходят для размещения в иерархических ПЛИС с симметричной структурой коммутаций. Ключевым недостатком данных алгоритмов является сложность учета дополнительных критериев (например, задержек цепей).

Алгоритмы размещения назначают логическим элементам места на ПЛИС таким образом, чтобы была обеспечена полная трассируемость пользовательских схем, отсутствовали перекрытия и выполнялись временные ограничения, для чего требуется контролировать суммарную длину цепей.

В современном маршруте проектирования к базовой задаче этапа размещения, обозначенной выше, может добавляться целый спектр второстепенных. Очевидная второстепенная задача – обеспечение высокого быстродействия, также в размещении может оптимизироваться трассируемость и/или энергопотребление схемы.

Роль начального размещения заключается в том, чтобы, не имея конкретных данных о физических межсоединениях, предсказать наиболее критические пути в схеме, определить элементы с высокой степенью связности и назначить им оптимальные позиции [14].

Для оценки оптимальности вводится целевая функция. Например, в работе [15] оценивается общая нормализованная задержка цепей для путей с учетом временных и геометрических ограничений. Нормализованная общая задержка пути вычисляется с учетом резерва задержки на данном пути. На оценочную функцию накладывается ряд ограничений, которые направлены на контроль за выполнением временных ограничений, легальностью и занимаемой площадью. В более современных методах [16-17] предлагается использовать более сложные оценочные функции с учетом важности цепей, основанной на сравнении предельно допустимой и фактической задержки в каждой конкретной цепи, что позволяет осуществлять строгий контроль за возникающими критическими путями. Также известны методы, использующие многофакторные оценки и учитывающие кроме задержек цепей связность, производительность и температурный режим [18]. В данной работе используется целевая функция, учитывающая виртуальную длину цепей, использующая модель полупериметра охватывающего прямоугольника (*Half-Perimeter Wire Length, HPWL*).

III. ОПИСАНИЕ АРХИТЕКТУРЫ ПЛИС

В данной работе использовалась ПЛИС с островной архитектурой [1]. Базовым элементом является конфигурируемый логический блок (ЛБ), который

состоит из двух логических ячеек (ЛЯ). Каждая ЛЯ состоит из 3-х входовой ячейки асинхронной памяти (*look-up table, LUT*) и триггера. ЛБ собраны в один большой массив (23 строк и 39 столбцов). По периметру расположены ячейки ввода-вывода.

Перед описанием реализованных в рамках работы алгоритмов начального размещения, рассмотрим подробнее архитектуру тактового дерева, так как она оказывает влияние на размещение. На рис. 1 видно, что в используемой ПЛИС имеются 4 источника тактового сигнала, которые условно делят всю площадь ПЛИС на 4 домена.

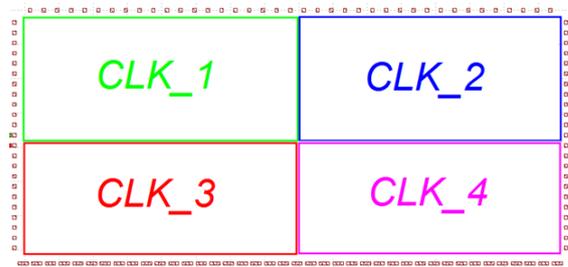


Рис. 1. Домены синхросигналов в ПЛИС

Таким образом, в алгоритме начального размещения должен присутствовать механизм определения допустимого размещения для каждого логического элемента. Это может быть осуществлено в виде отдельного подготовительного этапа, либо непосредственно в момент размещения конкретной логической ячейки с помощью дополнительных проверок.

Если в схеме отсутствуют тактовые сигналы или присутствует один тактовый сигнал, то ограничения на размещение элементов отсутствуют, так как ЛЯ могут быть подключены не более чем к одному источнику тактового сигнала.

Если в схеме используются два тактовых сигнала, то на размещение триггеров вводятся ограничения в зависимости от расположения источников тактового сигнала. В случае, когда они расположены напротив друг друга в одном ряду, то площадь ПЛИС делится на 2 части, одна из которых содержит домены CLK_1 и CLK_3 (рис. 1), а другая – домены CLK_2 и CLK_4. Таким образом, триггеры, подключенные к источнику тактового сигнала на левой половине кристалла, могут размещаться только в доменах CLK_1 и CLK_3. Если источники тактового сигнала расположены в одном столбце или в разных строках, то ПЛИС делится на следующие части: CLK_1 + CLK_2 и CLK_3 + CLK_4.

Если в схеме содержатся три источника тактовых сигналов, то остается один неиспользуемый домен. Он может быть присоединен только к смежным доменам. Приоритетом будет обладать тот тактовый блок, который связан с наибольшим количеством логических элементов.

Если в схеме используются все четыре доступных тактовых блока, то ПЛИС делится на такое же количество частей. Триггеры, связанные с верхним

левым тактовым блоком, размещаются в CLK_1, связанные с правым нижним – в CLK_4 и т.д.

IV. ОПИСАНИЕ ИСПОЛЪЗУЕМЫХ АЛГОРИТМОВ

Задача начального размещения состоит в том, чтобы за небольшой период времени подготовить предварительное размещение логических элементов, для последующей оптимизации, например, методом моделирования отжига. В данной работе мы предлагаем рассмотреть несколько подходов к начальному размещению, которые будут подробно описаны далее.

A. Силовое

Алгоритмы размещения с использованием сил притяжения либо отталкивания известны достаточно давно [19]. В силовом алгоритме связи между логическими элементами представляются в виде сил притяжения, которые определяют итоговое расположение ЛЭ. Чем выше сила притяжения, тем ближе будут размещены связанные элементы. За счет этого сокращается суммарная длина цепей пользовательской схемы и оптимизируется начальное размещение элементов. Оптимальное начальное решение повышает сходимость последующей оптимизации размещения методом моделирования отжига.

В данной работе силовой алгоритм, после определения допустимых областей размещения, вычисляется центр масс относительно фиксированных ЛЭ и ячеек ввода-вывода по следующей формуле:

$$x_0 = \frac{\sum_{i=1}^n x_i}{n}, y_0 = \frac{\sum_{i=1}^n y_i}{n},$$

где (x_0, y_0) – координаты центра масс, (x_i, y_i) – координата i -го фиксированного ЛЭ и/или ячейки ввода-вывода, n – количество фиксированных ЛЭ и ячеек ввода-вывода

После вычисления начального центра масс для каждого элемента вычисляется центр масс, относительно связанных с ним и размещенных ячеек. Затем выполняется поиск ближайшей посадочной площадки к данному центру масс, где и размещается логический элемент. Размещение ЛЭ выполняется последовательно. Первыми размещаются логические элементы, связанные с ячейками ввода-вывода. Затем размещаются оставшиеся логические элементы блоки.

B. Ранжированное

В качестве одного из альтернативных вариантов реализован подход, учитывающий длину цепей между элементами и положение элемента в конкретном пути относительно входных и выходных ячеек. Размещение элементов будет осуществляться в 3 этапа: размещение элементов, связанных с входными ячейками; размещение элементов, связанных с выходными ячейками; силовое итерационное размещение оставшихся элементов в схеме.

Наиболее сложной и потенциально трудоемкой на больших схемах задачей в данном подходе является процедура оценки удаленности логического элемента

от ячеек ввода-вывода, также называемая ранжированием элементов. Для сокращения времени проведения данной процедуры используется обход по графу элементов схемы, основанный на методе поиска в ширину по графу. Важная особенность предложенного метода ранжирования состоит в том, что логическому элементу назначается наименьший возможный ранг. Данная особенность разобрана на рис. 2.

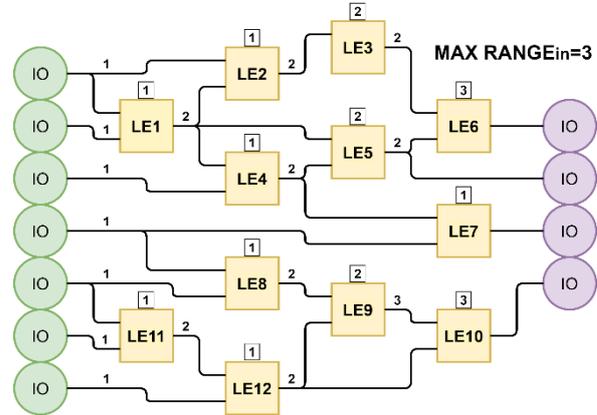


Рис. 2. Назначение ранга логического элемента

Например, у LE2, LE4, LE7, LE8 и LE12 один из контактов соединяется с входной ячейкой, что автоматически устанавливает ранг LE равным 1, несмотря на соединения другого контакта. В общем случае ранг задается следующими формулами:

$$r_{in}(v) = \min_{u \in fanin(v)} r_{in}(u) + 1,$$

$$r_{out}(v) = \min_{u \in fanout(v)} r_{out}(u) + 1,$$

где r_{in} – входной ранг, r_{out} – выходной ранг, v – это текущий ЛЭ в пути, u – множество предыдущих ЛЭ в пути, для ранжирования от входных и выходных ячеек соответственно. Распространяя ранги данным образом, для приведенного на рисунке примера максимальный ранг от входных ячеек равен 3.

Ранжирование от выходных ячеек выполняется аналогичным образом. К выходным ячейкам подключены выходные контакты логических элементов LE5, LE6, LE7, LE10, их выходной ранг равен 1. Элементы LE3, LE9, LE4, LE12, LE1 – следующие ближайшие к выходным ячейкам элементы, их ранг равен 2. По самым отдаленным от выходных ячеек элементам определяется максимальный выходной ранг в схеме. Для приведенного примера он равен максимальному рангу от входных ячеек.

Элементы с максимальным выходным рангом обладают минимальным входным рангом, при этом сам минимальный входной ранг не гарантирует, что элемент будет иметь максимальный выходной ранг. LE1 на рисунке 2 имеет входной и выходной ранг 1 и 2 соответственно. Именно поэтому ранжирование разделено на 2 операции.

После процедуры ранжирования отбираются ЛЭ с рангом по входу либо выходу равным 1, которые

размещаются в первую очередь. Далее в зависимости от выбранного режима проводится итерационное силовое размещение элементов с рангами выше 1. Режим позволяет управлять количеством и порядком размещения элементов на каждой итерации. Например, режим «от входа» размещает в одной итерации только те элементы, входной ранг которых равен счетчику количества итераций. Количество итераций в данном режиме ограничивается максимальным входным рангом. Режим «от выходов», соответственно, выбирает и размещает элементы на основании ранга от выхода, игнорируя ранг от входа и имеет количество итераций равное максимальному рангу от выхода.

С. Случайное

В алгоритме случайного размещения, для каждой размещаемого ЛЭ из списка соединений итерационно генерируются случайные координаты посадочных мест, которые затем проверяются на легальность (например, для схем с триггерами проверяется возможность размещения ЛЭ в данном тактовом домене). После успешной проверки логический элемент назначается на данное посадочное место, в противном случае генерируется новая пара координат. Для соблюдения баланса между поиском легальных координат и временем выполнения алгоритма вводится ограничение на количество итераций.

Д. Заполнение по строкам

В данном алгоритме размещение выполняется последовательно по строкам, сверху вниз. Размещение

элементов по строкам позволяет быстро сгенерировать начальное размещение для последующей оптимизации методом моделирования отжига. Аналогично методу случайного размещения, перед подтверждением назначения на посадочное место, проводится проверка на легальность. Если проверка завершается неудачно, то последовательно ищется легальное посадочное место во всех рядах и столбцах.

V. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

В данном разделе приведены результаты проведенного исследования. Для тестов используются схемы из наборов ISCAS85, ISCAS89 [20, 21] и ряд промышленных тестовых схем (ADC, ADC_control, test_3, test_4), которые поместились в ПЛИС.

В ходе экспериментов для каждой схемы запускался полный маршрут проектирования. Синтез схем выполнялся с помощью программы Yosys [22]. После начального размещения выполнялась оптимизация длины цепей методом моделирования отжига и трассировка.

Для всех схем было выставлено одинаковое временное ограничение равное 500 нс. Кроме полученных слеков (табл. 1) сравниваются время работы алгоритма (табл. 2), а также начальное и конечное значение целевой функции метода моделирования отжига (табл. 3).

Таблица 1

Сравнение быстродействия разных методов начального размещения

Название схемы	Количество ЛЭ (все/ранжированные)	Слек (наихудший)				
		Силовое	Ранжированное от входа	Ранжированное от выхода	Случайное	По строкам
ADC	101/64	418	420	420	367	422
ADC_control	199/55	435	435	430	430	390
c499	94/23	300	300	300	298	275
c1355	92/22	257	257	257	297	287
c1908	109/31	0	286	293	268	229
c3540	407/94	109	144	129	137	104
c6288	547/235	31	31	3	11	57
s953	152/55	303	303	289	311	266
s1488	261/167	233	233	168	207	206
s5378	399/242	179	42	42	-	-
s9234	287/226	235	181	181	217	193
test_3	559/430	93	105	82	87	76
test_4	568/348	132	115	101	158	199
	Среднее	210	219	207	214*	208*

Таблица 2

Сравнение времени работы разных методов размещения

Название схемы	Время работы, с				
	Силовое	Ранжированное от входа	Ранжированное от выхода	Случайное	По строкам
ADC	6	5	7	5	4
ADC_control	12	16	15	43	14
c499	5	4	5	4	5
c1355	4	5	4	4	4
c1908	5	6	5	6	7
c3540	35	35	36	37	37
c6288	60	63	63	67	72
s953	9	9	9	9	9
s1488	81	69	68	76	81
s5378	167	152	150	133	1097
s9234	23	22	23	25	83
test_3	64	64	63	63	217
test_4	60	219	209	227	239
Среднее	44	55	54	58	155

Экспериментальные результаты показывают, что характер начального размещения может оказывать серьезное влияние на последующий этап оптимизации и, как следствие, на трассируемость и быстродействие схемы. Некорректно подобранный метод начального размещения может даже привести к невозможности трассировки схемы, как это произошло со схемой s5378 при использовании методов случайного и построчного размещения.

Из табл. 1 видно, что первые 3 метода позволяют получить большой запас времени для критического пути для большинства схем. Вероятно, такой результат обусловлен механизмом размещения ЛЭ, связанных с ячейками ввода-вывода, которые занимают большую долю в общем числе ЛЭ. Данное предположение косвенно подтверждается схемами c6288 и test_4, где доля связанных с входами/выходами ЛЭ ниже, а запас по времени выше для случайного и построчного размещения.

Примечательно, что для некоторых схем (c1355, s923) наилучший результат получен в случайном методе размещения. Этот метод имеет достаточно простую программную реализацию генерации начального размещения и достаточно широко используется в САПР [7].

Если рассматривать усредненные данные о запасе времени для промоделированных схем, наилучший результат обеспечивает метод ранжирования от входных ячеек. Ранжирование позволяет расположить элементы более направленно от входных ячеек к выходным.

Вторым по средней эффективности стал случайный метод. У этого метода самое низкое число схем с наименьшим показателем слэка. В то же время данный метод не позволил добиться полной трассировки всех тестовых схем.

Метод ранжирования от выходных ячеек и построчный метод показали очень близкий усредненный результат и оба отличились наличием как наилучшего слэка для схемы (c499, c1908 и ADC, c6288, test_4 соответственно), так и наихудшего.

Время работы разных методов приблизительно одинаково в рамках одной схемы (таблица 2), таким образом, данная характеристика не должна быть решающим фактором в вопросе выбора подходящего метода.

Полученные результаты из таблицы 3 показывают, что начальное значение целевой функции перед оптимизацией может значительно изменяться в зависимости от выбранного метода начального размещения, при этом конечные значения целевой функции достаточно близки.

Более высокие значения целевой функции характерны для случайного и построчного методов размещения. Это объясняется большей «хаотичностью» данных способов. Учет связности логических элементов в силовом и ранжированных методах, позволяет создавать более компактные, расположенные между задействованными ячейками ввода-вывода структуры на этапе начального размещения, в то время как случайное размещение занимает всю доступную площадь кристалла, значительно увеличивая оценочные длины цепей, а построчное, в свою очередь, сконцентрировано в верхних строках, что так же увеличивает суммарную длину цепей перед оптимизацией.

VI. ЗАКЛЮЧЕНИЕ

В рамках данной работы было проведено исследование различных методов начального размещения элементов на ПЛИС. Рассмотрено пять методов: случайный, силовой, ранжированный от входов/выходов, построчный. Проведено измерение быстродействия, времени работы и суммарной длины цепи до и после оптимизации методом моделирования отжига. Сравнение методов по данным критериям показало, что способ начального размещения может оказывать сильное влияние на итоговое быстродействие схемы при близких временах работы и суммарной длины цепей.

Сравнение изменения целевой функции детального размещения для разных методов

Название схемы	Целевая функция (до и после оптимизации методом моделирования отжига)									
	Силовое		Ранжированное от входа		Ранжированное от выхода		Случайное		По строкам	
	до	после	до	после	до	после	до	после	до	после
ADC	238	136	238	136	237	136	856	136	642	136
ADC_control	720	314	726	313	726	313	1485	313	1215	313
c499	511	426	511	426	511	426	1003	425	996	426
c1355	531	425	531	425	531	425	1078	425	10003	425
c1908	521	390	521	390	519	391	1207	391	988	391
c3540	1840	795	1807	796	1837	795	3476	792	2850	794
c6288	3329	871	3329	871	3395	870	5491	867	4773	868
s953	675	400	675	400	673	399	1489	397	1296	400
s1488	1007	491	1007	491	1011	490	2123	490	1595	490
s5378	1831	964	1826	964	1826	964	3855	961	3134	963
s9234	1152	639	1154	634	1154	634	2896	635	2345	637
test_3	2611	1048	2547	1045	2572	1047	4448	1042	3690	1045
test_4	2517	901	2390	894	2340	893	5152	894	4181	884
Среднее	1345	600	1328	599	1333	599	2658	597	2208	598

ЛИТЕРАТУРА

- [1] Hauck S., DeHon A. Reconfigurable Computing: The Theory and Practice of Fpga-Based Computation San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 2007.
- [2] Гаврилов С.В., Железников Д.А., Заплетина М.А., Хватов В.М., Чочаев Р.Ж., Эннс В.И., Маршрут топологического синтеза для реконфигурируемых систем на кристалле специального назначения // Микроэлектроника, 2019, том 48, № 3, С. 211–223. DOI:10.1134/s0544126919030050
- [3] Фролова П.И., Чочаев Р., Иванова Г.А., Гаврилов С.В. Алгоритм размещения с оптимизацией быстродействия на основе матриц задержек для реконфигурируемых систем на кристалле // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2020. Вып. 1. С. 2-7. doi:10.31114/2078-7707-2020-1-2-7
- [4] Гаврилов С.В., Железников Д.А., Чочаев Р., Хватов В.М. Алгоритм декомпозиции на основе метода имитации отжига для реконфигурируемых систем на кристалле // Проблемы разработки перспективных микро- и нанoeлектронных систем(МЭС). 2018. Вып. 1. С. 199-204. doi:10.31114/2078-7707-2018-1-199-204
- [5] Гаврилов С.В., Железников Д.А., Чочаев Р.Ж. Разработка и сравнительный анализ методов решения задачи размещения для реконфигурируемых систем на кристалле // Изв. вузов. Электроника. 2020. Т. 25. № 1. С. 48–57. DOI: 10.24151/1561-5405-2020-25-1-48-57
- [6] Markov I.L., Hu J., Kim M. Progress and Challenges in VLSI Placement Research // Proceedings of the IEEE, Nov. 2015, vol. 103, no. 11, pp. 1985-2003. DOI: 10.1109/JPROC.2015.2478963
- [7] Betz V., Rose J. VPR: A new packing, placement and routing tool for FPGA research // Proceedings of the 7th International Workshop on Field-Programmable Logic and Applications, ser. FPL '97. 1997, pp. 213-222. DOI: 10.1007/3-540-63465-7_226
- [8] Лебедев Б. К., Степаненко С. А. Генетический алгоритм размещения, управляемый временными ограничениями // Известия ЮФУ. Технические науки. 2007. №1. URL: <https://cyberleninka.ru/article/n/geneticheskiy-algoritm-razmescheniya-upravlyaemyy-vremennymi-ogranicheniyami>
- [9] Hajek B. Cooling schedules for optimal annealing // Math. Oper. Res. 1988. Vol. 13. P. 311–329.
- [10] Marcel Gort, Jason H. Anderson. Analytical placement for heterogeneous FPGAs. // 22nd International Conference on Field Programmable Logic and Applications (FPL). - 2012. - С. 143 - 150.
- [11] M.-C. Kim, D. Lee, and I. Markov. SimPL: An effective placement algorithm. // IEEE TCAD, vol. 31, no. 1, - 2012. - С. 50-60.
- [12] M. Hutton, K. Adibsamii, A. Leaver. Adaptive delay estimation for partitioning driven PLD placement [Текст] // IEEE Transactions on VLSI 11(1). - 2003. - С. 60 - 63.
- [13] J. Rose, W. Snelgrove, Z. Vranesic. ALTOR: An automatic standard cell layout program. // Proceedings of the Canadian Conference on VLSI. - 1985. - С. 169 - 173.
- [14] Z. Lin, Y. Xie, G. Qian, S. Wang, J. Yu and J. Chen, "Late Breaking Results: An Analytical Timing-Driven Placer for Heterogeneous FPGAs*," 2020 57th ACM/IEEE Design Automation Conference (DAC), 2020, pp. 1-2, doi: 10.1109/DAC18072.2020.9218699.
- [15] S. Prasad, J. R. Kukula and W. J. Kubitz, "Path-Delay Constrained Floorplanning: A Mathematical Programming Approach for

- Initial Placement," 26th ACM/IEEE Design Automation Conference, 1989, pp. 364-369, doi: 10.1145/74382.74443.
- [16] Арутюнян А.Г. Начальное размещение логических ячеек интегральных схем с учетом важности цепей // Проблемы разработки перспективных микро- и наноэлектронных систем - 2014. Сборник трудов / под общ. ред. академика РАН А.Л. Стемповского. М.: ИППМ РАН, 2014. Часть 1. С. 143-146.
- [17] Kennings A., Vorwerk K., Force-Directed Methods for Generic Placement.// Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 2007. Vol. 25 № 10., pp. 2076 - 2087. 10.1109/TCAD.2005.862748.
- [18] V. S. Melikyan, A. G. Harutyunyan, N. S. Vagharshakyan and H. J. Harutyunyan, "Multifactor initial placement of IC cells," 2017 IEEE 37th International Conference on Electronics and Nanotechnology (ELNANO), 2017, pp. 104-107, doi: 10.1109/ELNANO.2017.7939726.
- [19] H. Eisenmann and F. M. Johannes, "Generic global placement and floorplanning," Proceedings 1998 Design and Automation Conference. 35th DAC. (Cat. No.98CH36175), 1998, pp. 269-274, doi: 10.1145/277044.277119.
- [20] Brglez F., Bryan D., Kozminski K. Combinational profiles of sequential benchmark circuits // Proc. Of the Intern. Symposium of Circuits and Systems. 1989. P. 1929–1934.
- [21] Bryan D. The ISCAS '85 Benchmark circuits and netlist format. North-Carolina State University, 1985. P. 4.
- [22] Yosys Open Synthesis Suite. Version 0.8. // <http://www.clifford.at/yosys/>

Development and Comparative Analysis of Initial Placement Methods for FPGA

P.I. Frolova, R.Zh. Chochaev

Institute for Design Problems in Microelectronics of RAS, Moscow

frolova_p@ippm.ru, chochaev_r@ippm.ru

Abstract — In this paper, we analyze different ways of generating initial placement for island-style FPGAs.

In this work we consider five algorithms: random, force-directed, input/output ranking, and line-by-line. After each initial placement we perform optimization using simulated annealing followed by routing. In the random algorithm, for each logic element random legal coordinates are iteratively generated. In the force-directed algorithm, the legal placement areas for flip-flops are determined firstly, and then each logic element is placed to the placement site closest to its center of mass. Line-by-line placement is done sequentially in each line from left to right starting from the top. In the ranking algorithm the position of the element on the paths, rank, from inputs to outputs and vice versa is considered. An element rank is calculated using breadth first search. The ranking algorithm is performed separately from input and output cells. The elements are placed starting with the lowest rank using a force-directed algorithm.

The obtained results showed that the initial placement can affect the runtime of simulated annealing, placement quality, and, as a result, the performance of circuits. On average, the input ranking algorithm generated placement with the smallest delays, followed by the random and the force-directed algorithms. Placements generated by the force-directed and ranked algorithms have better wirelength. However, the line-by-line and random algorithms did not generate a routable placement for some cases.

Keywords — FPGA, initial placement, electronic design automation (EDA)

REFERENCES

- [1] Hauck S., DeHon A. Reconfigurable Computing: The Theory and Practice of Fpga-Based Computation San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 2007.
- [2] Gavrilov S.V., Zheleznikov D.A., Zapletina M.A., Khvatov V.M., Chochaev R.Z., Enns V.I. Layout Synthesis Design Flow for Special-Purpose Reconfigurable Systems-on-a-Chip // Russian Microelectronics. 2019. 48(3), P. 176-186. DOI:10.1134/s1063739719030053
- [3] Frolova P.I., Chochaev R., Ivanova G.A., Gavrilov S.V. Timing-driven placement algorithm based on delay matrix model for reconfigurable system-on-chip // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2020. Issue 1. P. 2-7. doi:10.31114/2078-7707-2020-1-2-7
- [4] Gavrilov S.V., Zheleznikov D.A., Chochaev R., Khvatov V.M. Partitioning Algorithm Based on Simulated Annealing for Reconfigurable Systems-on-Chip // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2018. Issue 1. P. 199-204. doi:10.31114/2078-7707-2018-1-199-204
- [5] Gavrilov S.V., Zheleznikov D.A., Chochaev R.Z. Development and comparative analysis of placement methods for reconfigurable systems-on-a-chip. Proc. Univ. Electronics, 2020, vol. 25, no. 1, pp. 48–57. DOI: 10.24151/1561-5405-2020-25-1-48-57
- [6] Markov I.L., Hu J., Kim M. Progress and Challenges in VLSI Placement Research // Proceedings of the IEEE, Nov. 2015, vol. 103, no. 11, pp. 1985-2003. DOI: 10.1109/JPROC.2015.2478963
- [7] Betz V., Rose J. VPR: A new packing, placement and routing tool for FPGA research // Proceedings of the 7th International Workshop on Field-Programmable Logic and Applications, ser. fPL '97. 1997, pp. 213-222. DOI: 10.1007/3-540-63465-7_226
- [8] Lebedev B. K., Stepanenko S. A. Geneticheskij algoritm razmeshcheniya, upravlyaemyj vremennymi ogranicheniyami (Timing-driven genetic placement algorithm) // Izvestiya YUFU. Tekhnicheskie nauki. 2007. No1. URL: <https://cyberleninka.ru/article/n/geneticheskij->

- algorithm-razmescheniya-upravlyaemyy-vremennymi-ogranicheniyami
- [9] Hajek B. Cooling schedules for optimal annealing // *Math. Oper. Res.* 1988. Vol. 13. P. 311–329.
- [10] Marcel Gort, Jason H. Anderson. Analytical placement for heterogeneous FPGAs. // 22nd International Conference on Field Programmable Logic and Applications (FPL). - 2012. - C. 143 - 150.
- [11] M.-C. Kim, D. Lee, and I. Markov. SimPL: An effective placement algorithm. // *IEEE TCAD*, vol. 31, no. 1, - 2012. - C. 50-60.
- [12] M. Hutton, K. Adibsamii, A. Leaver. Adaptive delay estimation for partitioning driven PLD placement [Текст] // *IEEE Transactions on VLSI* 11(1). - 2003. - C. 60 - 63.
- [13] J. Rose, W. Snelgrove, Z. Vranesic. ALTOR: An automatic standard cell layout program. // *Proceedings of the Canadian Conference on VLSI*. - 1985. - C. 169 - 173.
- [14] Z. Lin, Y. Xie, G. Qian, S. Wang, J. Yu and J. Chen, "Late Breaking Results: An Analytical Timing-Driven Placer for Heterogeneous FPGAs*," 2020 57th ACM/IEEE Design Automation Conference (DAC), 2020, pp. 1-2, doi: 10.1109/DAC18072.2020.9218699.
- [15] S. Prasad and W. J. Kubitz, "Path-Delay Constrained Floorplanning: A Mathematical Programming Approach for Initial Placement," 26th ACM/IEEE Design Automation Conference, 1989, pp. 364-369, doi: 10.1145/74382.74443.
- [16] Harutyunyan A.G. Initial placement of digital logic cells in integrated circuits considering net priority // *Problems of Perspective Micro- and Nanoelectronic Systems Development* - 2014. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2014. Part 1. P. 143-146.
- [17] Kennings A., Vorwerk K., Force-Directed Methods for Generic Placement.// *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on 2007. Vol. 25 № 10., pp. 2076 - 2087. 10.1109/TCAD.2005.862748.
- [18] V. S. Melikyan, A. G. Harutyunyan, N. S. Vagharshakyan and H. J. Harutyunyan, "Multifactor initial placement of IC cells," 2017 IEEE 37th International Conference on Electronics and Nanotechnology (ELNANO), 2017, pp. 104-107, doi: 10.1109/ELNANO.2017.7939726.
- [19] H. Eisenmann and F. M. Johannes, "Generic global placement and floorplanning," *Proceedings 1998 Design and Automation Conference*. 35th DAC. (Cat. No.98CH36175), 1998, pp. 269-274, doi: 10.1145/277044.277119.
- [20] Brglez F., Bryan D., Kozminski K. Combinational profiles of sequential benchmark circuits // *Proc. Of the Intern. Symposium of Circuits and Systems*. 1989. P. 1929–1934.
- [21] Bryan D. The ISCAS '85 Benchmark circuits and netlist format. North-Carolina State University, 1985. P. 4.
- [22] Yosys Open Synthesis Suite. Version 0.8. // <http://www.clifford.at/yosys>