

Планирование кристалла СБИС с учетом энергопотребления

И.Н. Ерошенко

Таганрогский технологический институт Южного федерального университета,
lutaries@rambler.ru

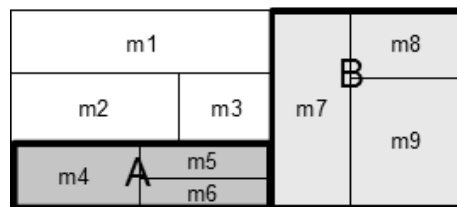
Аннотация — В статье рассматривается проблема планирования сверхбольших интегральных схем. Во введении обоснована актуальность задачи. В работе используется двухуровневая схема планирования систем на кристалле. На верхнем уровне используется генетический алгоритм разбиения на острова напряжения, позволяющий учитывать энергопотребление, на нижнем уровне используется меметический алгоритм, эффективно решающий задачу планирования в классической постановке. Приведено описание представления плана в виде обобщенной польской записи (ОПЗ). Дано описание структур хромосом, принципов их кодирования/декодирования для ОПЗ. Разработаны генетические операторы и гибридная схема поиска в виде меметического алгоритма. В качестве метода локального поиска в меметическом алгоритме использована коллективная альтернативная адаптация. Приведен псевдокод генетического алгоритма в качестве надстройки над меметическим алгоритмом для учета энергопотребления.

Ключевые слова — планирование СБИС, энергопотребление, генетический алгоритм, альтернативная адаптация, меметический алгоритм, обобщенная польская запись.

I. ВВЕДЕНИЕ

Эффективность энергопотребления стала важной проблемой при проектировании систем на кристалле в связи с повсеместным использованием портативных устройств [1]. Выделяют два вида потребляемой мощности: динамическое потребление и утечки. Динамическое потребление вызвано зарядом и разрядом емкости нагрузки во время переключения. Существует множество приемов, позволяющих снизить энергопотребление. Один из наиболее эффективных методов связан с понижением входного напряжения. Мультивольтажное проектирование используется для обеспечения минимально необходимой энергии для поддержки различных функциональных операций. Небольшие изменения напряжения приводят к значительным изменениям динамического энергопотребления, так как оно пропорционально квадрату напряжения.

Мультивольтажное проектирование включает разделение чипа на области, именуемые островами напряжения, которые могут работать при различных уровнях напряжения (рис. 1) [2].



Уровень напряжения на чипе 1.4 В
Остров напряжения А (1.3 В): m4, m5, m6
Остров напряжения В (1.2 В): m7, m8, m9

m1{1.2, 1.4}; m2{1.1, 1.2, 1.4}; m3{1.0, 1.3, 1.4}
m4{1.3, 1.4}; m5{1.1, 1.3}; m6{1.2, 1.3, 1.4}
m7{1.1, 1.2}; m8{1.1, 1.2, 1.3}; m9{1.0, 1.2, 1.4}

Рис. 1. Пример двух островов напряжения

При использовании островов напряжения процесс проектирования кристалла становится более сложным, так как приходится решать задачи разделения на острова, назначения напряжений и размещения модулей с учетом различных физических ограничений и необходимостью использования преобразователей уровня [1].

В данной работе рассмотрена двухэтапная схема проектирования, позволяющая осуществлять планирование СБИС с учетом энергопотребления [2]. На нижнем уровне используется гибридный генетический алгоритм, называемый меметическим, который решает задачу планирования в классической постановке. Он описан в разделе II. На верхнем уровне осуществляется разделение на острова напряжения с помощью генетического алгоритма. Данный подход рассматривается в разделе III.

II. МЕМЕТИЧЕСКИЙ АЛГОРИТМ ПЛАНИРОВАНИЯ СБИС

A. Классическая постановка задачи планирования СБИС

Имеется множество модулей $M = \{m_i \mid i = 1, 2, \dots, n\}$. Каждый модуль характеризуется кортежем $\langle w_i, h_i \rangle$, w_i – ширина i -го модуля, а h_i – высота i -го модуля. Модуль может свободно вращаться (по умолчанию). Кортеж $\langle x_i, y_i \rangle$ определяет координаты левого нижнего угла i -го модуля. Задается множество связей между модулями. Требуется определить такое размещение модулей, при котором площадь охватывающего прямоугольника является минимальной.

В. Представление плана «обобщенная польская запись»

В литературе представлены два класса планов: гильотинные и негильотинные.

План гильотинного типа получается путем рекурсивного деления прямоугольника с помощью вертикальных (V) и/или горизонтальных (H) гильотинных разрезов. Для гильотинного представления используется польская запись. Связь бинарного дерева разрезов и польской записи показана на рис. 2.

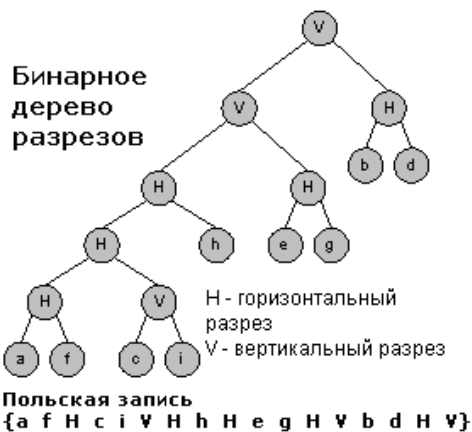


Рис. 2. Польская запись и бинарное дерево разрезов

Если план нельзя получить путем рекурсивных вертикальных и горизонтальных разрезов на части, то речь идет о негильотинном плане. Разработано множество негильотинных представлений. В данной работе используется обобщенная польская запись (ОПЗ), в которой используется дополнительный оператор @. Оператор @ называют угловым, он помещает модуль или супермодуль в угол, сформированный другими модулями [3]. Как показано на рис. 3, при наличии операторов только горизонтального и вертикального разрезов нельзя использовать мертвую зону. Угловой оператор @ задействует угол, образованный модулями А и В. Временная сложность трансформации ОПЗ в план – O(n).

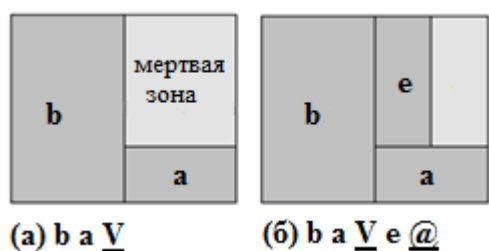


Рис. 3. Угловой оператор и ОПЗ

Точная позиция для углового оператора определяется угловым ограничением в виде кортежа (R, T), R – правая граница слева от вставляемого модуля, T – верхняя граница ниже вставляемого модуля. Угловой оператор может иметь несколько

угловых ограничений, поэтому выбирают то из них, которое лучше соответствует текущему модулю/супермодулю (не приводит к увеличению мертвой зоны).

С. Генетический алгоритм планирования СБИС

В генетическом алгоритме (ГА) эволюционный процесс представляется как способность лучших особей оказывать большее влияние на состав новой популяции на основе длительного выживания из более многочисленного потомства [4]. Основным блоком в ГА является блок генетических операторов [5].

Решение представляется набором хромосом S = {N1, N2, N3, N4}. Пусть n – число модулей. Тогда размеры хромосом N1, N2, N3 равны n-1, а размер хромосомы N4 равен n. Хромосома N1 кодирует разметку модулей (1 ≤ gi ≤ n+1-i), N2 определяет структуру бинарного дерева (1 ≤ gi ≤ n-1). С помощью хромосомы N3 задаются типы операторов (H, V, @). Значением гена является 0, 1 или 2 (флаг 0 – H-разрез, флаг 1 – V-разрез, флаг 2 – угловой оператор @). Хромосома N4 определяет ориентацию модулей. Если gi = 0, то высота прямоугольного модуля меньше либо равна ширине, при gi = 1 высота больше его ширины. Декодирование хромосом имеет оценку трудоемкости O(n), где n – число модулей. Пространственная сложность для одного решения имеет оценку O(n) [6].

В данном генетическом алгоритме используются три оператора кроссинговера: одноточечный K1, равномерный K2 и специализированный K3.

Одноточечный кроссинговер K1 является классическим, при его использовании две скрещиваемые хромосомы разрезаются один раз в соответствующей точке, затем производится обмен полученными частями. Оценка временной сложности O(n).

Равномерный кроссинговер K2 принципиально отличается от одноточечного кроссинговера. Каждый ген в потомстве создается посредством копирования соответствующего гена от одного или другого родителя с определенной вероятностью. Благодаря этому, пространство решений может быть исследовано более полно. Оценка временной сложности O(n).

Специализированный кроссинговер K3 учитывает специфику решаемой задачи путем анализа поддеревьев бинарных деревьев, но обладает большей трудоемкостью по сравнению с K1 и K2.

Рассмотрим работу кроссинговера K3. Анализируются деревья Ti и Tj, декодированные из особей i и j. Выбирается лучшее (в плане критерия оптимизации) поддерево STbest среди двух декодированных деревьев. Обозначим родителя, имеющего лучшее поддерево, как P1, а другого родителя – P2. Поддерево STbest помещается в специальный пул. Затем в пул добавляются поддеревья из P2, которые не содержат модулей из выбранного поддерева STbest. Множество модулей, не попавших в пул, помечаются как вырожденные поддеревья и

добавляются в пул. Из пула циклически выбираются два поддерева (парное сравнение), к ним применяются последовательно операторы \cap , \cup , \oplus . Определяется лучший результат при слиянии двух поддеревьев на текущем этапе. Лучшая пара поддеревьев удаляется из пула, а вместо них добавляется только что созданное поддерево. Процесс повторяется до тех пор, пока не будет получено дерево, содержащее все модули. Данный тип кроссинговера, осуществляющий иерархическую кластеризацию, имеет в худших случаях оценку временной сложности $O(n^2 \log n)$.

Мутация заключается в принятии геном случайного значения из заданного диапазона значений для гена в данном локусе. Оператор мутации является вторичным в ГА (по сравнению с кроссинговером), выполняется с небольшой вероятностью. Оценка временной сложности $O(n)$.

В основу селекции, как для выбора родительской пары хромосом, так и при отборе популяции, положен «принцип рулетки». В соответствии с этим принципом для всех решений рассчитывается вероятность выбора на основе приспособленности особей.

В работе используется адаптация частоты применения операторов кроссинговера. Используется пересчет вероятностей выбора операторов K_1 , K_2 , K_3 . Для этого вводится механизм подсчета количества получаемых прогрессивных особей за единицу времени. Прогрессивными считаются те особи, чей уровень приспособленности не хуже, чем у текущей лучшей особи. Динамический выбор операторов кроссинговера позволяет лучше подстраиваться под конкретное состояние популяции решений [10].

Псевдокод алгоритма планирования СБИС имеет следующий вид:

```

Алгоритм ПЛАНИРОВАНИЕ;
{
  задача := ИСХОДНЫЕ_ДАННЫЕ;
  генетика := НАСТРОЙКА;
  попул := НАЧ_ГЕНЕРАЦИЯ(задача, генетика);
  С := ЧИСЛО_ШАГОВ;
  К := 1;
  F = РАСЧЕТ(попул, генетика);
  ПОКА К <= С
  {
    ЭВОЛЮЦИЯ(попул);
    К := К + 1;
  }
}

```

В качестве исходных данных вводятся структуры *задача* и *генетика*. В структуре *задача* вводятся параметры модулей, критерии планирования и ограничения. В структуре *генетика* содержатся основные параметры, управляющие процессом генетического поиска.

Ниже представлен псевдокод процедуры *ЭВОЛЮЦИЯ* (кроссинговер указан обобщенно).

Процедура *ЭВОЛЮЦИЯ* (попул)

```

{
  Т := ЧИСЛО_ПОКОЛЕНИЙ;
  ПОКА Т > 0
  {
    кросс_попул :=  $\emptyset$ ;
    мут_попул :=  $\emptyset$ ;
    L := число_кросс;
    ПОКА L > 0
    {
      род_пара := ВЫБОР(попул);
      особи := КРОССИНГОВЕР(род_пара,
        генетика);
      кросс_попул := кросс_попул  $\cup$  особи;
      L := L - 1;
    }
    тек_попул := попул  $\cup$  кросс_попул;
    мут_попул := МУТАЦИЯ(тек_попул, генетика);
    тек_попул := тек_попул  $\cup$  мут_попул;
    F := РАСЧЕТ(тек_попул, генетика);
    лучш_реш := ВЫДЕЛИТЬ(тек_попул, F);
    попул := СЕЛЕКЦИЯ(тек_попул, F);
  }
}

```

Данный генетический алгоритм был включен в островную модель ГА. Островные ГА позволяют улучшить надежность получаемых решений. Период миграции задается динамически, миграция происходит тогда, когда хотя бы в одной из подпуляций длительно не улучшалось решение.

D. Коллективная альтернативная адаптация

М.А. Цетлин предложил использовать вероятностный автомат адаптации (АА). АА может принимать сигналы поощрения ($-$) и наказания ($+$). Под действием таких сигналов состояние АА меняется. В рамках адаптивного процесса автомат переходит в состояние, которое соответствует лучшей альтернативе [7].

Представим формулировку задачи в виде адаптивной системы.

Ориентация модуля m_i при его размещении в области задается параметром o_i . $o_i \in \{0, 1\}$, т.е. для m_i возможны два способа (две ориентации) размещения. Если $o_i = 0$, то высота прямоугольного модуля меньше, либо равна ширине, при $o_i = 1$ высота больше его ширины.

Пространство решений составляют решения, отличающиеся друг от друга значениями элементов множества O , задающими ориентацию модулей.

Введем для оценки состояния объекта адаптации в среде два параметра $\langle P_x^1, P_y^1 \rangle$. $P_x^1 = 1$, если размер w_i модуля входит в состав выражения, определяющего ширину w_0 охватывающего прямоугольника R ; $P_x^1 = 0$ – в противном случае. $P_y^1 = 1$, если размер h_i модуля m_i , входит в состав выражения, определяющего высоту h_0

прямоугольника R; $P_y^1 = 0$ – в противном случае (рис. 4).

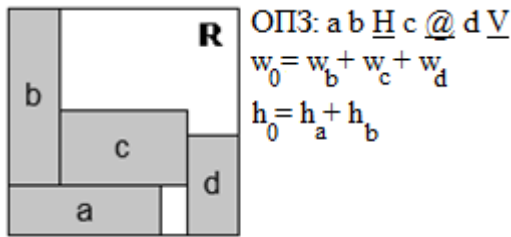


Рис. 4. Модули, определяющие высоту и ширину плана

Возможны четыре комбинации значений параметров $\langle P_x^1, P_y^1 \rangle$: $\langle 0, 0 \rangle$, $\langle 0, 1 \rangle$, $\langle 1, 0 \rangle$, $\langle 1, 1 \rangle$. Если $P_x^1 = 1$, то предпочтительней будет альтернатива A2, т.к. в выражение для w_0 войдет меньшая величина. Если $P_y^1 = 1$, то предпочтительней будет альтернатива A1, следовательно, в выражение для h_0 войдет меньшая величина [7].

Для реализации механизма адаптации каждому модулю ставится в соответствие автомат адаптации AA с двумя группами состояний $\{C_i^1, C_i^2\}$, соответствующих двум альтернативам A_1^1 и A_2^1 . Число состояний в группе задается параметром Q_i , называемым глубиной памяти. На вход автомата адаптации AA_i подается сигнал "поощрение" или "наказание" в зависимости от состояния объекта адаптации в среде. На рис. 5 показана схема переходов автомата адаптации. Знаком (+) помечены переходы под действием сигнала "поощрение", знаком (-) помечены переходы под действием сигнала "наказание".

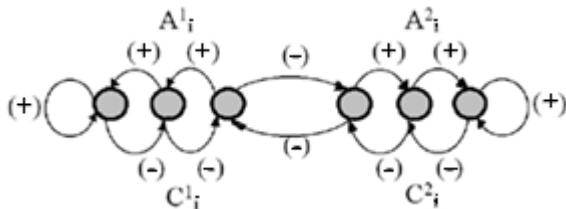


Рис. 5. AA с двумя группами состояний

Если предпочтительная альтернатива в соответствии со значениями пары параметров $\langle P_x^1, P_y^1 \rangle$ совпадает с реализованной в данный момент альтернативой, то вырабатывается сигнал поощрения, в противном случае – сигнал наказания. Отметим, что в случае, когда пара параметров $\langle P_x^1, P_y^1 \rangle$ имеет значение $\langle 0, 0 \rangle$ или $\langle 1, 1 \rangle$, предпочтительной альтернативы нет. Поэтому в случае $\langle 0, 0 \rangle$ управляющий сигнал не вырабатывается вообще (табл. 1).

Таблица 1

Правила генерации управляющих сигналов

P_x^1	P_y^1	Предпочтительная альтернатива	Реализованная альтернатива	Управляющий сигнал
0	1	A_1^1	A_1^1	(+)

0	1	A_1^1	A_2^1	(-)
1	0	A_2^1	A_1^1	(-)
1	0	A_1^1	A_2^1	(+)
0	0	Нет	A_1^1 или A_2^1	Нет
1	1	Нет	A_1^1 или A_2^1	(-) с вероятностью P

Работа адаптивной системы на каждой итерации осуществляется за четыре такта [7]:

1. В соответствии с выражениями для определения размеров сторон w_0 и h_0 прямоугольника R для каждого модуля m_i определяются значения пары параметров $\langle P_x^1, P_y^1 \rangle$.

2. Для каждого автомата адаптации в соответствии со значениями пары параметров $\langle P_x^1, P_y^1 \rangle$ и состоянием автомата адаптации вырабатываются управляющие сигналы ("поощрение" или "наказание").

3. Под действием управляющего сигнала осуществляется переход в автоматах адаптации.

4. В соответствии с состояниями автоматов адаптации реализуются альтернативы для модулей, т.е. выбираются их соответствующие ориентации. Затем с помощью процедуры свертки строится план, определяются выражения для параметров w_0 и h_0 .

Е. Композиция генетического алгоритма и коллективной альтернативной адаптации

Представленные алгоритмы могут работать автономно. Каждый обладает достоинствами и недостатками. ГА позволяет находить достаточно качественные решения, но для получения решения, близкого к оптимальному, требуется набор «осмысленных» действий, учитывающих специфику задачи. Альтернативная адаптация работает на локальном уровне: на вход подается ОПЗ, а затем подбирается ориентация модулей. Гибридизация метаэвристик в виде меметического алгоритма (МА) позволяет объединить достоинства генетического алгоритма и алгоритма альтернативной адаптации.

В данной работе связующим звеном для композиции метаэвристик выступает хромосома H4. При расчете приспособленности текущей особи можно подбирать конечные значения генов хромосомы H4 с помощью обучающихся автоматов. Для этого требуется установить взаимно однозначное соответствие между альтернативой (ориентацией модулей) и значением гена (рис. 6).



Рис. 6. Связь автоматов адаптации (AA) и хромосомы H4

В предлагаемом МА метод локального поиска, представленный коллективной альтернативной адаптацией, применяется только к тем особям, у которых уровень приспособленности выше заданного порогового значения.

III. АЛГОРИТМ ПЛАНИРОВАНИЯ СБИС С УЧЕТОМ ЭНЕРГОПОТРЕБЛЕНИЯ

A. Постановка задачи планирования СБИС с учетом энергопотребления

Имеется множество модулей $M = \{m_i \mid i=1,2,\dots,n\}$. Помимо геометрических параметров из классической постановки, с каждым модулем связана таблица энергопотребления T_i . Данная таблица определяет допустимые уровни напряжения и соответствующие усредненные значения потребляемой мощности.

При заданной константе K и напряжении питания на уровне кристалла V_c необходимо построить план F с числом островов напряжения $\leq K$ таким образом, чтобы снизить энергопотребление (учитывается и энергопотребление преобразователей уровня), площадь (включая площадь преобразователей уровня) и длину связей. Каждому острову назначается наименьший допустимый уровень напряжения, общий для всех модулей острова. Модули, не назначенные ни одному острову, будут работать с напряжением V_c .

B. Генетический алгоритм разбиения на острова напряжения

В рамках предлагаемого генетического алгоритма хромосома в виде кортежа целых чисел кодирует решение по разбиению множества модулей на острова напряжения. Все операторы в данном ГА имеют линейную оценку временной сложности.

Отбор особей для скрещивания осуществляется методом рулетки. Пример использования одноточечного кроссинговера приведен на рис. 7. Оператор мутации является тривиальным (ген принимает случайное значение из заданного диапазона значений для гена в данном локусе).

Данный алгоритм в отличие от канонического поколенческого ГА является стационарным. В остальном он не отличается от типовых ГА, что значительно облегчает его программную реализацию с использованием метаэвристических библиотек.



Рис. 7. Одноточечный кроссингвер

Как показано на рис. 8, ГА создает популяцию хромосом случайным образом, затем оценивается каждый остров. При планировании учитываются

характеристики преобразователей уровня (площадь, энергопотребление).

Алгоритм

```

{
  Дана информация
  о геометрических параметрах модулей
  и их энергопотреблении.
  Применение ГА для создания
  начальной популяции;
  Пока не выполнены условия останова
  {
    Для каждой хромосомы в популяции:
    {
      Для каждого острова напряжения,
      закодированного в хромосоме:
      {
        Применение МА
        для планирования
        в рамках текущего острова;
      }
      Применение МА
      для планирования на уровне кристалла;
      Оценивание приспособленности популяции;
      Применение операторов селекции,
      кроссинговера и мутации
      для создания новых хромосом;
    }
  }
}
  
```

Рис. 8. Генетический алгоритм планирования СБИС с учетом энергопотребления

IV. ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

Предлагаемый в данной статье алгоритм планирования СБИС на основе композиционного подхода реализован в виде оконного приложения для ОС Windows на языке C++ в среде разработки Microsoft Visual Studio 2010 с использованием стандартной библиотеки шаблонов STL (для проектирования ядра), программной библиотеки генетических алгоритмов GAlib (для реализации генетического алгоритма разбиения на острова напряжения) и графической библиотеки Qt 4.7.3.

Тестирование проводилось на ЭВМ с процессором Intel Core 2 Duo T6600 2200 МГц, 4 Гб ОЗУ (ОС Windows 7). Эксперименты подтвердили теоретическую оценку временной сложности для меметического алгоритма планирования СБИС (рис. 9). Она оказалась близка к линейной как при использовании обычной польской записи (ПЗ), так и при использовании обобщенной польской записи (ОПЗ).

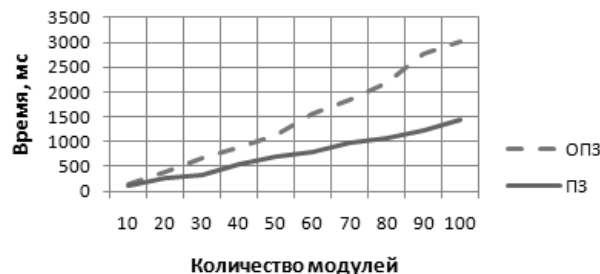


Рис. 9. График временной зависимости планирования от количества модулей

В табл. 2 приведены результаты экспериментов для MCNC бенчмарков. Для каждого модуля было выбрано подмножество допустимых уровней напряжений из множества значений {1.1 В, 1.3 В, 1.5 В, 1.8 В}, площадь каждого преобразователя уровня равна 10x10, потребляемая мощность равна 1 [1].

Для сравнения был выбран современный планировщик СБИС из работы [1], который использует классическое моделирование отжига и обычную польскую запись. Предлагаемый в статье комплексный алгоритм ГА+МА в связке с ОПЗ лучше справляется с упаковкой модулей при соизмеримом энергосбережении.

Таблица 2

Сравнения с аналогом

Тест	Энергосбережение (%)		Мертвая зона (%)		Время работы (с)	
	ГА+МА	[1]	ГА+МА	[1]	ГА+МА	[1]
apte	50.87	51.64	2.24	2.341	1.78	1.457
xerox	42.23	40.45	3.62	3.667	1.97	1.335
hp	39.62	41.34	3.59	3.855	2.13	1.570
ami33	51.15	48.69	4.31	4.376	24.4	19.43
ami49	48.5	47.21	4.18	4.840	73.8	68.78

В дальнейшем предполагается создание многопоточной версии данного планировщика, а также проведение дополнительной серии тестов для более полного изучения его характеристик.

V. ЗАКЛЮЧЕНИЕ

Мультивольтажное проектирование систем на кристалле становится все более распространенным. Назначение напряжения каждому модулю и последующее планирование с островами напряжения не является тривиальной задачей.

Благодаря совместному использованию подходов, предложенных в работах [2], [3], [6]-[10], создан прототип комплексного планировщика СБИС, который использует ГА на двух уровнях и альтернативную адаптацию для поиска субоптимального плана, представленного в виде довольно простой и эффективной структуры ОПЗ. ОПЗ позволяет получать более компактные планы по сравнению с классическими гильотинными [3].

Разработанные в [6] многохромосомные структуры, отличающиеся простотой декодирования и гомологичностью, были модифицированы для поддержки ОПЗ (негильотинных планов).

Для улучшения качества получаемых решений в МА помимо альтернативной адаптации [7] используются разные виды кроссинговера (включая специализированный для ОПЗ), адаптация вероятностей выбора оператора кроссинговера, островная модель ГА с динамическим периодом миграции особей [10], что выгодно отличает его от

обычных генетических алгоритмов и моделирования отжига.

К достоинствам разработанного подхода следует отнести возможность применения планировщиков, разработанных для классического академического планирования, с минимальными изменениями.

ЛИТЕРАТУРА

- [1] Ma Q., Young E.F.Y. Multivoltage Floorplan Design // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2010. V. 29. № 4. pp. 607 – 617.
- [2] Hung, W.L. et al. Temperature-Aware Voltage Islands Architecting in System-on-Chip Design // Proceedings of the 2005 International Conference on Computer Design. 2005. pp. 689–696.
- [3] Lin C.-T. et al. GPE: A New Representation for VLSI Floorplan Problem // IEEE International Conference on Computer Design (ICCD'02). 2002. pp. 42–44.
- [4] Курейчик В.М. Гибридные генетические алгоритмы // Известия Южного федерального университета. Технические науки. 2007. Т. 77. № 2. С. 5–12.
- [5] Курейчик В.М. Модифицированные генетические операторы // Известия Южного федерального университета. Технические науки. 2009. Т. 101. № 12. С. 7–14.
- [6] Лебедев В.Б. Планирование СБИС на основе многоуровневой эволюции // Известия Таганрогского государственного радиотехнического университета. 2001. Т. 22. № 4. С. 153–163.
- [7] Лебедев Б.К., Лебедев В.Б. Адаптивная процедура выбора ориентации модулей при планировании СБИС // II Всероссийская научно-техническая конференция "Проблемы разработки перспективных микроэлектронных систем (МЭС)". М.: ИППИ РАН, 2006. С. 120–125.
- [8] Maolin T., Xin Y. A memetic algorithm for VLSI floorplanning // IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics. 2007. № 37(1). pp. 62–69.
- [9] Chen D.-S. et al. A robust genetic algorithm for rectangle packing problem // Journal of Combinatorial Optimization. 2006. V. 12.
- [10] Мясников А.С. Островной генетический алгоритм с динамическим распределением вероятностей выбора генетических операторов // Наука и образование: электронное научно-техническое издание. 2010. № 1. С. 3-3.