

# Пакет программ для перевода проекта схемы ПЛИС в базис БМК

Сергей Артёмов (Московская обл.)

**В данной статье рассматривается программное обеспечение ConvChip, позволяющее конвертировать проект ПЛИС в базис БМК. Описываются методы перевода и сохранения временных соотношений между элементами схемы в базисах ПЛИС и БМК.**

## ВВЕДЕНИЕ

В последнее десятилетие проблема создания опытных образцов микросхем стоит особенно остро. Такие образцы необходимы для экспериментального исследования разработанного на ЭВМ проекта микросхемы на соответствующем стенде или непосредственно в аппаратуре. На практике первые опытные образцы микросхем часто не соответствуют предъявляемым к ним требованиям. Поэтому соответствующая модификация микросхемы выполняется на ЭВМ на базовых матричных кристаллах (БМК) с повторным выпуском опытных образцов. Этот процесс может повторяться многократно и длиться многие месяцы, что приведёт к большим финансовым затратам.

Одним из способов решения этой проблемы является отладка разработанного на ЭВМ проекта в аппаратуре типа программируемых логических интегральных схем (ПЛИС). В отличие от многомесячного изготовления опытного образца микросхемы, получение опытного образца ПЛИС с требуемыми функциями занимает несколько минут. Для этого к ЭВМ подсоединяют программатор, который конфигурирует ПЛИС под конкретный проект. Однако использование ПЛИС приводит к другой проблеме. Серийный выпуск ПЛИС стоит значительно дороже, чем выпуск аналогичной партии полужаказных микросхем. Особенно это чувствительно, когда необходимо выпустить партию микросхем специального назначения, т.е. с военной приёмкой. Для решения такой проблемы проект в базисе ПЛИС конвертируется в базис БМК полужаказных микросхем.

Современные системы автоматизированного проектирования (САПР)

таких фирм, как Cadence ([www.cadence.com](http://www.cadence.com)), Mentor Graphics ([www.mentorgraphics.com](http://www.mentorgraphics.com)), Synopsys ([www.synopsys.com](http://www.synopsys.com)), Logic Vision ([www.logicvision.com](http://www.logicvision.com)) и др., частично решают проблему конвертирования. Для этого проект в базисе ПЛИС импортируется в САПР, после чего динамические параметры всех примитивов в базисе ПЛИС заменяют в соответствии с базисом БМК. Фактически происходит замена базиса ПЛИС на базис БМК.

Если схема разработана на стандартных языках высокого уровня типа VHDL, Verilog 2001, SystemC, SystemVerilog, PSL, то такой подход полностью себя оправдывает [1]. Если схема имеет комбинированное описание, то автоматическое конвертирование таких схем из базиса ПЛИС в базис БМК несколько затруднено.

Существует другой подход, когда исходный проект разрабатывается в базисе БМК, а для отладки проект конвертируется в базис ПЛИС. После отладки ПЛИС в аппаратуре выпускается партия микросхем на основе БМК. Однако такой подход не всегда представляется возможным, поскольку существует большое количество проектов, изначально разработанных в базисе ПЛИС [2, 3].

Современные САПР не адаптированы под асинхронные схемы и ориентированы на разработку исключительно синхронных схем. Если схема частично содержит модули, не удовлетворяющие принципу синхронизации, то конвертирование таких проектов ПЛИС в базис БМК сильно затруднено. Такие схемы после конвертирования оказываются не работоспособными из-за разных динамических параметров базисов ПЛИС и БМК. Такие проекты отлаживаются в САПР с большим трудом.

Тем не менее, необходимость конвертирования проектов смешанных схем в настоящее время является актуальной задачей. Рассматриваемый в данной статье пакет программ ConvChip позволяет конвертировать проекты схем из базиса ПЛИС в базис БМК [4].

Для того чтобы конвертировать схему, необходимо каждому элементу в схеме ПЛИС поставить в соответствие некоторую подсхему замещения из базиса БМК. Оптимальный выбор такой подсхемы и является основной задачей конвертирования, решаемой конвертором ConvChip. При подстановке подсхем замещения из базиса БМК конечная схема часто оказывается неработоспособной. Это проявляется в том, что ответные реакции, полученные при моделировании схемы в базисе ПЛИС, не совпадают с ответными реакциями, полученными при моделировании в базисе БМК (рассматриваются только те моменты времени, когда схема находится в стабильно функционирующем состоянии). Несовпадение возникает в случае логически эквивалентных схем за счёт различий в динамических параметрах элементов, входящих в базисы ПЛИС и БМК, а также отличий архитектуры ПЛИС и полужаказной микросхемы.

Для того чтобы восстановить работоспособность схемы в базисе БМК, в конверторе ConvChip выполняются алгоритмы на основе следующих методов:

- метод модификации, позволяющий выполнить автоматическую коррекцию схемы без участия специалиста. Этот метод разделяется на метод перевода схемы и метод сохранения временных соотношений;
- метод диагностики, позволяющий исследовать схему, представленную одновременно в двух базисах, на предмет обнаружения причин её неработоспособности после конвертирования (рассмотрение этого метода не входит в рамки данной статьи [5]).

### Метод перевода схемы

Метод перевода схемы в САПР сводится к замене базиса ПЛИС на базис БМК. Для этого исходный проект в базисе ПЛИС импортируется в САПР. После чего по имени элемента схемы в САПР осуществляется его замена на элемент в базисе БМК. Такая замена в современных САПР имеет ряд недостатков.

Если для модуля из ПЛИС существует несколько вариантов замены, различающихся по количеству элементов в них, быстродействию и флуктуационной устойчивости, то выбор в базисе БМК осуществляется без анализа выбора оптимального варианта. Например, если для счётчика с именем В существуют три варианта замены – А, В и С – в базисе БМК, то выбор осуществляется по совпадению имени, т.е. вариант В. При этом анализ вариантов А и С в САПР не выполняется. Но желательно, чтобы выбор варианта происходил не по имени В, а по критерию оптимального соотношения количества элементов, быстродействия и флуктуационной устойчивости.

Если модуль из ПЛИС не представлен одним иерархическим блоком, то он будет заменён на примитивы из базиса ПЛИС поэлементно. Между тем необходим анализ циклов на графе схемы на предмет обнаружения фрагментов, для которых существует единый модуль из базиса БМК. Например, если для некоторой группы вентилях в ПЛИС, соединённых обратными связями, существует неделимый библиотечный элемент в базисе БМК, выполняющий ту же функцию, то необходимо выполнить замену группы вентилях целиком на этот библиотечный элемент. В САПР не реализованы алгоритмы анализа таких групп вентилях. В итоге каждый вентиль переводится в базис БМК отдельно от группы. Это хуже, чем замена на один библиотечный элемент всей группы вентилях целиком.

Реализованные в конверторе ConvChip алгоритмы перевода устраняют данные недостатки.

Обозначим схему ПЛИС за ориентированный граф  $G = G(A, R)$ , вершины А которого обозначают элементы, их выводы и связи, а  $R(a, b)$  – ребро между вершинами  $a, b \in A$ . Ребро  $R(a, b)$  характеризует наличие связи в схеме между элементами  $a$  и  $b$ . Любому примитиву соответствует граф  $G'$ . Введём для  $G'$  следующие обозначения:

- $|G'|$  – количество вершин в  $G'$ ;
- $\|G'\|$  – быстродействие  $G'$ ;
- $\|G'\|$  – флуктуационная устойчивость  $G'$ ;
- $F(G')$  – выполняемая логическая функция  $G'$ ;
- $D(G')$  – временная задержка от входа до выхода  $G'$ ;
- $\langle D(G') \rangle$  – функция отбрасывания дробной части задержки  $G'$ .

Библиотеки базиса БМК достаточно сильно отличаются между собой. В одних набор базисных элементов ограничивается набором различного типа вентилях и простейших триггеров. В других существует широкий набор модулей, выполняющих функции счётчиков, сдвиговых регистров, сумматоров и т.д. Таким образом, существует много вариантов замены одного элемента (примитива или модуля) из схемы в базисе ПЛИС на подсхемы замещения в базисе БМК. Выбор наиболее оптимального варианта и составляет суть задачи, решаемой методом перевода схемы.

При выполнении задачи перевода осуществляется преобразование (или подстановка)  $e \rightarrow \tilde{e} = \Omega(G)$ . Здесь элемент  $\tilde{e}$  в базисе БМК есть образ  $e$  в базисе ПЛИС при преобразовании  $\Omega$ . Фактически схема  $G$  в базисе ПЛИС отображается в схему  $B$  в базисе БМК. Таким образом, описание метода перевода схемы сводится к описанию преобразования  $\Omega$ . Этот процесс состоит из следующих этапов:

1. Выбор модуля  $G''$  из схемы в базисе ПЛИС. Если в  $G$  существует модуль  $G''$ , т.е.  $\exists G'' : G'' \subseteq G$ , то необходимо проверить возможность замены  $G''$  целиком на модуль  $B''$ . В САПР такая замена возможна только для модулей  $G''$ , оформленных в виде отдельных иерархических блоков;
2. Поиск двух модулей  $b$  и  $B''$  из базиса БМК для одного модуля  $G'$  из базиса ПЛИС. Такие модули выполняют одинаковую функцию  $F(b) = F(B'') = F(G')$ , что и модуль  $G''$ ;
3. Поиск одного варианта подсхемы замещения  $B''$  для модуля  $G''$ . Если в базисе БМК существует только одна схема замещения для модуля  $G''$ , т.е.  $\exists B'' : F(B'') = F(G'')$ , то она сравнивается с вариантом замены каждого примитива  $G'_i$  из модуля  $G''$  в базисе ПЛИС на примитив  $B'_i$  в базисе БМК. При этом  $B'_i = \Omega(G'_i) : G'' = UG'_i, B' = UB'_i$  где  $i = 1..Q$ . Здесь  $G''$  является модулем в базисе БМК, а  $B'$  не является модулем и представляет собой

подсхему, состоящую из примитивов в базисе БМК и  $F(B') = F(G'')$ ;

4. Поиск одного варианта подсхемы замещения  $B''$  для модуля  $G''$ . Вариант замены  $G''$  целиком на один модуль  $B''$  является более предпочтительным, чем вариант замены  $B'$  на множество примитивов  $B'_i$ . Иными словами,  $G'_i \in G'', B' = \Omega(G'')$ , как правило, менее оптимален, чем  $B'' = \Omega(G'')$ . Поэтому если хотя бы одно из следующих трёх условий:  $|B''| < |B'|, \|B''\| > \|B'\|, \|B''\| > \|B'\|$  на данном этапе выполняется, то осуществляется  $B'' = \Omega(G'')$ . В современных САПР выполнение таких условий не проводится;
5. При наличии для модуля  $G''$  двух вариантов модулей  $b$  и  $B''$  они сравниваются между собой (аналогично п. 4). Однако, в отличие от  $B'$  в п. 4, здесь модули  $b$  и  $B''$  представлены каждый одним модулем. Такое сравнение в известных методах перевода не выполняется;
6. После сравнения по п. 5 для модуля  $G''$  выбирается наиболее оптимальный вариант  $B''$  или  $b$ :  $B'' = \Omega(G'')$ ,  $b = \Omega(G'')$ . При этом  $G = gU(UG'_i)UG''$ . Здесь  $G'_i$ , где  $i = 1..Q$ , – примитивы, которые не рассматриваются на данной итерации;
7. В  $G$  существует часть, состоящая из элементов  $G'_b$  для которых невозможно найти  $B''$ . Поэтому после обработки всех модулей в  $G$  осуществляется переход к п. 8 для обработки примитивов, которые не вошли ни в один из рассматриваемых модулей  $G''$  на этапе п. 2;
8. Осуществляется поиск в  $G$  циклов  $C_i \in G$ ;
9. Если для циклов существуют модули  $B'_i$  в базисе БМК, которые выполняют ту же функцию, что и цикл  $F(B'_i) = F(C_i)$ , то циклы заменяются полностью без анализа элементов внутри них;
10. При наличии схем замещения  $B'_i$  для циклов  $C_i$  схема в базисе ПЛИС представляется через множество циклов и примитивов  $G'$ , при этом  $B'_i = \Omega(G'_i)$ , а  $B' = \Omega(G') : G = gU(UG'_i)UG'$ , где  $i = 1..W$ ,  $G' = UG'_b, B' = UB'_i$ , где  $i = 1..Q$ . Примитивы  $G'$  заменяются стандартными подсхемами замещения  $B'_i$ ;
11. При отсутствии схем замещения для модулей  $G'' \in G$  и циклов  $C_i \in G$ , схемы в базисе ПЛИС приводятся к не иерархическому виду, и для каждого примитива  $G'_i$  выполняется следующее преобразование:  $G = U(G'_i)UG$ ,  $B' = UB'_i$ , где  $i = 1..Q$ , а также  $B' = \Omega(G)$ ,

$B'_i = \Omega(G'_i)$ ,  $F(B'_i) = F(G'_i)$ , где  $B'_i$  – схемы замещения для всех примитивов  $G'_i$ , а  $Q$  – их количество в  $G''$ .

Итак, был рассмотрен метод перевода схемы, состоящий из одиннадцати этапов.

**Метод сохранения временных соотношений**

Целью данного метода является сохранение временных соотношений между элементами в базисах ПЛИС и БМК. Известные САПР реализуют алгоритмы на основе данного метода путём оптимизации фрагментов схемы по теоремам булевой алгебры с учётом установленных временных ограничений и специфики выбранного базиса. Однако такая оптимизация в САПР не всегда эффективна для смешанных схем, содержащих синхронные и асинхронные модули. Реализованные в конвертере ConvChip алгоритмы сохранения временных соотношений на практике успешно дополняют существующие инструментальные средства САПР.

Схема в базисе БМК  $B$  считается работоспособной  $F(G) = F(B)$  по отношению к схеме  $G$ , когда между всеми элементами схемы в базисе ПЛИС в базисе БМК сохранены временные отношения. Таким образом, задача, решаемая методом сохранения временных соотношений, сводится к выполнению следующего требования:

$$\forall B' \in B, \forall G' \in G, D(G')/D(B') = K = \text{const} \Leftrightarrow F(G) = F(B). \quad (1)$$

Суть метода состоит в том, чтобы найти такой коэффициент  $K$ , при котором отношение временной задержки  $D$  элемента  $G$  в базисе ПЛИС и его подсхемы замещения  $B$  в базисе БМК для всех элементов схемы было бы постоянным.

Метод сохранения временных соотношений [4, 5] выполняет требование (1) максимально точно. Погрешность выполнения условия (1) зависит только от типа базиса БМК и не зависит от самого метода. Как следует из формулы (1), коэффициент  $K$  должен быть одинаков для всех элементов. Он вычисляется для наихудшего случая, т.е. среди множества отношений  $D(G'_i)/D(B'_i)$ , образуемых всеми элементами схемы:

$$K = \max\{D(G'_i)/D(B'_i)\}. \quad (2)$$

К каждому отношению  $D(G')/D(B')$  добавляется задержка из чётного (!) числа  $Q$  инверторов  $I$ , которые не ме-

няют  $F(B')$ , но максимально приближают это отношение к коэффициенту  $K$  в формуле (2). Иными словами,  $\forall G' \in G$  и задержками «подгоняется» отношение  $D(G')/D(B')$  для выполнения (2). Причём чем меньше величина задержки инвертора  $D(I)$ , тем точнее для всех элементов схемы можно приблизить отношение  $D(G')/D(B')$  к числу  $K$ . Теоретически  $D(G')/D(B') \rightarrow K$  при  $D(I) \rightarrow 0$ . Следовательно:

$$\lim D(I) \rightarrow 0 \{D(G') - [K - D(G')/2D(I)]D(I)\} = 0. \quad (3)$$

Однако на практике  $D(I)$  – величина постоянная, и для всех элементов схемы условие (2) выполнить невозможно. Поэтому задача оптимизации временных соотношений, выполняемая методом сохранения временных соотношений, записывается следующим образом:

$$|D(G')/[D(B') + QD(I)] - \max\{D(G'_i)/D(B'_i)\}| = \min. \quad (4)$$

Однако библиотека в базисе БМК может содержать несколько вариантов инверторов  $I_k$  с разными задержками  $D(I_1) \neq \dots \neq D(I_k) \neq \dots \neq D(I_M)$ . Причём количество таких вариантов  $M$  зависит от типа САПР. Таким образом, соотношение (4) примет вид:

$$|D(G')/[D(B') + \sum Q_k D(I_k)] - \max\{D(G'_i)/D(B'_i)\}| = \min, \quad (5)$$

где  $k = 1..M$ , а сумма  $\sum Q_k$  всегда является чётной.

Множество различных типов инверторов  $I_k$  из  $M$  штук путём перебора  $\forall Q_k$ , где  $k \in [1, M]$ , образуют большое количество вариантов сумм  $S = \sum Q_k D(I_k)$ . При этом процесс перебора всех вариантов может занимать неприемлемо много машинного времени при решении соотношения (5) в конвертере. Для сокращения объёма вычислений вводят следующие ограничения:

- а) количество рассматриваемых в формуле (5) отношений  $D(G')/D(B')$ ,
- б) количество типов инверторов  $M$  в сумме  $S$ ,
- в) количество инверторов  $I_k$  конкретного типа  $k$ , где  $k \in [1, M]$ .

Рассмотрим следующие четыре случая для (а), возникающих при сокращении количества элементов для задачи (5).

- 1. Выбор модуля  $G''$  из схемы в базисе ПЛИС и его замена на модуль в ба-

зисе БМК, т.е.  $G' \in G \rightarrow B'' = \Omega(G'')$ . Для решения задачи (5) необходимо рассмотреть  $q' + q''$  элементов, где  $q'$  и  $q''$  – количество примитивов  $B'$  и модулей  $B''$  в схеме в базисе БМК.

- 2. Для сокращения  $q' + q''$  все примитивы  $e_k$ , входящие в состав модуля  $G''$ , из рассмотрения исключаются. Поэтому задержка в виде инверторов добавляется не к отдельным элементам  $e_1, e_2, \dots, e_q \in G''$ , а ко всему модулю  $G''$ . Таким образом, вместо  $q$ -отношений  $D(e_1)/D(b_1), D(e_2)/D(b_2), \dots, D(e_q)/D(b_q)$ , где  $b_i = \Omega(e_i), \forall i \in [1, q]$ , для формулы (5) рассматривается только одно отношение  $D(G'')/D(B'')$ .
- 3. Если в модуле найден цикл  $C_i \in G''$ , то добавление инверторов выполняется не последовательно к каждому элементу  $e_i \in C_p$  а в цепь обратной связи, состоящей из элементов  $a_i \in C_p$ . Таким образом, отношения  $D(e_1)/D(b_1), D(e_2)/D(b_2), \dots, D(e_q)/D(b_q)$ , где  $b_i = \Omega(e_i), \forall i \in [1, q]$  для (5),  $e_i \neq a_i$  для (5) игнорируются.
- 4. Если модуль  $G''$  не критичен к выполнению временных соотношений (5), то отношение  $D(G'')/D(B'')$  исключается из рассмотрения. Как правило, это различные комбинационные подсхемы.

Итак, вместо  $q' + q''$ , рассматривается теперь количество элементов  $q' + q'' - \Delta_q$  где  $\Delta_q$  – элементы, которые необходимо игнорировать при выполнении какого-либо условия на данном этапе.

Рассмотрим два случая, возникающих при сокращении количества типов инверторов  $M$  (вариант б) для задачи (5).

- 1. Из числа  $M$  исключают инвертор  $I_x$ , задержка  $D(I_x)$  которого может быть вычислена через сумму нескольких других задержек, т.е.  $\exists I_x: D(I_x) = \sum Q_j^* D(I_j)$ , где  $j = 1..q$  при  $x \in [1, M]$  и  $q \leq M$ . Практически поскольку  $D(I_x)$  – дробная величина, то для ещё большего уменьшения количества инверторов  $M$  вводят параметр  $\Delta D \in [\min(D(I_k))/4, \min(D(I_k))^*2], k \in [1, M]$ . Таким образом, все инверторы  $I_x$ , для которых

$$D(I_x) \in [\sum Q_j^* D(I_j) \pm \Delta D], \text{ где } j = 1..q, \quad (6)$$

исключаются из рассмотрения, т.е.  $I_x \in X^*$ . На практике уравнение (6) позволяет сократить  $M$  примерно в два раза. Таким образом, получается множество инверторов  $X^*$ , которые исключаются из рассмотрения.

2. Выбор набора из  $M$  инверторов из библиотеки САПР БМК. Если инвертор только один, то задача оптимизации решается по соотношениям (6). Иначе, при  $M > 1$  используется выражение (5).

И наконец, рассмотрим два случая для варианта (в) – сокращение максимального количества инверторов для задачи (5).

1. При решении задачи (5) необходимо учитывать свободный объём БМК, на основе которого будет выпущена полузаказная микросхема:  $|B| - |UB_i^i| - |UB_i^i|$ , где при  $B_i^i \ i = 1...q'$ , а при  $B_i^i \ i = 1...q''$ . Поэтому для каждого модуля  $B^k \in B$  и примитива  $B^k \in B$ , не входящего ни в один модуль, выделяется  $\theta_k$  процентов свободного ресурса:

$$\theta_k = \left( |B| - \left| \bigcup_{i=1}^{q'} B_i^i \right| - \left| \bigcup_{i=1}^{q''} B_i^i \right| \right) / \left[ \left( \sum_{i=1}^{q'+q''-\Delta q} \max \left\{ \frac{D(G_i^i)}{D(B_i^i)} \right\} - \frac{D(G^i)}{D(B^i)} \right) \times 100\% \right] \left[ \max \left\{ \frac{D(G^i)}{D(B^i)} \right\} - \frac{D(G^k)}{D(B^k)} \right] \quad (7).$$

Фактически чем выше в формуле (7) разница  $\Delta$  между  $D(G^i)/D(B^i)$  и коэффициентом  $K$  из формулы (2) (см. выражение в квадратных скобках), тем больше инверторов необходимо добавить для выполнения соотношения (5). Следовательно, чем больше  $\Delta$ , тем больше  $\theta_k$ . Знаменатель в формуле (7) характеризует суммарный объём всех таких  $\Delta$  для элементов  $q' + q'' - \Delta q$ , рассматриваемых при выполнении соотношения (5). Естественно, что при удачном выборе инверторов  $M$ -типа, часть объёма ( $\Delta\theta_k \in [0, \theta_k]$ ) остаётся невостребованной. Поэтому после выполнения задачи (5) для текущего отношения  $D(G^i)/D(B^i)$  эта часть объёма  $\Delta\theta_k$  распределяется между  $\theta_1, \theta_2, \dots, \theta_x$  для не вычисленных отношений. Таким образом,  $\theta_k$  накладывает ограничение на максимальное количество  $\Sigma Q_k$  инверторов  $I_k$ , где  $k = 1...M$ , которые можно использовать для одного элемента при выполнении соотношения (5).

2. Чем длиннее выбирается цепочка из инверторов  $I_k$  для элемента  $B$ , тем ниже флуктуационная устойчивость этой цепочки. Следовательно, чем больше  $\Sigma Q_k$ , где  $k = 1...M$ , в (5) для некоторого отношения  $D(G)/D(B)$ , тем больше РВД появится

ся при флуктуационном моделировании схемы в базисе БМК. Здесь под флуктуационным моделированием понимается моделирование, выполняемое с псевдослучайным разбросом временных параметров. При флуктуационном моделировании обычно генерируется несколько вариантов схемы и несколько вариантов запаздывания входного воздействия. Для выбора конкретных значений уменьшения задержек сигналов и запаздываний входных воздействий используются генераторы случайных чисел, имеющие линейное распределение вероятности в заданном интервале.

На практике на максимальное количество инверторов накладывается дополнительное ограничение  $\Sigma Q_k < 1$ . На практике  $1 \in (2, 4, 6, 8, 10, 12, 14)$ , причём выбор  $1$  выполняется с учётом условия, накладываемого  $\theta_k$ .

После выполнения задачи (5) необходимо выполнить оптимизацию результатов. Для этого рассматривается элемент  $G^*$  из схемы в базисе ПЛИС и подсхема его замещения  $V^*$ , вычисленная по методу перевода  $\Omega$ .

Если для  $G^*$  существует две подсхемы замещения –  $V^*$  и  $V^{**}$ , причём для замены выбрана  $V^*$ , т.е.  $G^* \in G \rightarrow V^* = \Omega(G^*)$  и  $V^{**}$  более оптимально выполняет (5), чем  $V^*$ , то  $V^* = \Omega(G^*)$  можно заменить на  $V^{**} = \Omega(G^*)$ . В таких ситуациях наблюдается противоречие. С одной стороны,  $V^*$  может содержать меньше элементов, чем  $V^{**}$ , или иметь более высокое быстроедействие и флуктуационную устойчивость, чем  $V^{**}$  (см. условия для  $|V^*|, |V^{**}|, ||V^*||$  в методе перевода). С другой стороны, для  $V^{**}$  задача (5) выполняется более оптимально, т.е.  $V^{**}$  более точно позволяет приблизить  $D(G^*)/D(B^*)$  к параметру  $K$  из (2), чем отношение  $D(G^*)/D(B^*)$  (см. описание параметров  $\lambda$  и  $\theta_k$ ). В таких ситуациях теоретически нельзя ответить на то, какая подсхема –  $V^*$  или  $V^{**}$  – лучше. Поэтому выбор остаётся за специалистом.

В методе сохранения временных соотношений проводится также оптимизация результатов выполнения задачи (5). Для этого анализируется следующее условие. Если полученную сумму задержек для выравнивания отношения  $D(G^*)/D(B)$ , где  $B = \Omega(G^*)$ , можно разложить на ряд слагаемых  $X_j$ , величина которых совпадает с одним из типов инверторов из множества  $X^*$ , исключённых из рассмотрения, то часть инверторов  $I_j$ , где  $j \in [1, Q_i]$ ,

входящих в  $X_j$ , заменяется на один инвертор  $I \in X^*$ . Запишем это условие:

$$\begin{aligned} & \Sigma Q_k^* D(I_k) \text{ [где } k = 1...M] = \\ & = \Sigma Q_k^* D(I_k) \text{ [где } k = 1...Q_1] + \\ & + \Sigma Q_k^* D(I_k) \text{ [где } k = Q_1 + 1...Q_2] + \\ & \dots + \Sigma Q_k^* D(I_k) \text{ [где } k = Q_i + 1...Q_j] = \\ & = X_1 + X_2 + \dots + X_q \end{aligned} \quad (8)$$

При этом необходимо иметь в виду следующие соотношения:  $D(I) = X_j = \Sigma Q_k^* D(I_k)$ , где  $k = Q_{j-1}...Q_j$ ,  $M = \Sigma Q_k$ , где  $k = 1...q$ , причём  $\exists X_j, j \in [1, q]$  и  $\exists D(I) \in X^*$ . При выполнении приведённых условий инверторы  $I_k \in [Q_{j-1}, Q_j]$  заменяются на инвертор  $I$  (описание множества  $X^*$  приведено после формулы (6)).

### ПОСЛЕСЛОВИЕ

Не рассмотренные в статье методы диагностики позволяют оценить поведение элементов схемы на основе результатов моделирования, полученных на внутренних связях схемы. К этой группе относятся разнообразные методы, применяемые для сравнения схемы до и после конвертирования. Целью выполнения алгоритмов, реализованных в ConvChip на основе методов диагностики, является анализ схемы на возникновение различных схемотехнических проблем. На сегодняшний день алгоритмы на основании методов диагностики, реализованные в современных САПР, не позволяют получить адекватный анализ схем, частично содержащих асинхронные модули.

### ЛИТЕРАТУРА

1. Лахов А. Обзор средств функциональной верификации компании Mentor Graphics. Современная электроника. 2005. № 5.
2. Денисов А.Н., Коняхин В.В., Гаврилов С.В. Средства оперативной разработки полузаказных БИС. Актуальные проблемы твердотельной электроники и микроэлектроники. Таганрог, 2002.
3. Денисов А.Н., Коняхин В.В. Тезисы докладов IV научно-практической конф. «Проблемы обеспечения изделий авиационной и ракетно-космической отрасли высококачественной элементной базой». Сочи, 2003. МНТО РЭС им. А.С. Попова, Москва, 2003.
4. Артёмов С.А. Методы обработки схем при конвертировании в базис базовых матричных кристаллов. Оборонный комплекс научно-техническому прогрессу России. 2005. № 3.
5. Артёмов С.А. Конвертор для перевода проектов схем из САПР ПЛИС в САПР БМК. Изв. вузов. Электроника. 2006. № 2.