

СПИСОК ЛИТЕРАТУРЫ

1. Poage J. F. Derivation of optimum tests to detect faults in combinational circuits // *Mathematical Theory of Automata*.— N. Y.: Polytechnic Press, 1963.— P. 483.
2. Armstrong D. B. A deductive method for simulation faults in logic circuits // *IEEE Trans.*— 1972.— C-21, N 5.— P. 464.
3. Ulrich E. G., Baker T. // *Computer*.— 1974, April.— P. 39.
4. Brglez F., Pownal P., Hum R. Application of testability analysis: from ATPG to critical delay path tracing // *International Test Conference*.— 1984.— P. 705.
5. *IEEE Des & Test Comput.*— 1985.— 2, N 1.— P. 38.
6. *Радиоэлектроника* (состояние и тенденции развития).— 1986.— № 2.
7. Лобов И. Е., Межов В. Е., Чевычелов Ю. А. Логическое моделирование и генерация тестов цифровых схем в системе «Кулон» // Школа-семинар молодых ученых и специалистов «Актуальные проблемы создания интеллектуальных САПР, РЭА и СБИС»: Тез. докл.— М., 1989.
8. Киркленд Т., Флорес В. Программируемые средства анализа тестируемости и автоматическая генерация тестов для СБИС // *Электроника*.— 1983.— № 5.
9. Сергеев А. А. Алгоритм выделения повторносходящихся и циклических путей в схемном графе // *Вопросы радиоэлектроники*.— 1977.— № 11.
10. Putzolu G. R., Roth J. P. A heuristic algorithm for testing of asynchronous circuits // *IEEE Trans.*— 1974.— C-23, N 10.— P. 1078.

Поступила в редакцию 27 июля 1990 г.

УДК 681.621.375

З. А. ЛИВНИЦ, А. В. ПИЧУЕВ

(Новосибирск)

SimSim: ПРОГРАММА ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ МОП БИС НА ПЕРЕКЛЮЧАТЕЛЬНОМ УРОВНЕ

Введение. Поведенческое моделирование электронных схем является основным способом проверки соответствия проекта схемы исходным спецификациям ее функционирования и выявления допущенных ошибок. Хорошо известно, что при разработке больших и сверхбольших интегральных схем именно тщательное моделирование позволяет избежать увеличения количества дорогостоящих и времязатратных производственных итераций, выполняемых до получения работоспособных кристаллов. С этой точки зрения особое значение имеет использование моделирования на этапе верификации топологии. Здесь исходной информацией для программ моделирования является описание принципиальной электрической схемы, экстрагированное с применением специальных программных средств из топологического описания разрабатываемого кристалла (при этом имеется возможность варьирования степени подробности этой информации). В течение длительного времени основными были два (в известной мере полярных) подхода: электрическое моделирование, т. е., по существу, решение систем дифференциальных уравнений, выражающих законы Кирхгофа и Ома (пример — широко распространенная программа SPICE [1]), и чисто логическое моделирование, опирающееся на представление схемы на вентиляльном уровне (типичный пример — программа TEGAS [2]).

Средства первого из указанных типов гарантируют исследование поведения анализируемой схемы с исчерпывающей точностью, однако необходимые временные затраты, связанные с решением систем уравнений высокой размерности, позволяют в настоящее время практически применять их лишь при моделировании схем сложностью до нескольких сотен транзисторов.

С использованием представления вентиляльного уровня удастся создавать весьма быстрые алгоритмы моделирования событийного типа, про-

производительность которых может быть еще значительно увеличена за счет применения специализированных аппаратных акселераторов. В то же время выяснилось, что при моделировании МОП БИС описание схем на вентиляльном уровне является не вполне адекватным: оно малоприспособно для кристаллов с большим удельным весом проходных транзисторов и не учитывает ряд важных эффектов, характерных для МОП-схем.

На рубеже 80-х годов появились работы [3—6], в которых была сделана попытка достичь компромисса между точностью моделирования, обеспечиваемой «аналоговыми» методами, и производительностью, отличающей «логические» методы. Предложенный в них подход восходит к ранним работам К. Шеннона по теории релейно-контактных схем [7]. Анализируемая схема представляется в рамках этого подхода как сеть, элементами которой являются транзисторы (рассматриваемые как переключатели, управляемые сигналами на их затворах), соединяющие схемные узлы. Такое описание позволяет использовать, как и в [7], аппарат алгебры логики, однако поскольку МОП-схемы являются более сложными объектами по сравнению с релейно-контактными, то применительно к ним потребовалось дальнейшее развитие идей Шеннона: основная особенность «переключательного» уровня моделирования МОП БИС — учет узловых емкостей и транзисторных сопротивлений, который осуществляется введением некоторых дискретных величин, задающих «силу» сигналов.

Разработанный Р. Брианом симулятор MOSSIM [3, 4] положил начало активному внедрению методов анализа схем на переключательном уровне в практику проектирования СБИС, а эффективность такого подхода стимулировало появление целого ряда новых работ, направленных как на совершенствование алгоритмов логического моделирования, так и на расширение сферы использования данного уровня описания схем (в частности, для моделирования неисправностей, генерации тестов, верификации временных соотношений и т. п.). Указанные работы различаются в основном применяемыми схемотехническими моделями, способами декомпозиции схем на компоненты, процедурами вычисления установившихся состояний. обстоятельные обзоры результатов, полученных в данном направлении, можно найти в [8—10].

Поскольку опыт разработки специализированных кристаллов в лаборатории автоматизированного проектирования заказных СБИС ИАиЭ СО АН СССР убедил нас в несомненной практической полезности средств моделирования на переключательном уровне, то было признано целесообразным создание собственного симулятора. При этом преследовались две основные цели: обеспечить повышенную скорость анализа «типичных» схем (и, следовательно, увеличение уровня интерактивности), а также максимально естественную «вписанность» симулятора в используемый нами технологический маршрут проектирования интегральных схем. Для достижения этих целей в разработанной программе SimSim (Simple Simulator) заложены следующие принципы: организация базы данных с использованием описания схемы в формате esim [8] (для обеспечения совместимости с наиболее распространенными программами экстракции); динамическое разбиение схемы на компоненты связности (для уменьшения затрат памяти и сокращения времени обхода событий); применение конструкции локальной цепи (для понижения размерности задач вычисления установившегося события, решаемых итеративными методами).

Конкретизации этих принципов и описанию наиболее существенных особенностей реализации программы SimSim посвящен следующий раздел статьи.

Программа моделирования: структуры данных, модели, алгоритмы. База данных. Важным практическим преимуществом описания схем на переключательном уровне является тот факт, что это описание однозначно определяется топологией схем (это не имеет места для вентиляльных представлений). Входной информацией для программы SimSim слу-

жит файл в символьном формате esim — стандартном выходном формате программ, экстрагирующих принципиальную электрическую схему из топологии. Каждая строка такого файла задает либо транзистор, либо емкость узла схемы, причем для определения транзистора указываются его тип (n -МОП/ p -МОП, обогащенный/обедненный и т. п.), а также имена (номера) узлов, к которым подсоединены его затвор, исток и сток.

Поскольку специфика программы SimSim требует обеспечения быстрого нахождения для произвольного узла всех подсоединенных к нему транзисторов, была использована специальная организация данных о логической сети, состоящей из транзисторов и узлов. База данных представляет собой набор многосвязных списков. В описании каждого узла имеются три ссылки: на первые элементы списков транзисторов, подсоединенных к этому узлу соответственно затвором, истоком и стоком. В описании каждого транзистора шесть ссылок: на узлы, к которым он подсоединен затвором, истоком и стоком, а также на транзисторы, следующие за ним в списках, соответствующих этим узлам.

Кроме того, в описании каждого транзистора хранятся данные о его типе и сопротивлении в открытом состоянии, а для каждого узла — величина его емкости и «текущее» логическое состояние, а также некоторая переменная служебная информация, используемая при работе алгоритма моделирования.

Модель сигнала. Предполагается, что каждый узел схемы находится в одном из трех логических состояний: «1» (высокий уровень), «0» (низкий уровень) или X (неопределенный или неизвестный уровень). Наряду с состояниями в SimSim, так же, как и в других программах, оперирующих на переключательном уровне, учитывается сила сигналов, «конкурирующих» в приведении узла в определенное состояние. Принятые в SimSim соглашения, по существу, эквивалентны рассмотренной в [8] дистрибутивной решетке пар «сила — состояние», в которой используются четыре градации силы сигналов (в порядке убывания): управляемая, слабая, сверхзарядная и зарядная. Первые две из них относятся к сигналам, поступающим в узел от источников тока соответственно через транзисторы с «малым» или «большим» сопротивлением; вторые две — к сигналам, возникающим в результате накопления заряда (соответственно «очень большой» емкости или «обычной» емкости). Основное отличие заключается в том, что в SimSim характеристики сигналов вычисляются не для отдельных узлов, а для некоторых их объединений — так называемых локальных цепей.

События. SimSim является симулятором событийного типа. Здесь событие — это изменение сигнала на входе схемы или на затворе любого транзистора. Событие меняет связность некоторой части схемы, распространение сигналов по которой приводит к появлению новых событий, и т. д. Таким образом, моделирование состоит в генерации воздействий на входные узлы схемы, обработке последовательности стимулированных ими событий (каждое событие, являющееся следствием предыдущих, регистрируется, ставится в очередь и затем «обсчитывается», т. е. вычисляются обусловленные им изменения в функционировании схемы) и наблюдении состояния определенных (например, выходных) узлов.

Из изложенного выше ясно, что специфика переключательного уровня описания касается двух основных вопросов: определения подсхемы, которая должна быть проанализирована в связи с «обсчетом» конкретного события, и организации вычислений новых установившихся состояний узлов этой подсхемы.

Связные динамические компоненты. Рассматриваемая на переключательном уровне схема может быть представлена неориентированным графом G , вершины которого соответствуют узлам схемы, а две вершины соединяются ребром в том и только в том случае, если существует транзистор, у которого один из этих узлов является истоком, а другой — стоком. Будем помечать ребра символами «1», «0» или « X »

в зависимости от состояния сигналов на затворах транзисторов («1» соответствует открытому транзистору, «0» — закрытому, «X» — находящемуся в неопределенном состоянии). Граф G характеризует «текущую» связность схемы.

Рассмотрим теперь некоторую вершину n_0 графа G . Образует подграф G_{n_0} следующим образом: вершинами G_{n_0} будут все вершины графа G , соединенные с n_0 путями, помеченными символами «1» и «X», а ребрами G_{n_0} — все ребра, помеченные символами «1» и «X» и соединявшие в G вершины из G_{n_0} . Графу G_{n_0} соответствует максимальная подсхема, содержащая узел n_0 , все узлы которой соединены путями, проходящими через стоки и истоки «незакрытых» транзисторов. Такую подсхему мы будем называть связной динамической компонентой.

В этих терминах обсчет некоторого схемного события, например изменения состояния сигнала на затворе некоторого транзистора, сводится к вычислению установившегося состояния узлов связных динамических компонент, содержащих сток и исток данного транзистора (эти компоненты могут быть слиты, например, если транзистор открылся). При этом следует отметить, что для определения новых состояний не требуется никакой «внешней» информации (т. е. данных о сигналах в узлах вне компоненты).

Подобный учет текущей связности схемы является одним из существенных средств понижения размерности многократно решаемой задачи — расчета вызываемых событием изменений сигналов. Поэтому несмотря на известные вычислительные затраты, связанные с необходимостью анализа связности при обсчете каждого события, такой подход обеспечивает значительные преимущества по сравнению со статическим разложением исходной схемы на подсхемы.

Локальные цепи. Еще один способ понижения размерности заключается в том, чтобы вместо отдельных узлов динамической компоненты рассматривать входящие в ее состав локальные цепи. Здесь под локальной цепью, содержащей некоторый узел, понимается совокупность всех узлов*, соединенных между собой путями, проходящими через истоки и стоки открытых транзисторов с малым сопротивлением («ключевых» транзисторов); по определению ясно, что установившееся состояние одинаково для всех узлов, входящих в локальную цепь.

Очевидно также, что каждая динамическая связная компонента может быть представлена в виде совокупности локальных цепей, соединенных между собой транзисторами с «большим» сопротивлением («нагрузочными» транзисторами) либо транзисторами с неопределенным значением сигнала на затворах («неопределенными» транзисторами).

Такое разложение динамической компоненты просто осуществляется с помощью рекурсивной процедуры обхода графа текущей связности, начинающейся с инициализированного узла (например, стока или истока транзистора, на затворе которого изменился сигнал).

Векторы характеристик путей. При расчете состояния локальной цепи приходится различать случаи, когда сигнал в цепи — результат протекания тока в узел или из узла (т. е. имеется путь из незакрытых транзисторов, связывающий цепь по крайней мере с одним из узлов VDD или GND) — либо когда цепь изолирована от источников тока и сигнал в цепи определяется наличием зарядов. В первом случае достаточной для вычисления состояния является информация о сравнительной «силе», с которой действуют на локальную цепь узлы VDD и GND. Эти силы, в свою очередь, зависят от свойств путей, связывающих цепь с этими узлами.

С учетом вышеизложенного в SimSim при образовании каждой локальной цепи ей ставится в соответствие двоичный вектор характеристик

* За исключением узлов питания VDD и GND, которые (по понятным причинам) не входят в состав никакой локальной цепи.

путей, который последовательно уточняется в процессе анализа динамической компоненты связности (см. п. «Релаксация»).

Вектор характеристик путей 8-битовый (по 4 бита для VDD и GND); каждый бит указывает на наличие пути определенного типа.

Типы путей упорядочены (в порядке убывания «силы») следующим образом:

- путь из открытых ключевых транзисторов;
- путь из открытых ключевых транзисторов и ключевых транзисторов с неопределенным сигналом на затворе;
- путь, содержащий открытые ключевые и нагрузочные транзисторы;
- путь, содержащий открытые ключевые и нагрузочные транзисторы, а также транзисторы с неопределенным состоянием.

Ясно, что ненулевое установившееся значение вектора характеристик полностью определяет логическое состояние сигнала в локальной цепи («0», «1» или «X»); правила вычисления состояния совершенно очевидны.

Релаксация. При формировании локальных цепей для каждой из них выясняется, имеет ли данная цепь непосредственную связь (т. е. путь, не проходящий через другие локальные цепи данной динамической компоненты) к узлам VDD и GND; таким образом определяются начальные значения векторов характеристик путей. Если окажется, что все такие векторы нулевые, то это будет означать, что динамическая компонента изолирована от источников тока и требуется процедура «деления заряда» (см. ниже). При наличии ненулевых векторов характеристик осуществляется процедура релаксации.

По существу, эта итеративная процедура работает над графом, вершины которого соответствуют узлам VDD, GND, а также «макроузлам», представляющим локальные цепи, входящие в динамическую компоненту, а ребра — соединяющим транзисторам. Алгоритм релаксации последовательно проходит ребра этого графа, причем если с помощью транзистора, соединяющего две локальные цепи, образуется путь, не учтенный в текущем векторе характеристик, то этот вектор соответственно обновляется; обход продолжается до тех пор, пока все векторы не окажутся в «установившемся состоянии».

Необходимо отметить, что при моделировании реальных схем процедура релаксации сходится очень быстро, что связано с небольшими размерами получающихся графов.

Деление зарядов. Как уже отмечалось, может оказаться, что для расчета состояния сигналов в динамической компоненте схемы потребуется информация о распределении в ней зарядов. Поэтому при формировании локальных цепей, наряду с векторами характеристик путей, определяются также начальные значения вектора характеристик зарядов. Это 6-битовый вектор, битам которого соответствует информация о наличии в цепях узлов, хранящих заряд (или «сверхзаряд») и имеющих состояние «0», «1» или «X».

Процедура деления заряда между такими локальными цепями, соединенными нагрузочными или «неопределенными» транзисторами, осуществляется стандартным образом (по этому поводу см., например, [8]).

Реализация. Программа SimSim реализована на языке C в операционной среде VAX/VMS. Она обеспечивает моделирование схем, выполненных как в *n*-МОП, так и в КМОП технологии.

Заключение. В течение 1990 г. программа прошла практическую проверку при разработке и верификации специализированных кристаллов различной сложности (от 5000 до 30 000 транзисторов) и показала достаточно высокие эксплуатационные качества.

СПИСОК ЛИТЕРАТУРЫ

1. Vladimiresen A., Lin S. The simulation of MOS integrated circuits using SPICE // Memo VCB/ERLM.— Berkeley: Univ. of Calif., 1980: 80/7.
2. Szygenda S. A. TEGAS2 — anatomy of a general purpose test generation and simulation system for digital logic // Proc. 9th Des. Aut. Workshop, 1972.
3. Bryant R. E. An algorithm for MOS logic simulation // Lambda.— 1980.— 4th qtr.— P. 46.
4. Bryant R. E. MOSSIM: a switch-level simulator for MOS LSI // Proc. 18th Design Automation Conf., 1981.— P. 786.
5. Baker C. M., Terman C. S. Tools for verifying integrated circuit design // Lambda.— 1980.— 4th qtr.— P. 22.
6. Hayes S. P. A unified switching theory with applications to VLSI design // Proc. IEEE.— 1982.— P. 1140.
7. Шеннон К. Символический анализ релейных и переключательных схем // Работы по теории информации и кибернетике.— М.: Изд-во иностр. лит., 1963.
8. Ульман Дж. Д. Вычислительные аспекты СВИС.— М.: Радио и связь, 1990.
9. Hayes S. P. An introduction to switch-level modeling // IEEE Design and Test of Computers.— 1987.— 4, N 4.— P. 18.
10. Bryant R. E. A survey of switch-level algorithms // Ibid.— P. 26.

Поступила в редакцию 6 февраля 1991 г.

УДК 681.3.06

А. В. ИОФФЕ
(Новосибирск)

ПОСТПРОЦЕССОРЫ ГРАФИЧЕСКОГО ВЫВОДА В СИСТЕМЕ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ БИС

Несмотря на широкое распространение интерактивных методов проектирования интегральных схем, твердые копии изображения топологии остаются важным и весьма информативным материалом, активно используемым разработчиками БИС. Поэтому в каждую систему автоматизированного проектирования включаются программные средства для получения чертежей топологии.

В данной статье кратко рассмотрены разработанные в ИАПЭ СО АН СССР постпроцессоры графического вывода, ориентированные на задачи прорисовки топологии. Эти программы работают в операционной среде VAX/VMS и поддерживают выводные устройства различных типов, включая как перьевые графопостроители (в нашей системе — «Планист» [1]), так и растровые устройства (лазерные принтеры — Lx03, «СО₂-РОМБ» [2]). Входной информацией для постпроцессоров служат текстовые файлы в формате CIF [3]. CIF широко применялся для описания топологии во многих системах автоматизированного проектирования 80-х годов. Кроме того, были написаны трансляторы для преобразования в CIF топологических файлов, представленных в других форматах (например, файл в формате SOURCE, полученный на распространенном комплексе «Кулон-1», можно перевести в формат CIF с использованием специальной утилиты QULTOCIF).

Ниже описаны некоторые особенности реализации программ вывода топологической информации на устройства указанного типа, наиболее важные с точки зрения достижения практической эффективности.

Программа CIF_PEN — прорисовка топологии на графопостроителе. В настоящее время графопостроители достаточно широко применяются как для послойных прорисовок, так и для изготовления «совмещенных» чертежей кристаллов. Однако сравнительно невысокая скорость движения пишущего узла (~1 м/с) и значительный объем графических данных приводят к тому, что даже для относительно небольших (~10⁴ транзисторов) БИС процесс прорисовки является многочасовым. Поэтому при разработке программы особое внимание уделялось средствам для сокращения длительности создания чертежа и нейтрализации нежелательных эффектов, связанных с большим временем рисования.