

Рис. 2

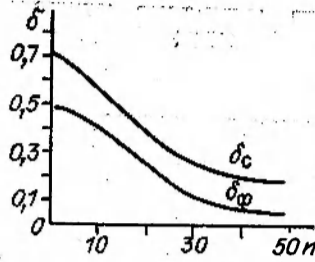


Рис. 3

Функция (8) задает больший вес периферийным точкам искаженного сигнала как более информативным по отношению к силуэту восстанавливаемого сигнала. Для удовлетворительного восстановления сигнала и искажающей функции (рис. 2, а, б) потребовалось 50 шагов двойных итераций. На рис. 3 показана зависимость от числа итерации n отклонений нормированных по энергии сигнала и искажающей функции δ_c и δ_ϕ от истинных. При использовании этого метода, как и всех градиентных методов, существенным оказывается наличие хорошего начального приближения. В рассматриваемом случае в качестве начального приближения для восстанавливаемого сигнала использовалась дискретная функция, равная 1 в области, где, по предположению, восстанавливаемый сигнал может быть отличен от нуля. При моделировании было выявлено, что лучшие результаты восстановления имеют место для сигналов, в пространственном спектре которых преобладают низкочастотные составляющие. Кроме того, восстановление комплексных $S_1(i)$ и $S_2(i)$ менее эффективно, чем положительных $S_1(i)$ и $S_2(i)$. Результаты моделирования подтвердили, что для восстановления двумерных пространственно-ограниченных сигналов можно использовать градиентные методы при целевой функции, равной взвешенному квадрату отклонения искаженного сигнала от свертки восстанавливаемых сигнала и искажающей функции.

ЛИТЕРАТУРА

1. Бакалов В. П., Русских Н. П. О возможности решения уравнения свертки при известном ядре в случае многомерных пространственно-ограниченных сигналов // Автометрия.— 1985.— № 5.
2. Бакалов В. П., Мартынов Ю. Ю., Русских Н. П. Цифровой алгоритм восстановления пространственно-ограниченного сигнала по свертке с неизвестной искажающей функцией // Автометрия.— 1988.— № 1.
3. Рейлейтис Э., Рейвиндран А., Рэгсдел К. Оптимизация в технике.— М.: Мир, 1986.— Кн. 1.

Поступило в редакцию 28 июля 1987 г.

УДК 681.3.06

В. Г. СЕЛИВАНОВ
(Томск)

АДАПТАЦИЯ СИСТЕМЫ МОДЕЛИРОВАНИЯ К УСЛОВИЯМ КОНКРЕТНЫХ СХЕМ РАДИОЭЛЕКТРОННОЙ АППАРАТУРЫ

В процессе разработки радиоэлектронной аппаратуры (РЭА) возникает необходимость создавать макет проектируемого аппаратного модуля, чтобы проверить рабочие параметры (задержка, синхронизация и т. д.), произвести отладку и внести в схему возможные изменения. Этап макетирования требует значительных затрат времени и ресурсов. В то же время с помощью системы логического моделирования работы электронных схем можно отследить прохождение сигналов, величины задержек и получить много другой информации, не делая макета [1, 2]. Таким образом может быть устранен целый этап в технологической цепочке разработки производства новых устройств РЭА. Задача создания таких моделирующих систем для верификации разрабатываемых проектов актуальна и привлекает внимание специалистов в этой области [3—6], особенно сложной здесь является временная верификация [7].

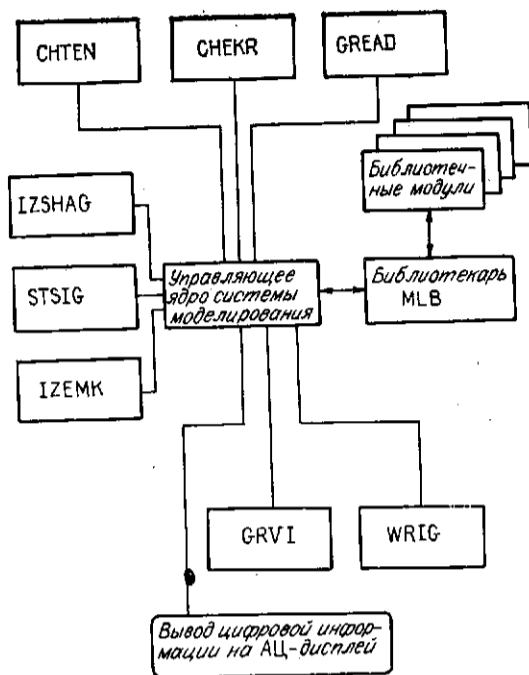


Рис. 1

Использование временной разреженности и событийного планирования во времени повышает быстродействие и эффективность моделирующих программ [8]. Применение систем моделирования при проектировании устройств РЭА уменьшает затраты времени и труда. Эффективность этих систем была бы еще большей при автоматизации процедуры адаптации модели к составу и параметрам конкретной электронной схемы.

Ниже описывается комплекс программных средств, обеспечивающий такую адаптацию системы графического моделирования [9]; в алгоритме моделирования использованы методы приращения времени, формы сигнала и временной разреженности, а также производится событийное планирование.

Структура системы моделирования. Способ построения и функционирования системы детально описан [9], поэтому дадим только краткое описание структуры системы моделирования. На рис. 1 схематично дано представление о составе и взаимодействии программных модулей системы. Подпрограмма CHTEN ввода списков цепей и имен элементов формирует записи, содержащие ин-

формацию о связях и элементах, необходимую управляющему ядру системы, которое после ввода информации о топологии схемы строит кольцевую очередь событий и в соответствии с ней делает запросы библиотекарю.

В зависимости от избранного пользователем режима к системе подключаются либо модуль GREAD, осуществляющий ввод графической информации о топологии электронной схемы, либо программа CHEKR ввода координат элементов, определяющих положение рисунков на экране алфавитно-цифрового (АЦ) дисплея.

Программа-библиотекарь MLB формирует обращения к библиотечным модулям [10], передает им входные параметры и возвращает ядру системы выходные значения сигналов после завершения работы библиотечных модулей.

В зависимости от режима работы система выводит графическую информацию о работе схемы либо на экран графического дисплея [11] с помощью модуля GRVI, либо на экран АЦ-дисплея с помощью программы WRIG. Параллельно на экран другого АЦ-дисплея можно выводить цифровые значения параметров сигналов. В процессе работы на параметры системы необходимо воздействовать с помощью подпрограмм изменения шага кольцевой очереди IZSHAG, установки сигналов на разъемах STSIG, изменения параметров элементов типа конденсатор и резистор IZEMK.

В структуре системы не предусматривается никакой подстройки системы под моделируемую электронную схему. Для того чтобы пользователь мог построить требуемую модель, ему необходимо провести цикл подготовительных операций: написать программы описаний (ПО) разъемов с соблюдением всех требований к их интерфейсу с системой моделирования [10]; провести корректировку модуля библиотекаря и блока внешних описаний заголовков библиотечных модулей в файле описания глобальных параметров, а также файла описания структуры перекрытий программных секций для построителя заданий; сделать перетрансляцию всех требуемых для схемы модулей и произвести построение задания.

Выполнение этих работ требует определенных навыков в программировании, знания в достаточном объеме структуры и принципов работы системы и, кроме того, довольно больших временных затрат, что существенно снижает эффективность системы моделирования как рабочего инструмента разработчика электронных схем.

Для устранения этих недостатков создан описываемый здесь комплекс программ адаптации системы графического моделирования к требованиям конкретных электронных схем.

Построение ПО разъемов и включение их в систему. Разъемы схемы моделируются по тем же правилам, что и радиоэлементы, поэтому для каждого из разъемов необходимо составить программу — описание его функций. Информацию о разъемах и их связях система получает из списков цепей и имен элементов, являющихся результатом работы транслятора графических изображений схем [12]. В соответствии с требованиями к интерфейсу с системой моделирования эти ПО должны по запросу первого уровня сообщить системе количество и номера своих

входов и выходов, а по запросам второго уровня — характеристики сигналов на выходах элемента [9]. Для построения ПО разъемов применяется метод использования заготовок (прообразов) [13]. На магнитном диске хранится файл, в котором находится написанный на языке Паскаль прообраз модели разъема. Программа построения разъемов считывает этот файл, по входной информации о связях и взаимосвязях элементов дополняет его описаниями входов и выходов и формирует новый файл с именем, соответствующим имени разъема в схеме. Таким образом, можно делать любое требуемое количество таких ПО разъемов.

Построенные ПО должны быть известны системе моделирования. Для этого их имена вносятся в секцию глобальных параметров, корректируются список обращений системы к моделям элементов в библиотеке и файл описания перекрытий программных секций. Для решения первых двух задач формируются два файла описаний соответственно перечисленным задачам. Имена этих файлов заранее определены в системе моделирования. В секцию глобальных параметров и в список обращений библиотеки соответствующие файлы включаются с помощью директивы INCLUDE языка Паскаль на этапе трансляции. Для корректировки структуры перекрытий программных секций используется прообраз файла описания перекрытий и формируется временный модуль на диске.

Тестирование моделей элементов. В процессе создания библиотеки элементов [10] выяснилось, что в составе системы моделирования необходимо средство, с помощью которого с минимальными затратами времени на адаптацию системы можно было бы отлаживать и тестировать впервые построенные ПО включаемых в библиотеку моделей элементов. Для решения этой задачи создана специальная программа настройки системы на тестирование модели одного элемента, которая запрашивает имя элемента, обрабатывает исходный текст его программной модели и определяет количество и номера входов и выходов. Для имитации входа — выхода схемы формируется элемент типа разъем, при этом используются данные о выводах элемента, полученные в результате обработки исходного текста тестируемой модели. Полятно, что в данном случае не вводятся списки имен и связей и программа строит их автоматически.

Так как при тестировании предусматривается вывод картинок элемента и разъема на АЦ-дисплей, чтобы можно было видеть изменения сигналов на выходах в зависимости от сигналов на входах, то для элемента, разъема и их выводов рассчитываются координаты, которые они будут иметь на экране АЦ-дисплея.

В результате в дополнение к сформированным спискам цепей и имен элементов и к списку координат создаются корректировочный файл с именем разъема для секции глобальных параметров системы и дополнение в список обращений библиотеки.

Окончательное определение системных параметров и включаемых в систему модулей производится следующим образом: просматривается список имен элементов схемы и формируется набор системных обращений в исходном тексте библиотеки. При этом используется прообраз файла библиотеки: он читается, в нужных местах производятся необходимые подстановки и создаваемый таким образом новый файл записывается на диск в качестве исходного текста библиотеки генерируемой версии системы моделирования.

Аналогично, используя временный файл, строится модуль описания перекрытий программных секций системы. Секция глобальных параметров дополняется с помощью директивы INCLUDE. Затем формируется косвенный командный файл трансляции отобранных модулей ПО элементов.

Таким образом, подстроенная под конкретную схему версия системы моделирования формируется практически автоматически. Состав и параметры генерируемой версии системы определяются в основном в результате анализа входной информации о схеме, подготавливаемой транслятором графических изображений. Программы комплекса, обеспечивающего адаптацию системы моделирования к составу и структуре конкретной схемы, написаны на языке Паскаль. Время, затрачи-

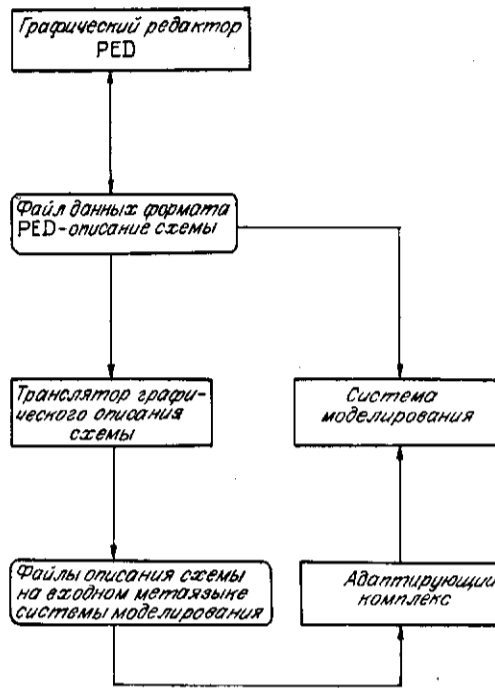


Рис. 2

ваемое на отработку заданий комплекса на ЭВМ СМ-4 в операционной среде ОС — РВ, 20—25 мин.

Комплекс адаптирующих программ, система моделирования работы электронных схем, транслятор графических изображений схем и редактор графических изображений PED [14] обеспечивают автоматизацию процесса ввода информации в систему моделирования и настройку системы на требования конкретной модели. На рис. 2 показана схема взаимодействия этих программных средств. Стрелками обозначены направления движения информации.

Заключение. В результате создания цепочки программных средств для ввода информации и автоматизированной адаптации системы моделирования к требованиям конкретных схем РЭА существенно упростилась процедура работы с системой, снизились трудозатраты и, самое главное, оказались практически снятыми требования к квалификации пользователя как программиста. Теперь для работы с системой разработчик должен знать только команды редактора PED и имя командного файла, адаптирующего и запускающего процесс моделирования. Важным и нужным средством оказались генерируемые версии системы для тестирования и отладки вновь построенных моделей радиоэлементов. Однако создание этих моделей требует навыков программирования и знания принципов построения системы и обработки системных параметров, что можно считать недостатком данного способа моделирования.

ЛИТЕРАТУРА

1. Хэтчел Г. Д., Санджованни-Винчелли А. Обзор методов моделирования третьего поколения // ТИИЭР.— 1981.— 69, № 10.
2. Ниссен К. Методология и средства иерархического проектирования СВИС // ТИИЭР.— 1983.— 71, № 1.
3. Chawla B., Gummel H. K., Kozah P. MOTIS — a MOS timing simulator // IEEE Trans. Circuit. Syst.— 1975.— CAS-22.— P. 301.
4. Tanabe M., Nakamura H., Kawakita K. MOSTAP: An MOS circuit simulator for LSI circuits // Proc. IEEE Int. Symp. Circuits Syst.— Houston, 1980.
5. Lelarsmee E., Sangiovanni-Vincentelli A. RELAX: A new circuit simulator for large-scale MOS integrated circuits.— Berkeley: Electronic Research Laboratory, Univ. of California, Memo UCB/ERL M82/6, 1982.
6. Bryant R. E. An algorithm for MOS logic simulation // Lambda Mag. Fourth Quarter.— 1980.— P. 46.
7. Hitchcock R. B., Smith G. L., Chang D. D. Timing analysis of computer hardware // IBM J. Res. Develop.— 1982.— 26.— P. 100.
8. Рюли А. Э., Дитлоу Г. С. Схемотехнический анализ, логическое моделирование и верификация СВИС // ТИИЭР.— 1983.— 71, № 1.
9. Селиванов В. Г., Тючкалов И. В. Моделирование работы электронных схем на логическом уровне // Электронное моделирование.— Киев: Наук. думка, 1986.— № 6.
10. Селиванов В. Г. Способ построения библиотеки элементов для системы моделирования работы электронных схем // Автометрия.— 1986.— № 5.
11. Ковалев А. М., Талныкин Э. А. Графический дисплей растрового типа для систем двухкоординатного проектирования // Там же.— 1984.— № 4.
12. Дементьев С. Г., Селиванов В. Г. Транслятор графического описания изображения принципиальной схемы РЭА в список цепей // Тез. докл. Всесоюз. науч.-техн. конф. «Методы и микроволновые средства цифрового преобразования и обработки сигналов, Рига, 1986».— Рига: ИЭиВТ, 1986.— Т. 3.
13. Сленой Л. И. Средство настройки заданий в ОС — ЕС // Программирование.— 1986.— № 1.
14. Талныкин Э. А. PED — графический редактор в системе проектирования печатных плат // Автометрия.— 1984.— № 5.

Поступило в редакцию 16 января 1987 г.

УДК 681.322.067

И. В. ТЮЧКАЛОВ
(Томск)

ИНТЕРАКТИВНАЯ СИСТЕМА УПРАВЛЕНИЯ ГРАФИЧЕСКОЙ БАЗОЙ ДАННЫХ

В настоящей работе рассматривается развиваемая в рамках комплекса систем автоматизированного проектирования (КАПРИ — САПР) интерактивная графическая система управления базой данных (СУБД ГИ) со встроенным в нее графиче-