На правах рукописи

Зыков Анатолий Геннадьевич

МЕТОДЫ ВЕРИФИКАЦИИ АППАРАТНО-ПРОГРАММНЫХ КОМПОНЕНТОВ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Специальность – 05.13.12 “Системы автоматизации проектирования”
(приборостроение)

АВТОРЕФЕРАТ
диссертации на соискание учёной степени
кандидата технических наук

Санкт-Петербург
2008

Работа выполнена в Санкт-Петербургском государственном университете информационных технологий, механики и оптики

Научный руководитель
профессор, доктор технических наук О.Ф. Немолочнов

Официальные оппоненты:
профессор, доктор технических наук В.И. Анисимов
доцент, кандидат технических наук В.Б. Тарасов

Ведущее предприятие ФГУП ОКБ «Электроавтоматика» Санкт-Петербург

Защита диссертации состоится 23 декабря 2008г. в 15-50 часов в ауд. 466 на заседании диссертационного совета Д 212.227.05 при Санкт-Петербургском государственном университете информационных технологий, механики и оптики по адресу: 197101, г. Санкт-Петербург, Кронверкский пр., д. 49.

С диссертацией можно ознакомиться в библиотеке СПб ГУ ИТМО

Автореферат разослан 21 ноября 2008 г.

Учёный секретарь
совета по защите докторских и кандидатских диссертаций
Д 212.227.05 В.И.Поляков

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность проблемы. Проблема анализа качества аппаратного и программного обеспечения становится сегодня все более острой, особенно по мере расширения использования нанотехнологий в приборостроении и информационных технологий при разработке программного обеспечения. Экспоненциальный рост сложности аппаратного и программного обеспечения вычислительных процессов порождает повышенные требования к бездефектному проектированию. Известны примеры, как дорого обходятся ошибки, допущенные на различных этапах проектирования, поэтому все современные САПР обязательно снабжаются методологическими, программными и инструментальными средствами анализа разрабатываемого изделия на всех этапах автоматизированного проектирования. Не менее актуальными являются проблемы, связанные с обеспечением проектирования надежных программ. Большой вклад в становление и развитие методов решения данной проблемы внесли отечественные ученые Пархоменко П.П., Липаев В.В., Согомонян Е.С., Майоров С.А., Немолочнов О.Ф., Рябов Г.Г., Селютин В.А., Курейчик В.М. и многие другие.

Однако возможности средств верификации сегодня заметно отстают от возможностей систем проектирования и технологии изготовления, поэтому разработка машинно-ориентированных методов верификации аппаратно-программных компонентов вычислительных процессов является *актуальной.*

Цель работы: разработка методов верификации аппаратно–программных компонентов вычислительных систем; разработка машинно-ориентированных алгоритмов построения комплексных кубических покрытий цифровых схем и графо-аналитических моделей программ; разработка структуры и основных подсистем учебно-исследовательской САПР (УИ САПР) верификации вычислительных процессов. В соответствии с поставленной целью необходимо решить следующие основные задачи:

- разработать универсальную модель последовательностной схемы;
- разработать модель вычислительного процесса;
- разработать методы и алгоритмы построения комплексных кубических покрытий цифровых схем и программ;
- разработать методы верификации аппаратно-программных компонентов вычислительных систем;

Методы исследования. Поставленные в диссертационной работе задачи решаются с использованием положений и методов математической логики, теории множеств, теории переключательных схем, теоретического программирования, теории графов, теории алгоритмов.

Научная новизна. В работе получены следующие существенные результаты:

- разработана универсальная модель функционально-логической схемы, позволяющая описывать состояния и переходы схемы в виде комплексного кубического покрытия;
- разработана концептуальная итерационно-рекурсивная двухконтурная модель вычислительного процесса;
- разработан метод построения комплексного кубического покрытия цифровой схемы;
- разработан метод построения комплексного кубического покрытия графо-аналитической модели (ГАМ) программы;
- разработаны методы верификации моделей аппаратно-программных компонентов вычислительных систем различного уровня.

Практическая ценность. Разработаны методы, алгоритмы и программы, осуществляющие построение комплексных кубических покрытий цифровых схем и ГАМ программ на уровне исполняемого кода. Разработаны алгоритмы и программы локальной и глобальной оптимизации построения комплексных кубических покрытий цифровых схем. Разработаны алгоритмы и программы, осуществляющие верификацию аппаратных и программных компонентов вычислительных систем. Разработана структура и алгоритмы УИ САПР верификации и тестирования аппаратных и программных компонентов вычислительных систем.

Внедрение результатов. Разработанная УИ САПР верификации и тестирования вычислительных процессов используется в СПб ГУ ИТМО на кафедре информатики и прикладной математики для анализа результатов лабораторных работ по курсам «Верификация моделей», «Системное программное обеспечение», «Программирование на языке ассемблер», «Технология программирования» для студентов специальности 220100 «Вычислительные машины, системы, комплексы и сети» и по курсу «Автоматизация логического проектирования ЭВС» для студентов специальности 210202 «Проектирование и технология вычислительных средств».

Результаты работы были использованы при выполнении проекта «Рефрен – Н» в ФГУП СПб ОКБ «Электроавтоматика», а также в ФГУП «Научно-исследовательский институт физической оптики, оптики лазеров и информационных оптических систем» Всероссийского научного центра «Государственный оптический институт им. С.И. Вавилова».

Апробация работы: Основные результаты диссертационной работы докладывались и получили одобрение на научных и учебно-методических конференциях профессорско-преподавательского состава ГИТМО(ТУ) (С.-Петербург 1996 - 2000, 2002, 2003 г.г.) и СПб ГУ ИТМО (С.-Петербург 2004 - 2008 г.г.); на Межвуз. науч. -техн. семинаре с междунар. участием «Автоматизация проектирования, технология элементов и узлов компьютерных систем». - СПб: 1998; на Всероссийской НТК «Интеллектуальные САПР-94», Таганрог, 1994; на Юбилейной НТК ППС, посвященная 100-летию университета 29-31 марта 2000 года.- СПб.: СПб ГИТМО (ТУ), 2000; на 6-й МНПК «Безопасность и защита информации сетевых техноло-

гий. COMMON CRITERIA" СПб, 13-15 июня 2001.- СПб.: СПб ГИТМО (ТУ), 2001; на 9-й научно-технической конференции «Теория и технология программирования и защиты информации, применение вычислительной техники» -СПб: СПбГУ ИТМО 2002г.; на Международных научно-технических конференциях «Интеллектуальные системы» (IEEE AIS'04) и «Интеллектуальные САПР» (CAD-2002, 2004 - 2008) Дивноморское; на 11-ой международной научно-практической конференции «Теория и технология программирования и защиты информации»/ СПб: СПбГУ ИТМО, 18 мая 2007; на Первом СПб конгрессе «Профессиональное образование, наука, инновации в XXI веке» / СПб: СПбГУ ИТМО, 26-27 октября 2007.

Публикации. По материалам диссертации опубликовано 36 работ, в том числе - 12 из списка, рекомендованного ВАК.

Структура и объем работы Диссертация состоит из введения, четырех глав, заключения, библиографического списка из 92 наименований, содержит 105 страниц текста, 44 рисунка и 5 таблиц.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы, определены цели и методы исследования.

В первой главе приведен обзор существующих методов верификации аппаратного и программного обеспечения вычислительных систем. Приведены основные термины и определения. Показано место верификации при проектировании соответствующих компонентов вычислительных систем.

Система может представлять собой аппаратные или программные средства, а также некоторую их комбинацию. Аппаратура, особенно цифровая, проектируется с помощью языков описания аппаратуры, например таких, как VHDL и Verilog, проект тем самым представляет программу, поэтому в верификации аппаратных и программных систем много общего. Спецификация может представлять собой требования к функциональным возможностям, временным параметрам, потребляемой мощности, возможности использовать программные средства на различных вычислительных машинах, производительности, габаритам и т.д. Соответственно, верификации подвергаются различные аспекты проектируемой системы.

При анализе аппаратных компонентов выделяют такие методы, как моделирование, верификацию и тестирование. Моделирование и формальная верификация являются основными методами функциональной верификации.

Отмечено, что в таких САПР, как Cadence, Mentor Graphic, Synopsis, Polis и др. основной анализ проводится с помощью подсистем моделирования с использованием FSM – моделей.

При анализе ПО выделяют два основных метода контроля и анализа программ: статический и динамический. Статические методы основаны на исследовании тех или иных свойств ПО без его выполнения с использова-

нием различных инструментальных сред. Наиболее широкое распространение имеют следующие методы статического контроля:

- синтаксический контроль;
- контроль структурированности ПО;
- контроль правдоподобия программ;
- верификация ПО.

Методы верификации основаны на доказательстве корректности или некорректности ПО, соответствии разработанного ПО его спецификациям. Верификация ПО весьма сложна, но практически гарантирует правильность полученного ПО относительно сформулированных для него спецификаций. К недостаткам методов верификации можно отнести то, что формирование метаописания ПО весьма сложно. Кроме того, верификация практически не применяется на поздних этапах жизненного цикла ПО.

Существует два основных вида тестирования ПО:

- функциональное тестирование;
- структурное тестирование.

Функциональное тестирование рассматривает программу как «черный ящик». Проверяется соответствие программы ее внешней спецификации. Отмечено, что исчерпывающее функциональное тестирование невозможно, а для программных продуктов большого размера применение его нецелесообразно. Структурное тестирование основано на знании внутренней структуры программы и предполагает обход всех ветвей по графу управления программы.

На основе сделанных выводов показано единство синтеза и анализа при проектировании логических схем и программ, реализующих вычислительный процесс. Сформулирована цель исследования и задачи, которые необходимо решить для её достижения.

Во второй главе предложена универсальная модель последовательностной схемы. В общем виде любое цифровое устройство может быть представлено в виде операционного устройства (ОУ) и устройства управления (УУ). Устройство управления в отличии от ОУ имеет, как правило, нерегулярную структуру и задача построения его моделей является сложной проблемой и актуальна при проектировании микропроцессоров, заказных БИС, устройств на базе ПЛИС и т.п.. Методы построения моделей УУ в виде графа структурно-временного автомата достаточно исследованы и в работе не рассматриваются. В качестве модели функционально-логической схемы УУ в самом общем виде предлагается модель, представленная на рис. 1. Здесь вектор X есть независимые входы схемы, из которых особо выделены сигнал синхронизации T и сигнал начальной установки элементов памяти R, который может быть как асинхронным сбросом, так и синхронизированным тактом T, что в общем случае не является принципиальным. Значение сигналов обратной связи Y разбито на два подмножества Y^M и Y^S,

соответственно $Y' = Y'^{M} \cup Y'^{S}$. Значения Y' определены как булевы функции $Y' = Y'(X,Y)$. Значения выходных сигналов также определены как $Z = Z(X,Y)$ для автомата Мили и как $Z = Z(Y)$ для автомата Мура. Для схемы также определены некоторые промежуточные сигналы Z_k, как выходы комбинационных подсхем (КС), которые являются вспомогательными и введены как следствие разбиения КС на одновыходные комбинационные подсхемы. Значение вектора сигналов Y определяет предыдущее значение состояния схемы, а значение вектора Y' - последующее состояние. В общем случае векторы Y и Y' задают некоторые классы состояний. Любой сигнал y и y' может принимать три значения: 0,1 и ×, где × - есть безразличное состояние, которое произвольным образом можно доопределить в значение 1 или 0, если это не вызывает противоречия при переходе схемы из состояния Y в состояние Y'.

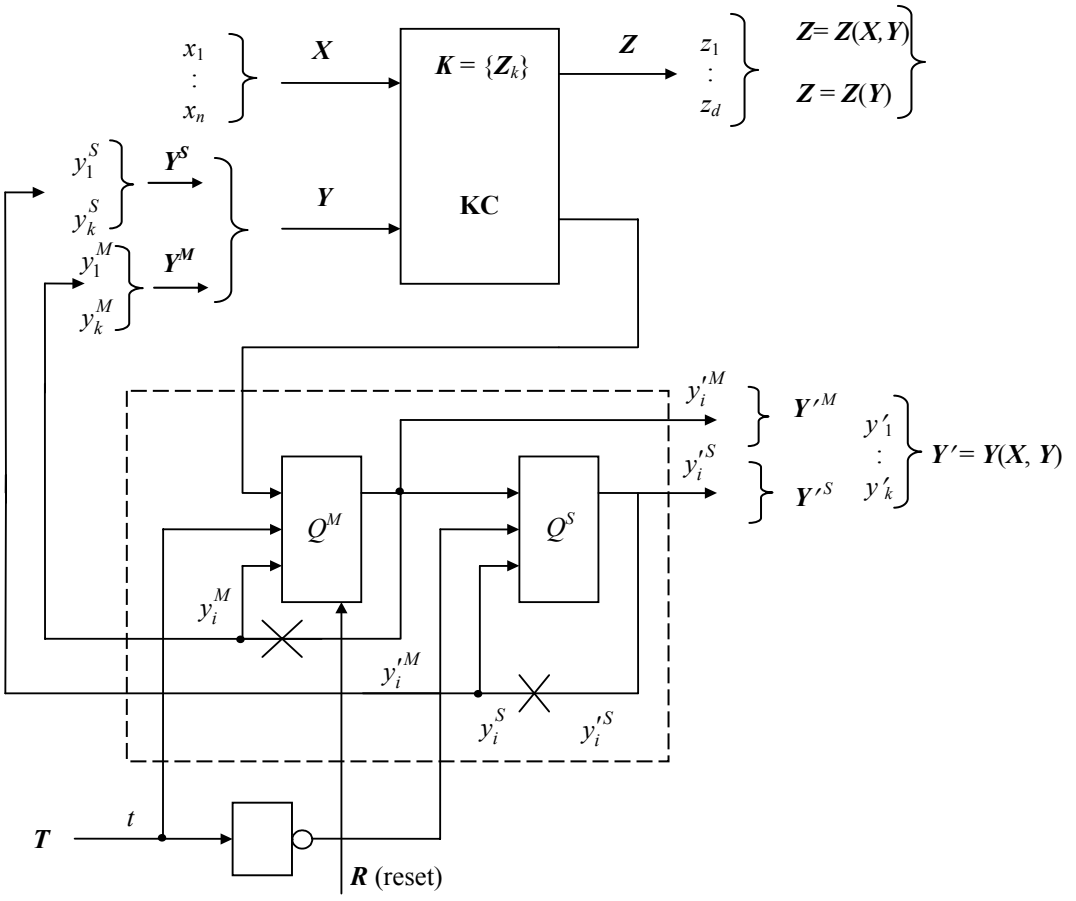


Рис. 1. Универсальная модель функционально-логической схемы

Соответствие между значениями сигналов Y и Y' определяется по интервальным покрытиям $C(y') = C^{x}(y') \cup C^{p}(y')$, где $C^{x}(y')$ есть область установки и $C^{p}(y')$ есть область хранения. Если некоторое значение y' вычислено кубом $c \in C^{x}(y')$, то $y=x$ и $y'=p$, если же значение y' вычислено кубом $C \in C^{p}(y')$, то значение $y = y' = p$.

В соответствии с предложенной структурной моделью схемы (рис. 1) введено понятие куба единого формата $\Phi_0 = X\, Y^{M}\, Y^{S}\, Z_K\ \ Y'^{M}\, Y'^{S}\, Z$. Данный формат в векторном виде описывает все значения сигналов и их разбиение

на функциональные классы. Если не делать различия между классами Y^M, Y^S, Y'^M и Y'^S, то можно ограничиться форматом $\Phi_1 = X\,Y\,Z_K\,Y'\,Z$. Далее, если Z_K отсутствуют, можно ограничиться форматом $\Phi_2 = X\,Y\,\,Y'\,Z$ и, наконец, для случая, когда рассматриваются только переходы схемы, можно перейти к формату $\Phi_3 = X\,Y\,\,Y'$, что является вполне допустимым при теоретическом рассмотрении состояний и переходов схемы, так как значения Z_K и Z являются импликациями от независимых входов X и сигналов обратной связи Y или Y' и их можно вычислить после построения переходов.

При построение графа переходов схемы возможен поиск переходов схемы в прямом или обратном направлении. Любой переход схемы может быть представлен в виде вектора: $X\,Y\,Y'$. Здесь значение вектора Y является исходным состоянием схемы, а Y' - последующим состоянием, в которое схема переходит под действием входных сигналов X в исходном состоянии схемы. Так как УУ является сильно связанным устройством, можно ограничиться построением графа переходов в прямом направлении. В этом случае необходимо отдельно рассмотреть вопрос о поиске начальной вершины графа переходов схемы, а именно, переход из безразличного состояния сигналов обратной связи $Y = (xx...x)$ в некоторое множество исходных состояний Y'. Поиск таких переходов можно осуществить путем пересечения покрытий $\{ C(\,y'\,) \}$ между собой при условии наложения ограничений на значение сигналов обратной связи Y. При поиске начальных вершин:

$$Y_0' = \bigcap_i C_i(y') \text{ , при } Y = xx...x,$$

и при поиске последующих вершин графа переходов:

$$Y'_{j+1} = \bigcap_i C_i(y'), \text{ при } Y = Y_j'.$$

Теперь по изоморфизму построенного графа переходов функционально-логической схемы УУ графу структурно-временного автомата можно сделать заключение, что рассматриваемые модели верифицированы.

В работе рассматривается алгебро-топологическая модель последовательностной схемы в виде комплексного кубического покрытия. Для САПР одним из наиболее компактных и информационных является описание схем в виде кубических покрытий. Предложенная выше универсальная модель последовательностной схемы является основой для построения комплексного покрытия схемы. При этом схема декомпозируется на одновыходные подсхемы, каждая из которых описывается вырожденными покрытиями $C(f_i)$ в локальном формате Фл = $\{X, Y, Y', Z\}$, где X – независимые входы, Y- сигналы обратной связи, определяющие исходное состояние схемы, Y'- сигналы обратной связи, определяющие последующее состояние схемы и Z – выходные или внутренние значения комбинационных элементов (подсхем) схемы. Так как значения Z являются импликациями

от независимых входов X и сигналов обратной связи Y и Y', то можно перейти к локальному формату $Фл_1 = \{ X, Y, Y' \}$.

Построение комплексного покрытия всей схемы позволяет получить полную модель схемы для решения требуемых задач анализа. Комплексное кубическое покрытие (*КР*) строится в глобальном формате Фкс = { *X, Y, Y', Z* }. Любой куб комплексного покрытия $к_i \in KP$ трактуется как некоторое состояние логической схемы. *КР* строится методом пересечения множеств кубов покрытий подсхем на основе полного перебора, то есть всех возможных сочетаний. Таким образом:

$$\boldsymbol{КР} = \bigcap_{i=1}^{n} C(f_i) - \{a \mid a \subseteq b; a, b \in KC\},$$

где n количество покрытий подсхем.

Простой куб $кс_i$ есть такой куб, в котором ни одну координату со значением $p = 1$ или 0, нельзя заменить на × без нарушения импликации $y'_i = p_i$ и $z_j = p_j$.

В работе предложена одноконтурная модель вычислительного процесса (ВП), в развитие которой предлагается итерационно-рекурсивная модель вычислительного процесса с двумя контурами обратной связи. В основе любой программы, вне зависимости от того, для какой вычислительной платформы и на каком алгоритмическом языке она спроектирована, и цифровой схемы, вне зависимости от используемой элементной базы, лежит вычислительный процесс. Рассмотрим ВП, порождаемый вычислительной машиной при интерпретации команд выполняемой программы. Если в качестве описания программы взять ее машинный код, полученный интерпретирующим или компилирующим способом, то, опираясь на знание системы команд процессора и структуры данных вычислительной машины, в которую входит процессор, вычислительный процесс может быть расшифрован без ее исполнения. Поэтому, можно перейти от описания программ со всеми несущественными подробностями, порождаемыми конкретными трансляторами, операционными системами и процессорами, к описанию вычислительного процесса, которое будет инвариантным и независимым от вычислительной среды.

Построение систем логических уравнений для булевых функций в виде кубических покрытий для ациклических и циклических структур вычислительного процесса позволяет получить строгое математическое описание вычислительного процесса. Решение построенных логических уравнений при заданных ограничениях на типы и диапазон изменения переменных с применением аппарата исчисления кубических комплексов позволяет эффективно решать задачи верификации, тестирования, поиска недекларированных возможностей. Построение кубических покрытий булевых функций для циклических вычислительных процессов программ на основе концептуальной итерационно-рекурсивной модели с двумя выделенными контурами обратной связи и являются одной из задач диссертационной работы.

Концептуальная двухконтурная модель вычислительных процессов программ приведена на рис. 2.

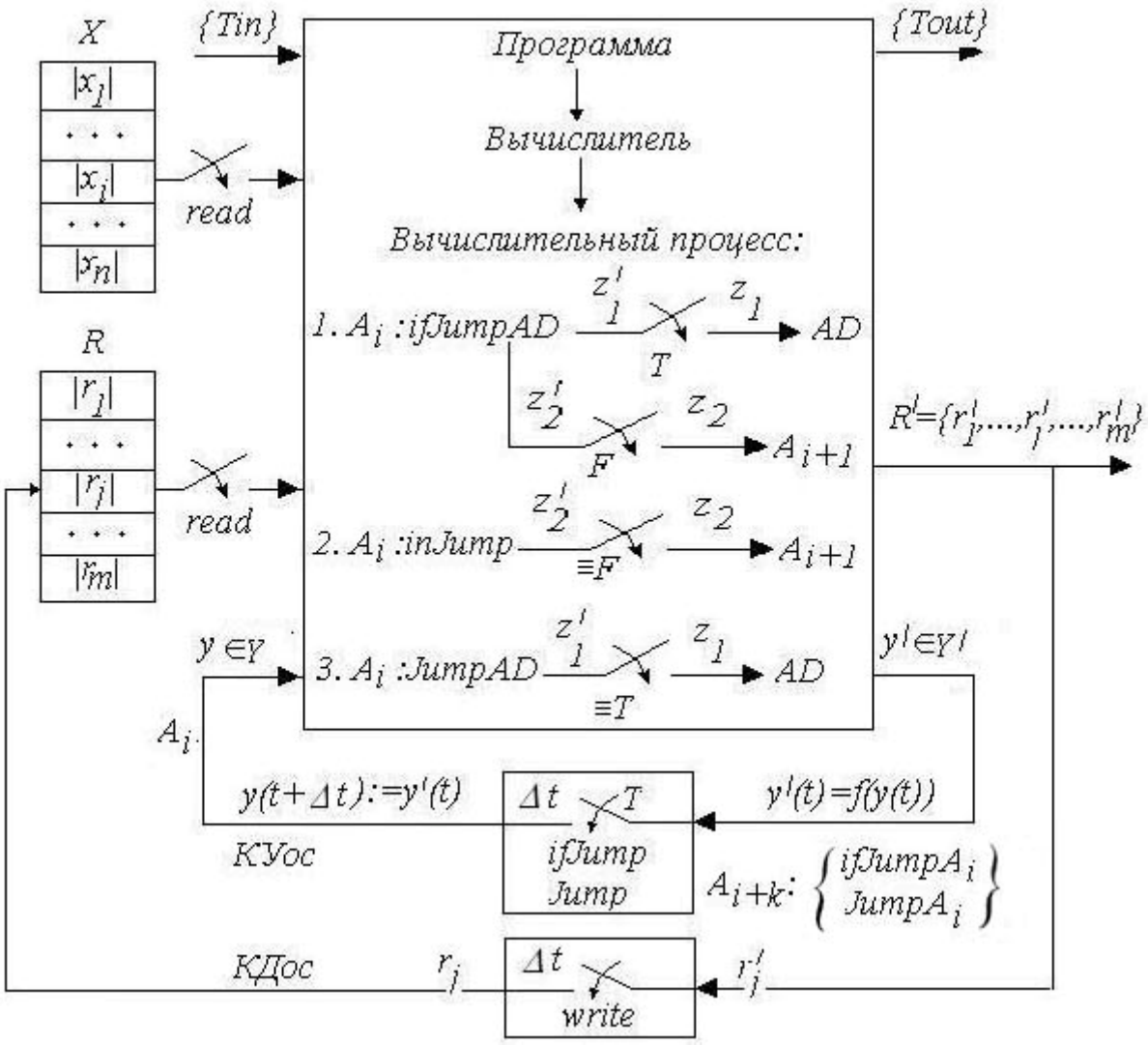


Рис.2. Двухконтурная итерационно-рекурсивная модель вычислительного процесса

Согласно данной модели на вычислительный процесс накладываются следующие ограничения:

- вычислительный процесс начинается в одной из точек входа (*Tin*) и за любое сколь угодно большое, но конечное время, заканчивается в одной из точек выхода (*Tout*), т.е. не рассматриваются расходящиеся процессы;
- данные на входе вычислительного процесса могут изменяться только после его окончания путем перехода от *Tin* в *Tout*, т.е. рассматриваются только детерминированные (как результат дискретизации по времени) процессы.

Контуры обратной связи по управлению описывают циклы в программах и, соответственно, циклические вычислительные процессы, поро-

ждаемые ими, в любой их композиции и с любым уровнем вложенности друг в друга.

Программа представляется как устройство управления в виде сдвигового регистра с бегущей единицей. Наличие значения $z(y)=1$ означает исполнение некоторой текущей команды или вершины, являющейся замкнутым подмножеством команд на графо – аналитической модели (ГАМ).

Под циклом понимается некоторое замкнутое множество команд программы, которое может исполнятся два и более раз и имеет команду возврата по *Jump* или *if Jump* к ранее пройденному адресу. Наличие обращений к процедурам не нарушает данное положение, т.к. любая процедура может быть втянута в тело цикла. На ГАМ циклам будет соответствовать любое множество вершин, охваченных одним и только одним контуром обратной связи (КУо.с.). Контуры обратной связи соответствуют адресам возврата на продолжение цикла в программах и могут быть построены в процессе структурирования программного кода. Для кодирования и обозначения КУо.с. выделены специальные переменные $y \in Y$ из множества управляющих вычислительным процессом $Z \supset Y$.

Для линеаризации логических уравнений, описывающих циклический вычислительный процесс, в КУо.с. вносится фиктивная точка разрыва, а значение переменной управления контуром обратной связи до точки разрыва обозначается как y' и после точки разрыва - как y.

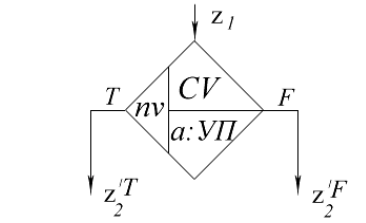
Значения логических переменных y и y', описывающих КУо.с., должны удовлетворять следующим соотношениям:

$$y'(t) = f[y(t)] \quad \text{и} \quad y(t+\Delta t) := y'(t),$$

т.е. значение $y'(t)$ есть некоторая булева функция f от предыдущего значения $y(t)$, а присвоение нового значения y' переменной y происходит в момент времени $t+\Delta t$, где Δt есть переменная задержка, равная времени исполнения тела цикла, что полностью соответствует разрыву сигналов обратной связи в логических схемах, например, при построении покрытий триггеров.

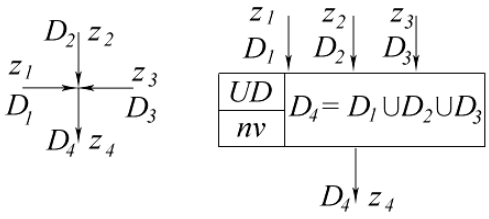
На основании разработанной модели ВП предложены примитивы и их вырожденные покрытия вершин ГАМ (рис.3).

Здесь через z обозначены переменные, которые инициализируют заданную вершину, а через z', z'^{T} и z'^{F} – переменные передачи управления к следующим вершинам. Итеративные формулы обозначены как *IFR,* а рекуррентные – как *RFR*. В покрытиях выделены интервалы хранения ($z=0$) и вычисления ($z=1$). Вершина *UD* для простоты показана для объединения трех дуг (D_1, D_2, D_3 – входные дуги и D_4 – выходная дуга). Заметим, что формулы *IFR* и *RFR* носят произвольный характер. В покрытии *UD* отображен тот факт, что инициализация z_4' может быть только по одной из дуг.



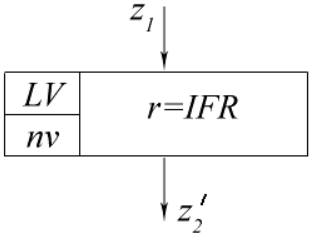
z_1	a	$z_2'^T$	$z_2'^F$
1	1	1	0
1	0	0	1
0	×	0	0

а)

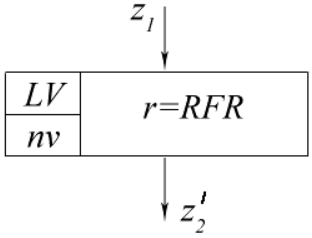


z_1	z_2	z_3	z_4
1	0	0	1
0	1	0	1
0	0	1	1
0	0	0	0

б)



z_1	r	r'	z_2'
1	×	/IFR/	1
0	p_r	p_r	0



z_1	r	r'	z_2'
1	p_r	/RFR/	1
0	p_r	p_r	0

в)

Рис.3. Примитивы и покрытия типовых вершин ГАМ: а) условной вершины (*CV*); б) объединения дуг (*UD*); в) линейной вершины (*LV*); УП – условие-предикат; *nv* – номер вершины; p_r – предыдущее значение переменной *r*; × – безразличное значение переменной.

В третьей главе рассматриваются методы верификации вычислительных процессов в логических схемах и программах. В качестве примера предложенного метода верификации моделей разного уровня рассмотрена графовая модель абстрактного и структурного автоматов схемы пересчета, его схемная реализация и построение комплексного покрытия синтезированной схемы методом пересечения всех кубов вырожденных покрытий подсхем и методом пересечения с ограничениями. На основе комплексного кубического покрытия схемы был восстановлен граф переходов схемы и сделано заключение об изоморфизме графов и, следовательно, об их идентичности, что позволило сделать заключение о верификации моделей разного уровня.

Для моделей одного уровня абстракции предложен метод верификации с использованием комплексных покрытий на основании операций # и ∩ кубов комплексного покрытия анализируемых схем. Рассмотрим применение условий необходимости и достаточности при верификации схемных решений методом моделирования. Пусть задана некоторая булева функция f

своими покрытиями $C^1(f)$, при $f = 1$, и $C^0(f)$ при $f = 0$, которые построены произвольным методом, например по картам Карно. Требуется верифицировать некоторое схемное решение для f в виде логической схемы N (с внешним выходом ZN), спроектированной, например, эвристическим способом в произвольном базисе с применением методов факторизации, декомпозиции или их сочетания.

Необходимым условием верификации является совпадение реакций схемы N ZN =1 (ZN = 0) для $\forall\ c \in C^1(f)$ ($\forall\ c \in C^0(f)$).

Достаточным условием верификации является совпадение реакции схемы ZN =0 (ZN = 1) для $\forall\ c \in C^0(f)$ ($\forall\ c \in C^1(f)$).

Таким образом необходимым и достаточным условиями верификации схемы N является совпадение реакции схемы ZN со значениями булевой функции f для всех кубов $c \in C^1(f) \cup C^0(f)$. Комплексное покрытие $KP = C^1(f) \cup C^0(f)$ и есть метамодель схемы N,

Верификацию схемы N можно осуществить и другим способом, отличным от метода моделирования.

Построим метамодель схемы для значений выхода ZN, равных 0 и 1, в виде комплексного покрытия $KP = C^0(ZN) \cup C^1(ZN)$. В этом случае требуется установить соответствие между покрытиями $C^1(f) \cup C^0(f)$ и $C^0(ZN) \cup C^1(ZN)$, которое может быть выполнено либо с использованием операции вычитания кубов покрытий (#), либо с помощью операции пересечения кубов покрытий ($\cap$). При использовании операции вычитания для верификации схемы N необходимо, чтобы $C^1(f) \# C^1(ZN) = \varnothing$, и достаточно, чтобы $C^1(ZN) \# C^1(f) = \varnothing$. Аналогично для покрытий $C^0(f)$ и $C^0(ZN)$. Из этого следует, что при использовании операции вычитания достаточно иметь только один тип покрытий: либо единичное $C^1(f)$ и $C^1(ZN)$, либо нулевое $C^0(f)$ и $C^0(ZN)$.

Применение алгебро-топологических операций вычитания и пересечения покрытий позволяет при сравнении множеств избежать точного соответствия элементов множеств между собой, поэтому $C^1(f) \# C^1(ZN)$ и $C^1(ZN) \# C^1(f)$ не соответствуют аналогам (не топологическим) обычного вычитания множеств (A \ B и B \ A).

Итак, применив операции вычитания (#) и пересечения ($\cap$) покрытий при верификации объектов получим четыре возможных отношения:

1. $C^1(f) \# C^1(ZN) = \varnothing$ и $C^1(ZN) \# C^1(f) = \varnothing$, или $C^1(f) \cap C^0(ZN) = \varnothing$ и $C^0(f) \cap C^1(ZN) = \varnothing$ - условия полной верификации.
2. $C^1(f) \# C^1(ZN) \neq \varnothing$ и $C^1(ZN) \# C^1(f) = \varnothing$, или $C^1(f) \cap C^0(ZN) \neq \varnothing$ и $C^0(f) \cap C^1(ZN) = \varnothing$ - есть необходимое, но недостаточное условие верификации.
3. $C^1(f) \# C^1(ZN) = \varnothing$ и $C^1(ZN) \# C^1(f) \neq \varnothing$, или $C^1(f) \cap C^0(ZN) = \varnothing$ и $C^0(f) \cap C^1(ZN) \neq \varnothing$ - есть достаточное, но не необходимое условие верификации.

4. $C^1(f)\ \#\ C^1(ZN) \neq \varnothing$ и $C^1(ZN)\ \#\ C^1(f) \neq \varnothing$, или $C^1(f) \cap C^0(ZN) \neq \varnothing$ и $C^0(f) \cap C^1(ZN) \neq \varnothing$ - условия верификации отсутствуют.

Во 2-ом и 3-ем случаях можно говорить о верификации частично определенных функций (объектов) с использованием «размытых» множеств, иначе говоря, один объект «делает» больше, чем другой. Таким образом, из вышеизложенного следует, что полная верификация существует только при выполнении соотношений 1-го случая. Данное рассуждение проведено для случая одновыходной схемы N и исходного покрытия булевой функции f, которое является простейшим случаем метамодели высшего ранга. Аналогичные методы можно применить при верификации граф-схем алгоритмов, конечных автоматов (абстрактных и структурных), многовыходных последовательностных схем и программ.

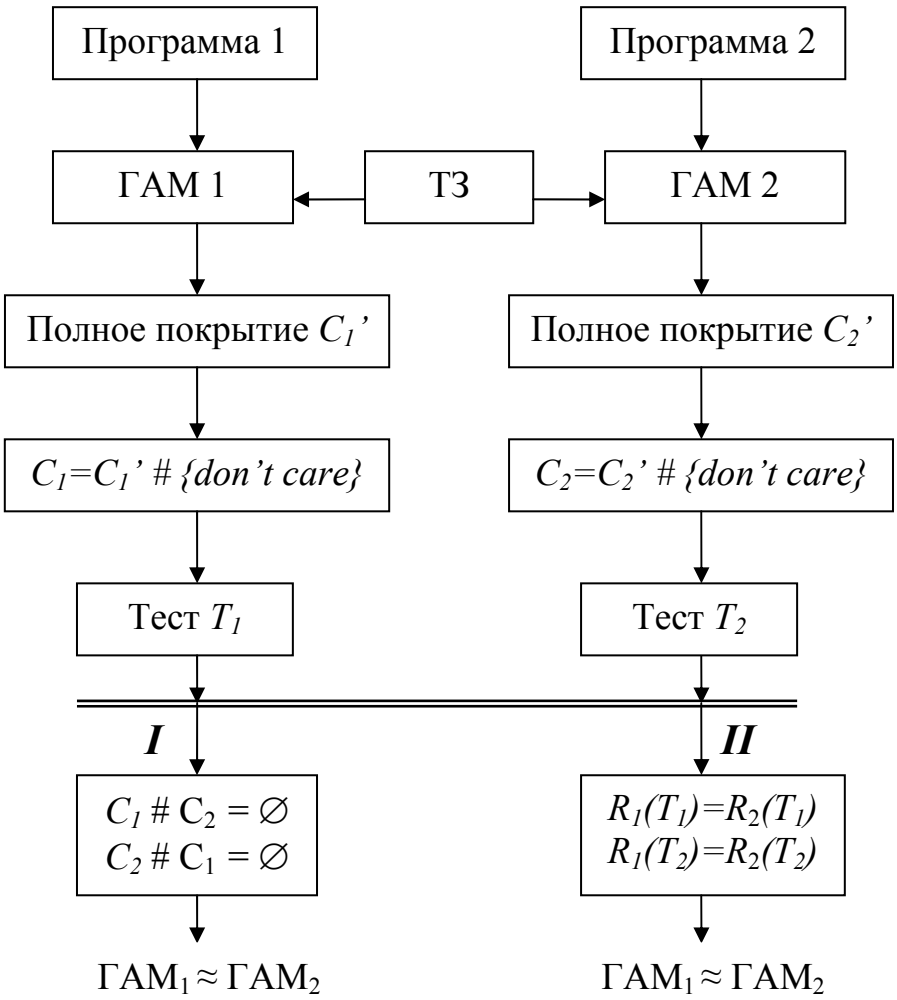


Рис.4. Общая схема процесса верификации вычислительных процессов

На рис. 4 один и тот же вычислительный процесс реализован двумя различными способами. На схеме показано, что на основании технических заданий (ТЗ) или разработанных программ, строятся ГАМ и комплексные покрытия двух рассматриваемых вычислительных процессов. Из них методом алгебро-топологического вычитания исключаются значения, на ко-

торых функция не определена (*don't care*), и строятся контролирующие тесты. Предложены два способа верификации этих процессов:

– метод алгебро-топологического вычитания покрытий каждого из каждого. При условии пустого значения результата делается заключение о эквивалентности данных вычислительных процессов;

– метод построения тестовых наборов по комплексным покрытиям путем пересечения кубов из интервальных частей покрытий и перекрестное тестирование, по результатам которого делается заключение о результатах верификации.

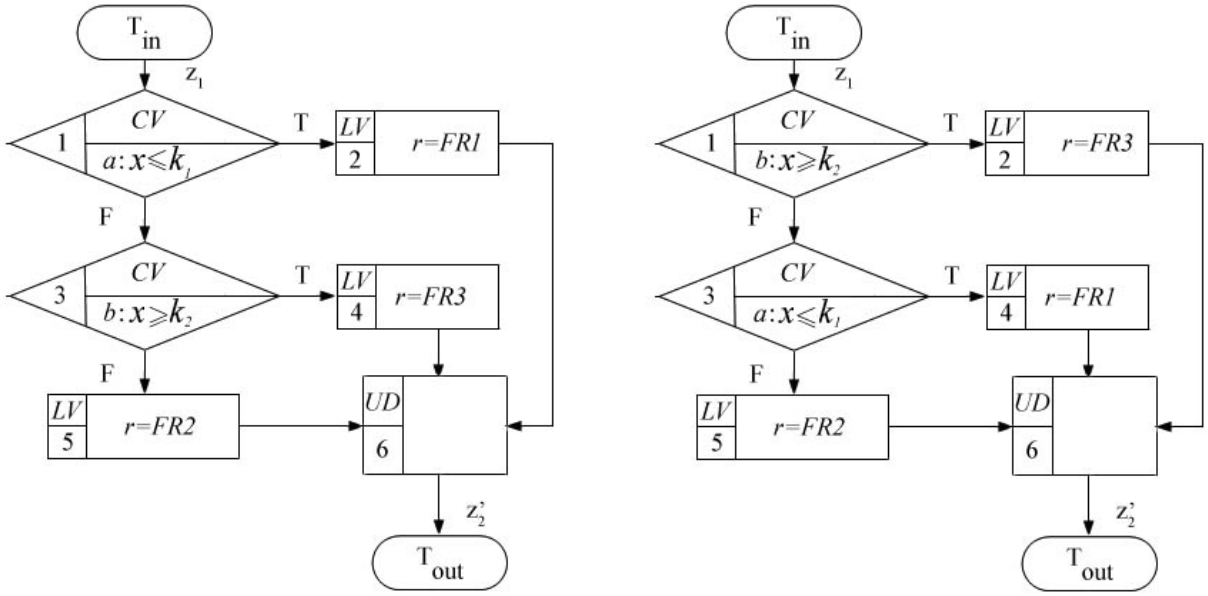
Рассмотрим пример верификации ациклического процесса. Пусть задана некоторая интервальная формула:

$$r = \begin{cases} FR1, & при \quad x \le k_1; \\ FR2, & при \quad k_1 < x < k_2; \\ FR3, & при \quad x \ge k_2, \end{cases}$$

реализующая вычисления некоторой переменной *r* по различным формулам: *FR1, FR2* и *FR3* произвольного вида в зависимости от двух булевых переменных, задающих некоторые условия-предикаты в виде неравенств: *a:* $x \le k_1$ и *b:* $x \ge k_2$.

Переход от неравенств к булевым переменным при проектировании вычислительного процесса позволяет абстрагироваться от конкретного смысла неравенств и им соответствующих условий-предикатов и рассмотреть решение задачи верификации в общем виде.

ГАМ вычисления переменной *r* приведены на рис.5. На рис.5.а) показана функциональная декомпозиция булевой функции *f = f(a,b)* с начальной вершиной условия-предиката *a* (ГАМ 1), а на рис.5.б) – с начальной вершиной *b* (ГАМ 2).



ab	*00*	*01*	*11*	*10*
	FR2	*FR3*	***FR1***	*FR1*

а)

ab	*00*	*01*	*11*	*10*
	FR2	*FR3*	***FR3***	*FR1*

б)

Рис.5. ГАМ вычисления интервальной формулы

Построим комплексные кубические покрытия $C_1(r)$ и $C_2(r)$ для вычисляемой переменной r по ГАМ 1 и ГАМ 2:

$C_1(r)=$

z_1	a	b	r	r'	z_2'	$\{c\}$
1	1	×	×	/*FR1*/	1	c_1
1	0	1	×	/*FR3*/	1	c_2
1	0	0	×	/*FR2*/	1	c_3
0	×	×	p	p	0	c_4

$C_2(r)=$

z_1	a	b	r	r'	z_2'	$\{c\}$
1	×	1	×	/*FR3*/	1	c_1
1	0	0	×	/*FR2*/	1	c_2
1	1	0	×	/*FR1*/	1	c_3
0	×	×	p	p	0	c_4

По покрытиям $C_1(r)$ и $C_2(r)$ построим тесты $T_1(r)$ и $T_2(r)$ путем пересечения кубов из интервальных частей покрытий $C_1(r)$ и $C_2(r)$, соответственно. Получим тесты:

$T_1(r)=$

z_1	a	b	r	r'	z_2'	$\{t\}$
1	1′	0	×	/*FR1*/	1	t_1
1	0′	0	×	/*FR2*/	1	$\tilde{t}_1$
1	0	1′	×	/*FR3*/	1	t_2
1	0	0′	×	/*FR2*/	1	$\tilde{t}_2$
1′	0	1	p	/*FR3*/	1	t_3
0′	0	1	p	p	0	$\tilde{t}_3$

Примеч. $t_1/\tilde{t}_1 \in c_1 \cap c_3$ *из* $C_1(r)$

$T_2(r)=$

z_1	a	b	r	r'	z_2'	$\{t\}$
1	0	1′	×	/*FR3*/	1	t_1
1	0	0′	×	/*FR2*/	1	$\tilde{t}_1$
1	1′	0	×	/*FR1*/	1	t_2
1	0′	0	×	/*FR2*/	1	$\tilde{t}_2$
1′	1	0	p	/*FR1*/	1	t_3
0′	0	1	p	p	0	$\tilde{t}_3$

Примеч. $t_1/\tilde{t}_1 \in c_1 \cap c_3$ *из* $C_2(r)$

Заметим, что для формул должны выполняться условия: /*FR1*/ ≠ /*FR2*/ ≠ /*FR3*/ ≠ p, т.е. вычисляемые и хранимые значения должны различаться на разных наборах теста. Штрихами в тестах отмечены значения активно изменяемых условий-предикатов.

С учетом удаления *ab=11* верификация по покрытиям дает $C_1 \# C_2 = \varnothing$ и $C_2 \# C_1 = \varnothing$, что и свидетельствует об эквивалентности вычислительных процессов. Это наглядно можно наблюдать на картах Карно приведенных на рис.5.

Перекрестное тестирование дает следующий результат: $R_1(T_1)=R_2(T_1)$ и $R_1(T_2)=R_2(T_2)$, что также подтверждает эквивалентность вычислительных процессов. Заметим, что если переменную *r* вычислять по разным упрощенным формулам *FR1, FR2* и *FR3*, то метод перекрестного тестирования является предпочтительным, так как не требует приведения выражений формул к каноническому виду.

В четвертой главе диссертационной работы приводятся машинно-ориентированные алгоритмы реализации основных операций по вычислению комплексных кубических покрытий. Алгоритм построения комплексного кубического покрытия представлен на рис. 6.

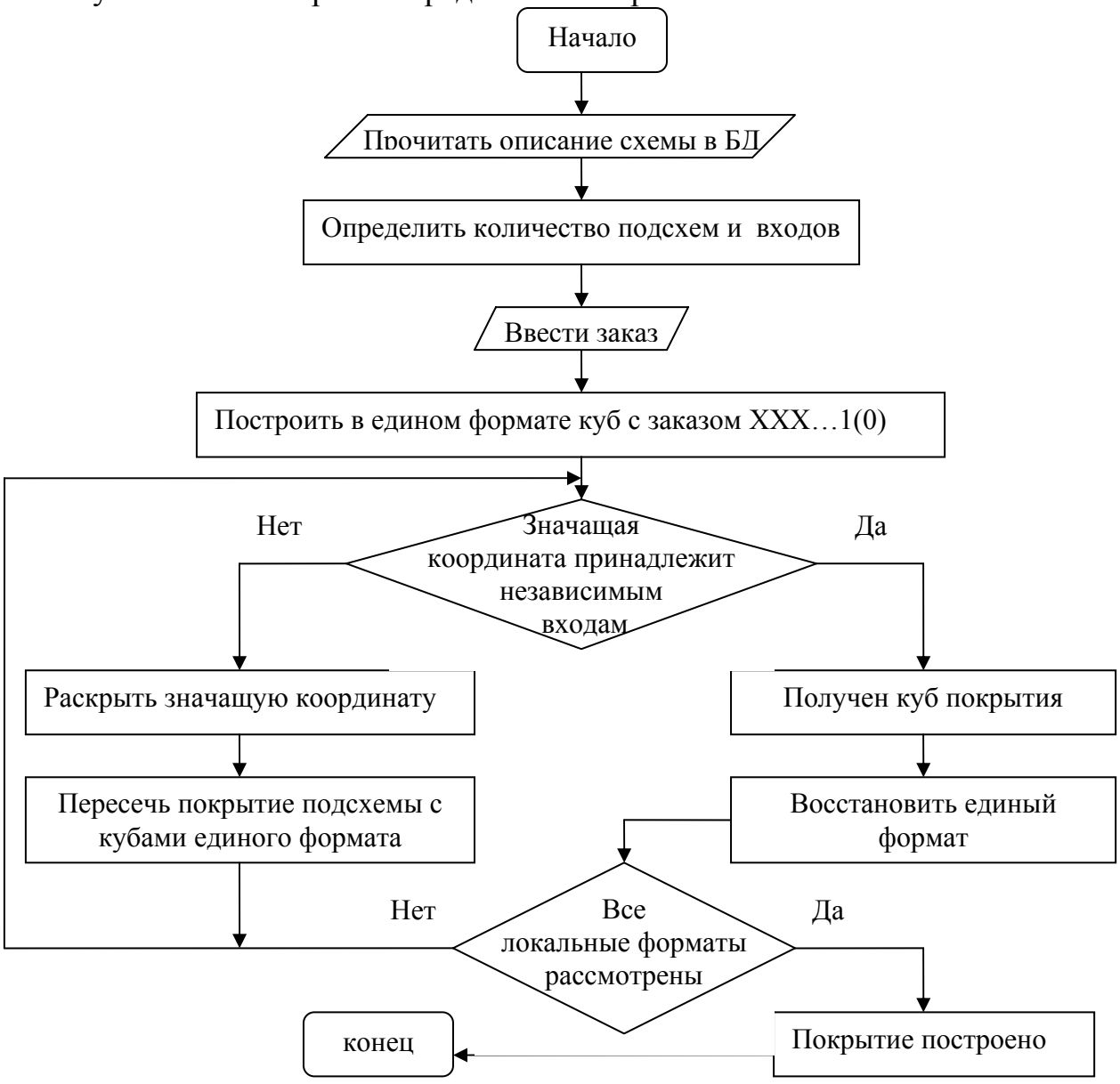


Рис. 6. Алгоритм построения комплексного кубического покрытия

На рис. 7. приводится структура учебно-исследовательской САПР верификации вычислительных процессов (УИ САПР ВВП)

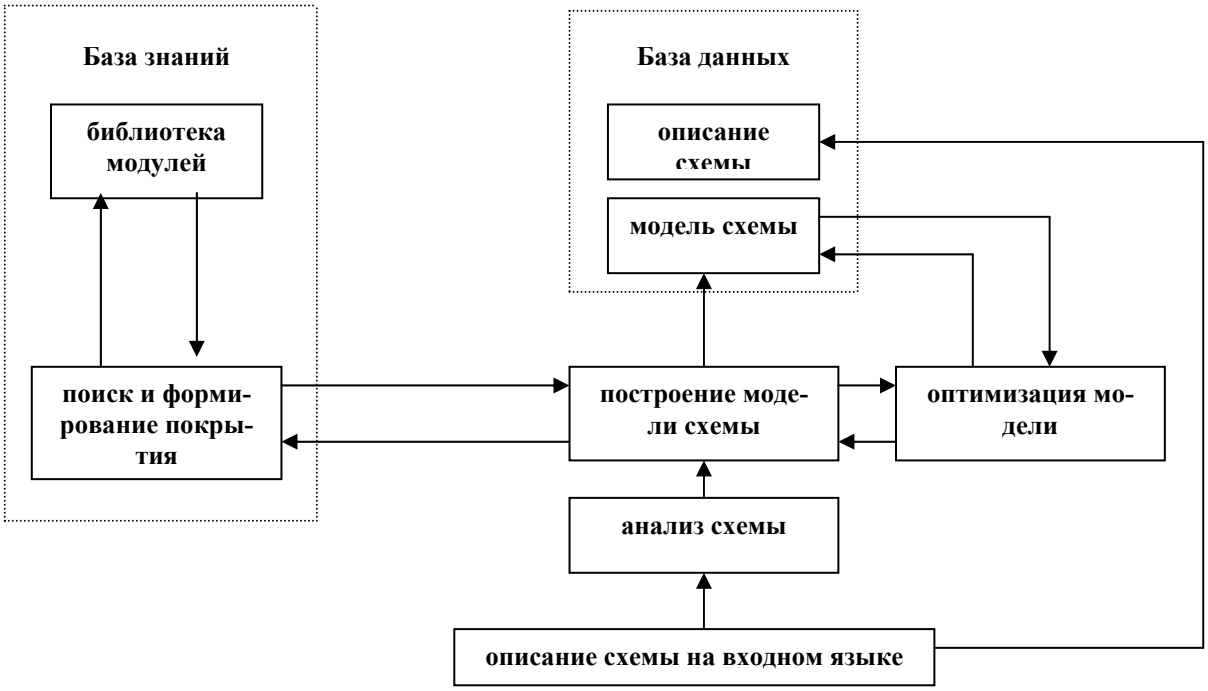


Рис.7. Структурная схема экспериментальной САПР

Система представляет собой консольное приложение для ОС Windows NT/2000 /XP и является программной реализацией разработанных в диссертационной работе алгоритмов. Дано описание состава модулей, входящих в систему. УИ САПР ВВП состоит из трех основных частей: базы знаний, базы данных и монитора управлениями событиями.

В заключении содержатся основные результаты, полученные в диссертационной работе.

1. Предложена универсальная модель последовательностной схемы. На основе предложенной модели разработана методика построения комплексного кубического покрытия схемы для различных задач анализа в САПР.

2. Разработана концептуальная итерационно-рекурсивная двухконтурная модель вычислительного процесса.

3. Предложена модель вычислительного процесса в виде орграфа и кубических покрытий условий-предикатов для вычисляемых переменных.

4. Для машинно-ориентированного описания ГАМ предложены примитивы и вырожденные покрытия типовых вершин.

5. Предложены методы верификации через эквивалентность покрытий и перекрестное тестирование вычислительного процесса вне зависимости от его физической реализации.

6. Разработана методика структурирования вычислительного процесса путем разбиения множества команд машинного кода на замкнутые подмножества команд с одной точкой входа и одной точкой выхода.

7. Разработана структура экспериментальной учебно-исследовательской САПР верификации вычислительных процессов, алгоритмы и программы решения отдельных задач, в частности: построение комплексного покрытия схемы, построение графа и покрытий частично определённых булевых функций, моделирования логических неисправностей условий-предикатов.

Список работ, опубликованных по теме диссертации

1. Блохин В.Н., Голованевский Г.Л., Зыков А.Г., Немолочнов О.Ф. Доступная система контроля цифровых узлов и верификация логических модулей. / Сб. «ЭВМ в проектировании и производстве» / Вып. 4, «Машиностроение», 1989.

2. Зыков А.Г., Немолочнов О.Ф. Автоматная модель устройства управления в САПР при верификации проекта. / Межвузовский сборник научных трудов «Автоматизированное проектирование в радиоэлектронике и приборостроении», С.-Пб, ГЭТУ, 1994. С. 21-23.

3. Зыков А.Г., Немолочнов О.Ф. Создание моделей сложных процессоров при верификации проектов. / Материалы Всероссийской НТК «Интеллектуальные САПР-94», Таганрог, 1994.

4. Зыков А.Г. Построение моделей устройств управления микропроцессоров в виде абстрактного графа автомата. / Тезисы док. Межвуз. науч. -техн. семинара с междунар. участием «Автоматизация проектирования, технология элементов и узлов компьютерных систем». - СПб: 1998. С. 3-4

5. Зыков А.Г., Немолочнов О.Ф. К вопросу верификации функционально-логической схемы на основе абстрактного графа автомата. / Тезисы док. Межвуз. науч. -техн. семинара с междунар. участием «Автоматизация проектирования, технология элементов и узлов компьютерных систем». - СПб: 1998. С. 5-6.

6. Зыков А.Г., Копорский Н.С., Лаздин А.В. К вопросу о принципах верификации результатов имитационного моделирования комплексированных оптоэлектронных систем. / Тезисы док. Межвуз. НТС с междунар. участием «Автоматизация проектирования, технология элементов и узлов компьютерных систем». - СПб: 1998. С.6.

7. Зыков А.Г., Немолочнов О.Ф. Функционально-логическая модель устройства управления / Тезисы докладов XXX НТК ППС -СПб.: СПбГИТМО (ТУ), 1999. С. 90.

8. Немолочнов О.Ф., Зыков А.Г. Построение графа переходов устройства управления по функционально-логической схеме. / Тезисы докладов XXX НТК ППС СПбГИТМО (ТУ) - СПб.: СПб ГИТМО (ТУ), 1999. с. 95.

9. Немолочнов О.Ф., Поляков В.И., Зыков А.Г. Верификация и тестирование программ по проблеме 2000 года (смены дат). / Тезисы докладов. Часть II. Юбилейная НТК ППС, посвященная 100-летию университета 29-31 марта 2000 года.- СПб.: СПб ГИТМО (ТУ), 2000. с.3.

10. Зыков А.Г., Немолочнов О.Ф. Метод пересечения покрытий как средство анализа функционально-логических схем. / Тезисы докладов. Часть II. Юбилейная НТК ППС, посвященная 100-летию университета 29-31 марта 2000 года.- СПб.: СПб ГИТМО (ТУ), 2000. с.5.

11. Зыков А.Г., Немолочнов О.Ф. Метод пересечения покрытий с ограничением при решении задач верификации проекта. / Тезисы докладов. Часть II. Юбилейная НТК ППС, посвященная 100-летию университета 29-31 марта 2000 года.- СПб.: СПб ГИТМО (ТУ), 2000. с.6.

12. Воробьев А.В., Зыков А.Г., Поляков В.И., Шевченко А.А. Обеспечение безопасности функционирования распределенных информационно-управляющих систем. / Труды 6-й МНПК «Безопасность и защита информации сетевых технологий. COMMON CRITERIA" СПб, 13-15 июня 2001.- СПб.: СПб ГИТМО (ТУ), 2001. с.33-34.

13. Зыков А.Г., Немолочнов О.Ф., Поляков В.И. Формализация ограничений на пересечение множеств кубов при верификации. / Известия ТРТУ. №3 (26) Тематический выпуск «Интеллектуальные САПР». Материалы Международной НТК «Интеллектуальные САПР». - Таганрог: Изд-во ТРТУ, 2002. с.198-199.

14. Зыков А.Г., Немолочнов О.Ф., Поляков В.И. Универсальная модель последовательностных схем в САПР / Научно-технический вестник СПБ ГИТМО (ТУ). Выпуск 6. Информационные, вычислительные и управляющие системы / Гл. ред. В.Н. Васильев.- СПб.: СПб ГИТМО (ТУ), 2002. с.107-108.

15. Зыков А.Г., Немолочнов О.Ф., Поляков В.И. Построение комплексного покрытия последовательностных схем методом пересечения покрытий систем булевых функций / Научно-технический вестник СПБ ГИТМО (ТУ). Выпуск 6. Информационные, вычислительные и управляющие системы / Гл. ред. В.Н. Васильев.- СПб.: СПб ГИТМО (ТУ), 2002. с.109-111.

16. Зыков А.Г., Немолочнов О.Ф., Поляков В.И. Верификация моделей цифровых устройств в САПР. / Труды Международных конференций «Искусственные интеллектуальные системы»(IEEE AIS'02) и «Интеллектуальные САПР» (CAD-2002). Научное издание. – М.: Издательство Физматлит, 2002. –609 с. –ISBN 5-940520031-6/. С.320-323.

17. Зыков А.Г., Немолочнов О.Ф., Виноградов Ю.Н., Поляков В.И. Верификация как средство отладки моделей различного уровня. / «Известия вузов. Приборостроение»/ Том 46, №2, СПб, февраль 2003. с.51-55.

18. Немолочнов О.Ф., Зыков А.Г., Лаздин А.В., Поляков В.И. Верификация в исследовательских, учебных и промышленных системах / Научно-технический вестник СПбГУ ИТМО. Выпуск 11. Актуальные проблемы анализа и синтеза сложных технических систем. / Под ред. В.О. Никифорова- СПб: СПбГУ ИТМО, 2003. С. 146-151.

19. Зыков А.Г., Немолочнов О.Ф., Поляков В.И. Методы верификации цифровых устройств / Труды Международных научно-технических конференций «Интеллектуальные системы» (IEEE AIS'04) и «Интеллектуальные САПР» (CAD-2004). Научное издание и 3-х томах. М.: Изд-во Физматлит, 2004, Т.2. - 468 с. –ISBN 5-9221-0531-5. С. 39-41.

20. Зыков А.Г., Немолочнов О.Ф., Поляков В.И. Методы и алгоритмы анализа цифровых устройств в САПР / Труды Международных научно-технических конференций «Интеллектуальные системы» (IEEE AIS'04) и «Интеллектуальные САПР» (CAD-2004). Научное издание и 3-х томах. М.: Изд-во Физматлит, 2004, Т.2. - 468 с. –ISBN 5-9221-0531-5. С. 41-45.

21. Немолочнов О.Ф., Зыков А.Г., Поляков В.И. Кубические покрытия логических условий вычислительных процессов и программ. / Научно-технический вестник СПбГУ ИТМО. Выпуск 14. Информационные технологии, вычислительные и управляющие системы. / Гл. ред. В.Н. Васильев.- СПб: СПбГУ ИТМО, 2004. с.225-233.

22. Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Сидоров А.В. Структурирование программ и вычислительных процессов на множество линейных и условных вершин / Научно-технический вестник СПбГУ ИТМО. Выпуск 19. Программирование, управление и информационные технологии / Гл. ред. В.Н. Васильев.- СПб: СПбГУ ИТМО, 2005. с.207-212.

23. Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Осовецкий Л.Г., Сидоров А.В., Кулагин В.С. Итерационно-рекурсивная модель вычислительных процессов программ / «Известия вузов. Приборостроение» / Том 48, №12, СПб, декабрь 2005, С.14-20.

24. Немолочнов О.Ф., Зыков А.Г., Поляков В.И. Методы анализа вычислительных процессов программ в исследовательских и учебных проектах / Труды межд.конф. «Интеллектуальные системы» (AIS’05) и «Интеллектуальные САПР» (CAD-2005), Научное издание в 3-х томах-М.: Физматлит, 2005,Т2. 532с. - ISBN 5-9221-0621-X. С.424-430.

25. Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Кулагин В.С., Петров К.В. Логические неисправности вычислительных процессов программ / Труды 9-й научно-технической конференции «Теория и технология программирования и защиты информации, применение вычислительной техники» -СПб: СПбГУ ИТМО, 18 мая 2005. С.2-3.

26. Комплексные кубические покрытия и графо-аналитические модели как средство описания вычислительных процессов программ /

Немолочнов О.Ф., Зыков А.Г., Поляков В.И. /Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS'06) и «Интеллектуальные САПР» (CAD-2006), Научное издание в 3-х томах-М.: Физматлит, 2006, Т2.- 588с. - ISBN 5-9221-0686-4. С.3-7.

27. Моделирование простых логических неисправностей вычислительных процессов программ / Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Петров К.В. / Научно-технический вестник СПбГУ ИТМО. Выпуск 32. Информационные технологии: теория, методы, приложения / Гл. ред. В.Н. Васильев.- СПб: СПбГУ ИТМО, май 2006. С. 113-118.

28. Учебно-исследовательская САПР верификации и тестирования вычислительных процессов программ / Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Петров К.В. / Научно-технический вестник СПбГУ ИТМО. Выпуск 32. Информационные технологии: Теория, методы, приложения / Гл. ред. В.Н. Васильев.- СПб: СПбГУ ИТМО, май 2006. С. 127-128.

29. Методы тестирования вычислительных процессов /Немолочнов О.Ф., Зыков А.Г., Осовецкий Л.Г., Поляков В.И. / Научно-технический вестник СПбГУ ИТМО. Выпуск 45. Информационные технологии: теория, методы, приложения / Гл. ред. В.Н. Васильев.- СПб: СПбГУ ИТМО, май 2007. С.121-125.

30. Методы формализации графо-аналитических моделей вычислительных процессов программ / Немолочнов О.Ф., Зыков А.Г., Поляков В.И. / Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS'07) и «Интеллектуальные САПР» (CAD-2007), Научное издание в 4-х томах-М.: Физматлит, 2007, Т.3.- 496с. - ISBN 978-5-9221-0856-0. С.88-96.

31. Модель и примитивы покрытий вершин циклических вычислительных процессов / Немолочнов О.Ф., Зыков А.Г., Осовецкий Л.Г., Поляков В.И. //Известия вузов. Приборостроение. 2007. Том 50, №8, С.18-23.

32. Структурирование вычислительного процесса по программному коду / Немолочнов О.Ф., Зыков А.Г., Осовецкий Л.Г., Поляков В.И. // - Труды 11-й международной научно-практической конференции «Теория и технология программирования и защиты информации»/ СПб: СПбГУ ИТМО, 18 мая 2007. С.7-9.

33. Автоматизированная учебно-исследовательская система верификации и тестирования программ / Немолочнов О.Ф., Зыков А.Г., Осовецкий Л.Г., Петров К.В. // Труды Первого Санкт-Петербургского конгресса «Профессиональное образование, наука, инновации в XXI веке» / СПб: СПбГУ ИТМО, 26-27 октября 2007. С.209-210.

34. Верификация как средство бездефектного проектирования программных продуктов / Немолочнов О.Ф., Зыков А.Г., Осовецкий Л.Г. // Труды Первого Санкт-Петербургского конгресса «Профессиональное

образование, наука, инновации в XXI веке» / СПб: СПбГУ ИТМО, 26-27 октября 2007. С.221-223.

35. Тестирование логических неисправностей вычислительных процессов в программах / Немолочнов О.Ф., Зыков А.Г., Осовецкий Л.Г., Поляков В.И., Петров К.В. / Журнал «Информационные технологии» / №12, М., 2007, С.2-5.

36. Импликация и неопределенные значения условий-предикатов вычислительных процессов / Немолочнов О.Ф., Зыков А.Г., Поляков В.И. / Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS'08) и «Интеллектуальные САПР» (CAD-2008), Научное издание в 4-х томах. - М.: Физматлит, 2008, Т.1.- 400с. - ISBN 978-5-9221-0922-2. С.140-145.