

A 05.13.02
E-29

ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ ԵՎ
ԳԻՏՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

ՀԱՅԱՍՏԱՆԻ ՊԵՏԱԿԱՆ ԾԱՐՏԱՐԱԳԻՏԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ
(ՊՈԼԻՏԵԽՆԻԿ)

Եղիազարյան Վլադիմիր Սահակի

**ԹՎԱՅԻՆ ՄԽԵՄԱՆԵՐԻ ԱՎՏՈՄԱՏԱՑՎԱԾ ՆԱԽԱԳԾՄԱՆ
ԼԵԶՎԱԿԱՆ ԵՎ ԱԼԳՈՐԻԹՄԱԿԱՆ ՀԻՄՆԱԽՆԴԻՐՆԵՐ**

Ե.13.02-«Ավտոմատացման համակարգեր» մասնագիտությամբ
տեխնիկական գիտությունների դոկտորի զիտական աստիճանի
հայցման ատենախոսության

ՍԵՂՄԱԳԻՐ
Երևան 2012

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
АРМЕНИЯ

ГОСУДАРСТВЕННЫЙ ИНЖЕНЕРНЫЙ УНИВЕРСИТЕТ
АРМЕНИИ (ПОЛИТЕХНИК)

Егиазарян Владимир Саакович

**ЯЗЫКОВЫЕ И АЛГОРИТМИЧЕСКИЕ ПРОБЛЕМЫ
АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ
ЦИФРОВЫХ СХЕМ**

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
доктора технических наук по специальности
05.13.02-“Системы автоматизации”
Ереван 2012

Ատենախոսության թեման հաստատվել է Հայաստանի պետական
ճարտարագիտական համալսարանում (Պոլիտեխնիկ):

Գիտական խորհրդատու՝ ֆ.-մ.գ.դ., պրոֆեսոր Շ.Ե. Բոզոյան

Պաշտոնական ընդդիմախոսներ՝
տ.գ.դ., պրոֆեսոր, ՀՀ ԳԱԱ
ակադեմիկոս Ա.Թ. Քուչուկյան
տ.գ.դ., պրոֆեսոր Վ.Շ. Մելիքյան
տ.գ.դ., պրոֆեսոր Ս.Վ. Գավրիլով

Առաջատար կազմակերպություն՝ Խարկովի ռադիոէլեկտրոնիկայի
ազգային համալսարան

Ատենախոսության պաշտպանությունը տեղի կունենա 2012թ. հունիսի 8-ին, ժամը
14⁰⁰-ին, ՀՊՃՀ-ում գործող «Կառավարման, ավտոմատացման համակարգերի և
էլեկտրոնիկայի» 032 մասնագիտական խորհրդի նիստում (հասցեն՝ 0009, Երևան,
Տերյան փ., 105, 17 մասնաշենք):

Ատենախոսությանը կարելի է ծանոթանալ ՀՊՃՀ-ի գրադարանում:

Սեղմագիրն առաքված է 2012թ. մայիսի 7-ին:

032 մասնագիտական խորհրդի
գիտական քարտուղար, տ.գ.թ., դոց.

Ա.Գ. Ավետիսյան

Тема диссертации утверждена в Государственном инженерном университете Армении
(Политехник).

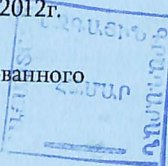
Научный консультант: д.т.н., профессор Ш.Е. Бозоян

Официальные оппоненты: д.т.н., профессор,
академик НАН РА А.Т. Кучукян
д.т.н., профессор В.Ш. Меликян
д.т.н., профессор С.В. Гаврилов

Ведущая организация: Харьковский национальный
университет радиоэлектроники

Защита диссертации состоится 8-ого июня 2012г. в 14⁰⁰ на заседании
специализированного совета 032-“Управления, систем автоматизации и электроники”,
действующего в ГИУА (адрес: 0009, г. Ереван, ул. Теряна 105, корпус 17).
С диссертацией можно ознакомиться в библиотеке ГИУА.
Автореферат разослан 7-ого мая 2012г.

Ученый секретарь специализированного
совета 032, к.т.н., доц.



А.Г. Аветисян

2402-2012

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Задачи проектирования больших интегральных
схем (БИС) в настоящее время стали настолько трудоемкими, в них нужно
учитывать такое большое число технических и технологических
ограничений, что использование вычислительных устройств является
единственным способом получить работающую конструкцию. Тенденции к
росту степени интеграции микросхем, сохраняющихся в микроэлектронике,
привели к появлению сверхбольших интегральных схем (СБИС). С одной
стороны, техника СБИС- это новые возможности создания сложных
систем, с другой – увеличение трудностей проектирования.

Преодоление этих трудностей немыслимо без систем автоматизации
проектирования и, как следствие, без использования языков описания
аппаратур.

Не касаясь истории развития способов описания аппаратур,
рассмотрим те языки, которые находятся на арене в настоящее время.

Это языки Verilog, VHDL и System C, которые в настоящее время
являются международными стандартами. Язык Verilog был разработан
фирмой Automated Integrated Design Systems (с 1985г. Gateway Design
Automatic) как внутренний язык симуляции. Фирма Cadens приобрела
Gateway в 1989г. и открыла Verilog для общественного использования. В
1995г. был определен стандарт языка - Verilog LRM (Language Reference
Manual), IEEE 1364-1995. Язык Verilog изначально был предназначен для
верификации с помощью моделирования, временного анализа и
логического синтеза схем. Наряду с этим, стандарт IEEE 1364-1995
дополнительно определяет интерфейс языка программирования. Этот
интерфейс является набором системных программ, обеспечивающих
двустороннюю связь между языком Verilog и языком программирования
высокого уровня (как правило – язык программирования C).

В 2001г. и 2005г. появились новые дополнения и редакции стандарта
языка Verilog IEEE 1364-2001 и IEEE 1364-2005. В этих стандартах был
сделан шаг в сторону языков программирования, в частности,
поддерживается работа со знаковыми переменными, добавлены встроенные
операции (такие как +, -, /, * и т.д.). Язык Verilog имеет возможность
описания как цифровых, так и аналоговых схем. Сейчас распространяется
Verilog AMS (Analog Mixed Signal) - новый вариант языка, который
используется для описания смешанных, аналого-цифровых схем.

В 1987 году был объявлен стандарт другого языка высокого уровня
для описания принципиальных схем VHDL (Very high - speed IC Hardway
Description Language) IEEE Std. 1076-1987. В 1993г. появилась новая
редакция стандарта IEEE Std. 1164-1993.

VHDL обладает большей универсальностью и может быть использован не только для описания моделей цифровых электронных схем, но и для других моделей (например, модели экосистемы). Однако из-за своих расширенных возможностей VHDL проигрывает в эффективности и простоте, то есть на описание одной и той же конструкции в Verilog-е потребуется в 3-4 раза меньше символов, чем в VHDL.

В 2005г. был опубликован стандарт языка проектирования и верификации моделей системного уровня System C (IEEE 1666-2005). Язык проектирования System C реализован в виде библиотеки языка C ++ с открытым кодом, которая включает ядро событийного, поведенческого, транзакционного моделирования. System C наряду с допустимыми конструкциями языка C ++ включает также такие понятия, которые присущи “чистым” языкам проектирования аппаратур Verilog и VHDL, в частности: интерфейсы, сигналы, событийность, процессы и т.д.

Как правило, существующие универсальные и специализированные языки, в том числе и языки описания устройств, построены недедуктивным образом. При таком подходе громоздкость языка зависит от той предметной области и понятий, для которых он создан. Поэтому при расширении области применения языка его громоздкость увеличивается. Таким образом, универсальность приобретает за счет его громоздкости. Отсюда, неудивительно, что язык описания устройств – это фактически совокупность многих узкоспециализированных языков.

Создание новых языков описания цифровых схем и соответствующих алгоритмов, удовлетворяющих современным требованиям, продолжает оставаться актуальной задачей.

Целью диссертационной работы является дальнейшее развитие языка Alex и его применение в области автоматизации проектирования цифровых схем.

Язык Alex построен дедуктивным принципом и лишен вышеуказанных недостатков. Он является аксиоматической теорией. В этом языке описанием схемы является конечная последовательность символов из некоторого алфавита. Естественно, не любая последовательность символов из этого алфавита является описанием схемы. Теоремами этой аксиоматической теории являются те последовательности, которые являются описанием схем. Доказывается теорема о записи, указывающая условия, которые необходимы и достаточны, чтобы данная последовательность символов являлась описанием некоторой схемы. Эта теорема фактически является теоремой о полноте данной аксиоматической теории. Таким образом, язык Alex обладает следующими особенностями:

1. В отличие от других аналогичных языков, Alex HDL построен дедуктивным (аксиоматическим) принципом, что позволяет создать

алгебру эквивалентных и иных преобразований с помощью их описания.

2. Он является оптимальным: длина (число символов) описания совпадает с числом рёбер графа (если элементы трактовать как вершины, а связи между ними - рёбра). В этом отношении Verilog HDL примерно в два раза хуже оптимального.
3. Любая подсхема с одним выходом, входы которой являются входами исходной схемы, в описании на языке Alex изображается как сплошной отрезок символов.
4. Функциональное моделирование схемы осуществляется одним «просмотром» описания схемы, что исключает затраты времени на поиски, тем самым обеспечивая большую скорость верификации, тестирования и всех тех процедур, в которых как составная часть присутствует функциональное моделирование.
5. Теорема о записи (моделирования) схемы позволяет осуществить эффективный синтаксический контроль правильности построения описания схемы.
6. Язык Alex позволяет формальными методами осуществить переход из одного уровня детализации описания схемы к другому. Эта процедура может быть использована при проектировании схем «сверху-вниз» или «снизу-вверх».

Методы исследований.

Использовались методы математической логики, теории алгоритмов, теории булевых функций, теории автоматов.

Научная новизна.

1. Разработаны новые методы быстрого функционального моделирования цифровых схем с использованием языка Alex, которые успешно апробированы на эталонных схемах серий ISCAS-85, ISCAS-89, ITC'-99.
2. Разработан метод наилучшего приближения булевых функций многих переменных функциями меньшим числом аргументов.
3. Разработан метод выделения активной части цифровой схемы с одним выходом с использованием метода наилучшего приближения булевых функций.
4. Расширен язык Alex для описания схем, построенных из элементов со многими выходами.
5. Разработан новый метод временной синхронизации цифровых схем.
6. Построено новое стандартное представление булевых функций с использованием булевых дифференциалов.
7. Разработан метод быстрой верификации с использованием активной части схемы.

Основные положения, выносимые на защиту:

1. Метод наилучшего приближения булевых функций многих переменных функциями меньшим числом аргументов.
2. Метод выделения активной части цифровой схемы с одним выходом с использованием метода наилучшего приближения булевых функций.
3. Расширение языка Alex для описания схем, построенных из элементов со многими выходами.
4. Метод быстрой верификации с использованием активной части схемы.
5. Методы быстрого функционального моделирования цифровой схемы с использованием языка Alex.

Теоретическая и практическая ценность. Результаты, полученные в диссертационной работе, носят как теоретический, так и прикладной характер. Эксперименты, проведенные на эталонных схемах, показали, что полученные результаты могут быть успешно использованы в области автоматизации проектирования цифровых схем.

Апробация полученных результатов. Основные результаты диссертации докладывались на второй международной IEEE конференции ICCSC'04 (Москва, 2004г.), международных конференциях SICPRO'04, SICPRO'08, SICPRO'09 (Москва), третьей, шестой, девятой, одиннадцатой международных научно-практических конференциях "Исследование, разработка и применение высоких технологий в промышленности" (Санкт-Петербург, 2007г., 2008г., 2010г., 2011г.), на годичных научных конференциях РАУ (Ереван, 2006г., 2007г., 2008г., 2009г., 2010г.), на годичной научной конференции ГИУА (Ереван, 2006г.), конференции ГИУА, посвященной 70-летию академика НАН РА Закаряна В.С (Ереван, 2006г.), на научно-методическом семинаре факультета прикладной математики и информатики РАУ, на научно-методическом семинаре кафедры ИТиА факультета КСиИ ГИУА.

Публикации. По теме диссертации опубликовано 30 работ.

Структура и краткое содержание работы.

Диссертация состоит из введения, пяти глав, заключения, списка использованной литературы, содержащего 127 наименований и двух приложений (актов внедрения). Работа изложена на 227 страницах.

Введение посвящено краткому описанию возникновения языков описания цифровых схем Verilog, VHDL и System C, которые стали международными стандартами. Обосновывается актуальность темы, формулируются основные цели, задачи и методы их решения.

Первая глава посвящена описанию нового отредактированного варианта языка Alex и его расширению для описания схем, построенных из элементов со многими входами и выходами, содержащих также запоминающие элементы.

Фиксируем множество $X = \{0, 1, x_1, x_2, \dots, x_n, \dots\}$, называемое множеством сигналов. Введем следующие первичные понятия:

а) **Элементарная схема.** Она не имеет входов и имеет один выход, и в любой дискретный момент времени на его выходе образуется некоторый сигнал из X (рис. 1а).

б) **Функциональный элемент.** Он имеет n входов и один выход, реализует булеву функцию n переменных (рис. 1б).

в) **Элемент ветвления.** Он имеет один вход и m ($m \geq 1$) выходов (рис. 1в) и обладает тем свойством, что если на его вход в данный момент времени поступает некоторый сигнал из X , то в этот же момент на всех его m выходах образуется тот же сигнал.

Элемент ветвления обозначим через M (с индексом или без).

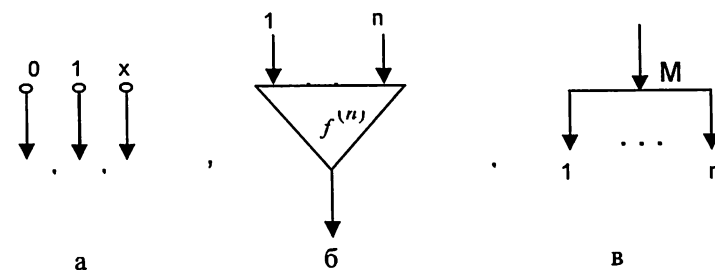


Рис. 1.

Комбинационную схему с одним выходом определим следующим образом (в дальнейшем схему указанного типа будем называть просто схемой, поскольку из контекста ясно, о каком типе схемы идет речь).

Определение 1.1. Базис индукции:

1) Любая элементарная схема является схемой. Схемы 1 и 0 не имеют входов, а выходами являются выходы этих же элементарных схем. Входом схемы типа x является сама элементарная схема, выходом – выход этой элементарной схемы.

2) Пусть z_1, \dots, z_n – элементарные схемы, $f^{(n)}$ – функциональный элемент. Конструкция, полученная отождествлением выходов z_1, \dots, z_n с соответствующими входами $f^{(n)}$, является схемой (рис. 2а). Входами этой схемы являются элементарные схемы из z_1, \dots, z_n типа x , выходом – выход $f^{(n)}$.

Индукционный шаг.

1) Пусть определена схема S' , которая наряду с другими входами

содержит также входы x_1, \dots, x_m . Пусть z – элементарная схема, M_k – элемент ветвления с m выходами (z и M_k не входят в S'). Тогда отождествляя выходы M_k с соответствующими входами x_1, \dots, x_m , а выход z – со входом M_k , получится некоторая конструкция S , которая также является схемой (рис. 2, б). Входами S являются z , если z – символ типа x , и все входы S , кроме x_1, \dots, x_m , выходом – выход S' .

2) Пусть определены непересекающиеся схемы S_1 и S_2 , где S_2 , наряду с другими входами, имеет вход x . Тогда при отождествлении выхода S_1 со входом x схемы S_2 , получится некоторая конструкция S , которая также является схемой (рис. 2в). Входами S являются все входы S_1 и S_2 , кроме x , выходом – выход S_2 .

3) Других схем нет.

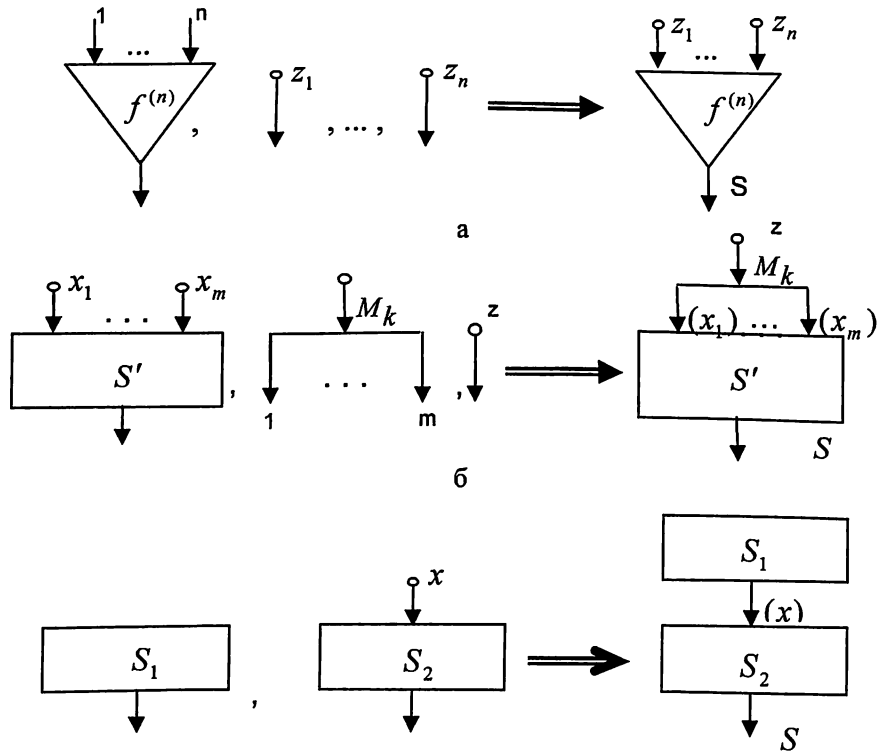


Рис. 2.

ОПИСАНИЕ СХЕМЫ НА ЯЗЫКЕ ALEX

Описание схемы S на языке Alex обозначим через $A(S)$. Описание схемы $A(S)$ индуктивно определяется следующим образом.

Определение 1.2. Базис индукции.

- 1) Если S является элементарной схемой z , то $A(S) = z(0)$.
- 2) Если схема S построена из функционального элемента $f^{(n)}$ и элементарных схем z_1, \dots, z_n , согласно второму пункту базиса индукции определения схемы, то

$$A(S) = f^{(n)} z_1(0) \dots z_n(0).$$

Индукционный шаг.

- 1) Пусть определена запись $A(S')$ схемы S' , а S получена из S' , z и M_k , согласно первому пункту индукционного шага определения схемы. Тогда $A(S)$ получится из $A(S')$ заменой $x_i(0)$ на $M_k(1)z(0)$, а все $x_i(0)$ ($i = \overline{2, m}$) – на $M_k(0)$, т.е.

$$A(S) = [A(S')]_{M_k(1)z(0), M_k(0), \dots, M_k(0)}^{x_1(0), x_2(0), \dots, x_m(0)}$$

Символы $M_k(1)$ и $M_k(0)$ называются метками.

- 2) Пусть определены $A(S_1)$ и $A(S_2)$, а схема S получена из непересекающихся схем S_1 и S_2 , согласно второму пункту индукционного шага определения схемы. Тогда

$$A(S) = [A(S_2)]_{A(S_1)}^{x(0)}.$$

Введем оператор ω , областью определения которого является множество всех слов в алфавите

$$B = \{0(0), 1(0), x_i(0), M_j(1), M_j(0), f_k(n)\},$$

где i, j, k, n – натуральные числа ($i, j, k, n \in \{1, 2, 3, \dots\}$), а областью изменения – множество целых чисел.

На ω налагаются следующие ограничения:

1. Для любой схемы S с одним выходом $\omega(A(S)) = -1$,
2. $\omega(s_1 s_2 \dots s_N) = \sum_{i=1}^N \omega(s_i)$, где s_i ($i = \overline{1, N}$) – символы из B .

$\omega(s_1 s_2 \dots s_n)$ называется **весом** $s_1 s_2 \dots s_n$.

Эти условия достаточны для однозначного определения оператора ω .
Очевидно, что

$$\omega(0(0)) = \omega(1(0)) = \omega(x_i(0)) = \omega(M_j(0)) = -1$$

$$\omega(M_j(1)) = 0, \quad \omega(f_k(n)) = n - 1.$$

Определение 1.3. Пусть $s_1 s_2 \dots s_N$ - слово из алфавита В. Если оно содержит единственное вхождение метки $M_i(1)$ и, если слово содержит минимальный отрезок с весом -1 непосредственно следующий за $M_i(1)$, то этот отрезок называется **описанием** метки $M_i(1)$ в $s_1 s_2 \dots s_N$.

Описание $M_i(1)$ обозначим через $OnM_i(1)$.

Определение 1.4. Пусть $s_1 s_2 \dots s_N$ - слово в алфавите В. Пусть любая метка $M_i(1)$, содержащаяся в $s_1 s_2 \dots s_N$, имеет единственное вхождение и имеет описание. Если в $s_1 s_2 \dots s_N$

- 1) $OnM_i(1)$ содержит $M_j(1)$ или $M_j(0)$, то $M_i(1)$ зависит от $M_j(1)$;
- 2) $M_i(1)$ зависит от $M_k(1)$, а $M_k(1)$ зависит от $M_j(1)$, то $M_i(1)$ зависит от $M_j(1)$;
- 3) $M_i(1)$ зависит от $M_j(1)$ только согласно п.п.1,2.

Теорема 1.1. (о записи) Слово

$$s_1 s_2 \dots s_N \quad (1)$$

в алфавите В является описанием некоторой схемы с одним выходом тогда и только тогда, когда выполняются следующие условия:

- 1) $\omega(s_1 s_2 \dots s_N) = -1$
- 2) $\omega(s_1 s_2 \dots s_k) \geq 0$ для любого $k < N$,
- 3) если $M_i(0)$ содержится в (1), то (1) содержит и $M_i(1)$, причем точно один раз.
- 4) если $M_i(1)$ содержится в (1), то $M_i(1)$ не зависит от $M_i(1)$.

Определение логической схемы со многими выходами и её описание рассматривается аналогичным образом. Схема со многими выходами

получается из схемы с одним выходом, устранением её выходного элемента (т.е. элемента, выходом которого является выход схемы) (рис. 3).

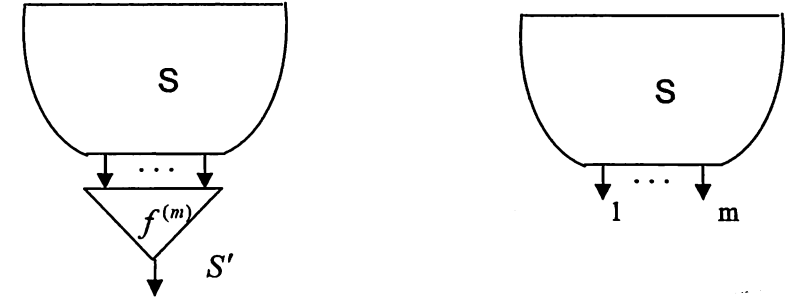


Рис. 3

Для данной схемы теорема о записи выглядит следующим образом:

Теорема 1.2. Слово $s_1 s_2 \dots s_N$ из символов множества В является записью некоторой схемы с m выходами тогда и только тогда, когда выполняются следующие условия:

- 1) $\omega(s_1 s_2 \dots s_N) = -m$,
- 2) $\omega(s_1 s_2 \dots s_k) \geq -m$ для любого $k < N$,
- 3) если $M_i(0)$ содержится в (1), то (1) содержит и $M_i(1)$, причем точно один раз,
- 4) если $M_i(1)$ содержится в (1), то $M_i(1)$ не зависит от $M_i(1)$.

Описание $A(S)$ схемы S со многими выходами можно представить в следующем виде:

$$A(S) = A(S,1) A(S,2) \dots A(S,m),$$

где $A(S,1)$ - первый минимальный отрезок в $A(S)$ с весом -1, $A(S,2)$ - следующий аналогичный отрезок и т.д. Назовем $A(S,i)$ ($i = 1,2, \dots, m$) i -ым компонентом $A(S)$. Очевидно, $A(S,i)$ ($i = 1,2, \dots, m$) соответствует i -ому выходу схемы и, если он содержит метку $M_r(0)$, но не содержит $M_r(1)$, то одна из меток $M_r(0)$ (например, первый слева) заменив на $M_r(1)OnM_r(1)$, а остальные, оставив без изменения, и эту процедуру производя относительно всех меток в $A(S,i)$, получится запись той подсхемы, выходом которой является i -ый выход схемы, а входами являются некоторые входы схемы

S . Полученный таким образом отрезок обозначим через $A[S, i]$ и назовем i -ым расширенным компонентом описания $A(S)$.

В первой главе рассмотрены также схемы из логических и запоминающих элементов.

Определение 1.5. Скажем, что i -ый выход схемы S зависит от входа x_j , если i -ый расширенный компонент $A[S, i]$ описания $A(S)$ содержит символ $x_j(0)$.

Пусть τ - элемент памяти, имеющий один вход и один выход, обладающий тем свойством, что если в данный момент времени на его вход поступает сигнал $z(0) \in X$, то в следующий момент времени на его выходе образуется тот же сигнал $z(0)$.

Определение 1.6. Пусть S' - схема с $n+k$ входами и $m+k$ выходами. Если выходы схемы S' с номерами $m+1, \dots, m+k$ отождествить, соответственно, с входами элементов памяти τ_1, \dots, τ_k , а выходы этих элементов памяти отождествить, соответственно, с входами x_{n+1}, \dots, x_{n+k} схемы S' , то получится некоторая конструкция S (рис. 4,б), которая называется **схемой из логических и запоминающих элементов (СЛЗЭ)**.

Заметим, что $k \geq 0$, поэтому класс СЛЗЭ охватывает также логические схемы со многими выходами.

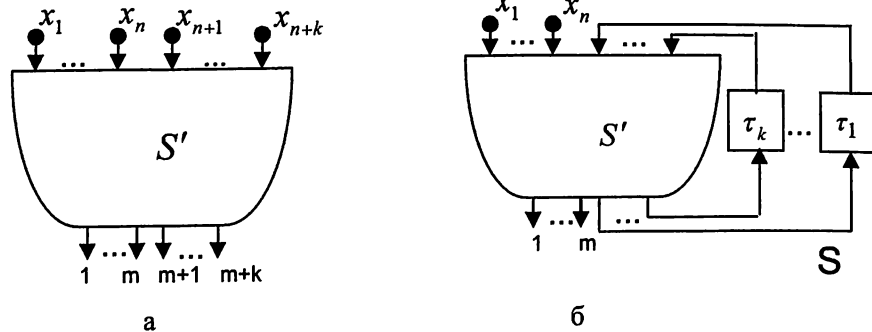


Рис. 4.

Здесь выбор данного типа запоминающего элемента весьма условный и можно использовать любой другой тип (например, статический триггер и т.п.).

При синхронной (цифровой) работе схемы предполагается, что частота

поступления на входы x_1, \dots, x_n наборов сигналов совпадает с частотой переключения запоминающих элементов τ_1, \dots, τ_k , т.е. время "задержки" сигналов в этих элементах равно времени между двумя соседними поступлениями наборов входных сигналов на входы x_1, \dots, x_n .

Замечание 1.1. Заметим, что определение 1.6 при построении СЛЗЭ предполагает параллельно, одним шагом, ввести запоминающие элементы τ_1, \dots, τ_k схемы, изображенной на рис. 4,а, которая является логической схемой со многими выходами (без запоминающих элементов). Не трудно заметить, что это построение можно было осуществить и последовательно. Это означает, что введение запоминающих элементов можно применить и относительно СЛЗЭ.

Аналогичным образом определяются понятия схемы из логических и запоминающих элементов и её записи.

Рассматривается функциональное моделирование комбинационной схемы и схемы из логических и запоминающих элементов.

Язык *Alex* позволяет эффективно осуществить процедуру функционального моделирования комбинационной схемы. Под функциональным моделированием понимается нахождение значений сигналов на выходах схемы при наличии значений сигналов на её входах. Продемонстрируем эту процедуру в виде алгоритма на частном примере, однако в нем сущность алгоритма и техника его осуществления обрисовываются четко.

Рассмотрим комбинационную схему *c17* из эталонных схем *ISCAS-85*. Она имеет пять входов и два выхода и описана на упрощенном варианте языка *Verilog*:

```

INPUT(G1gat)
INPUT(G2gat)
INPUT(G3gat)
INPUT(G6gat)
INPUT(G7gat)
OUTPUT(G22gat)
OUTPUT(G23gat)
G10gat = nand(G1gat, G3gat)
G11gat = nand(G3gat, G6gat)
G16gat = nand(G2gat, G11gat)
G19gat = nand(G11gat, G7gat)
G22gat = nand(G10gat, G16gat)

```

$$G23gat = \text{nand}(G16gat, G19gat)$$

Её графическое изображение приведено на рис.5.

С целью представления схемы на языке Alex приведем основную идею построения алгоритма перевода описания схемы с языка Verilog на язык Alex. Заметим, что из описания схемы на языке Verilog видно, что все элементы (входы также считаются элементами) пронумерованы таким образом, что разным элементам приписаны разные номера, а входы элементов и выходы схемы определенным образом упорядочены.

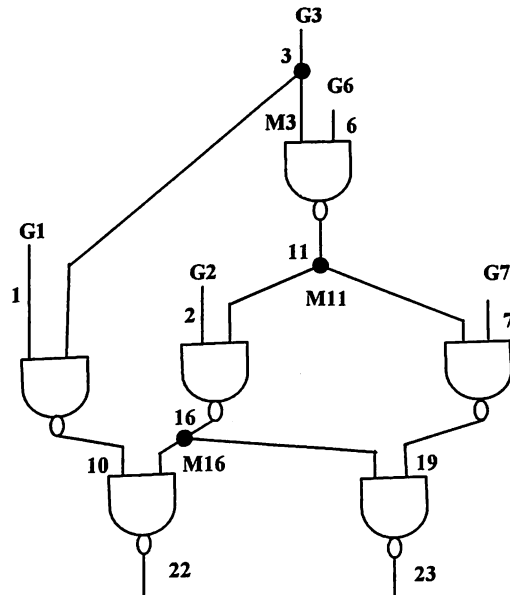


Рис. 5.

Например, строка $G16gat = \text{nand}(G2gat, G11gat)$ показывает, что элемент **nand** с номером 16, выход которого обозначен **G16gat**, имеет два входа, причем к первому входу подключен **G2gat**, т.е. выход элемента с номером 2, а ко второму входу - **G11gat**, т.е. выход элемента с номером 11. Что касается выходов схемы, то они упорядочены в описании на языке Verilog “сверху-вниз”, т.е. в нашем примере по порядку **OUTPUT(G22gat)**, **OUTPUT(G23gat)**.

Далее, поскольку построение записи схемы на языке Alex осуществляется относительно выходов схемы в том порядке, в каком они расположены по упорядочению выходов, то процесс построения записи начинается с первого выхода и кончается последним выходом. Первым

выходом (**Output(G22gat)**) является выход элемента **nand** с номером 22. Поэтому первым символом записи схемы на языке Alex должен быть **nand22(2)**. Первым входом этого элемента является выход элемента **nand** с номером 10. Поэтому вторым символом записи должен быть **nand10(2)**. Далее, первым входом элемента с номером 10 является выход элемента (вход схемы) с номером 1. Поэтому третьим символом записи будет **G1(0)**. Поскольку элемент **G1gat** не имеет входов, переходим к рассмотрению второго входа элемента с номером 10. Этот вход является выходом элемента с номером 3. Однако выход элемента с номером 3 ветвится, и только одна из ветвей является вторым входом элемента с номером 10. Поскольку эта точка ветвления рассматривается впервые, то последующими символами записи схемы будут метка **M3(1)** и её описание, т.е. **G3(0)**. Если точка ветвления рассматривается не впервые, то вместо указанной пары символов **M3(1)** и **G3(0)** записывается единственный символ **M3(0)**. Наличием факта присутствия точки ветвления на выходе элемента с номером 3 обнаруживается повторением символа **G3(0)** в двух строках Verilog – описании схемы, а именно: **G10gat=nand(G1gat, G3gat)** и **G11gat=nand(G3gat, G6gat)**. После завершения рассмотрения всех входов данного элемента рассматривается очередной вход того элемента, последний рассмотренный вход которого был выходом данного элемента. Очередным шагом является переход к рассмотрению следующего выхода схемы. Запись схемы полностью завершается с завершением записи относительно её последнего выхода.

Записью схемы **c17** является:

```
nand22(2)nand10(2)G1(0)M3(1)G3(0)M16(1)nand16(2)G2(0)M11(1)nand11(2)M3(0)G6(0) nand23(2)M16(0)nand19(2)M11(0)G7(0)
```

Описаниями меток **M3(1)**, **M16(1)** и **M11(1)** в записи схемы являются соответственно:

OnM3(1)=G3(0),

OnM16(1)=nand16(2)G2(0)M11(1)nand11(2)M3(0)G6(0),

OnM11(1)= nand11(2)M3(0)G6(0).

Поскольку $Mi(1)OnMi(1)$ и $Mi(0)$ ($i = 1, 2, \dots$) содержательно эквивалентны, то они взаимно заменяемы. С целью наиболее эффективного осуществления алгоритма функционального моделирования схемы, её запись преобразуем таким образом, чтобы для любого индекса i все метки $Mi(0)$ в записи предшествовали $Mi(1)$. После такого преобразования записи получим:

```
nand22(2)nand10(2)G1(0)M3(0)M16(0)nand23(2)M16(1)nand16(2)G2(0)M11(0)nand19(2) M11(1)nand11(2)M3(1)G3(0)G6(0)G7(0)
```

В том случае, когда после осуществления данной процедуры над

схемой, описанной на языке Alex, нет необходимости вернуться к описанию на языке Verilog, можно запись схемы разгрузить от лишней информации. Такой процедурой, в частности, является функциональное моделирование, для которого лишними являются номера элементов. После исключения номеров элементов, для нашего примера получим соответствующий вариант записи:

$\text{nand}(2)\text{nand}(2)\text{G1}(0)\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)\text{G2}(0)\text{M11}(0)\text{nand}(2)\text{M11}(1)\text{nand}(2)\text{M3}(1)\text{G3}(0)\text{G6}(0)\text{G7}(0)$.

Если символы (входы) $\text{G1}(0)$, $\text{G2}(0)$, $\text{G3}(0)$, $\text{G6}(0)$ и $\text{G7}(0)$ заменить метаобозначениями x_1, x_2, x_3, x_6 и x_7 соответственно, то получим более наглядное представление записи:

$\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_2\text{M11}(0)\text{nand}(2)\text{M11}(1)\text{nand}(2)\text{M3}(1) x_3x_6x_7$.

Функциональное моделирование схемы для данного набора входных сигналов осуществляется последовательным (в любом порядке) применением следующих процедур:

- 1) $f(k)a_1 \dots a_k$ заменить на $f(k)(a_1, \dots, a_k)$, где $a_i \in \{0,1\}$, $i = 1, 2, \dots, k$.
- 2) $Mi(1)a$ и $Mi(0)$ ($i = 1, 2, \dots$) заменить на $a \in \{0,1\}$.

Во избежание многочисленных поисков символов типа $Mi(0)$ при замене $Mi(0)$ на a , если пара символов $Mi(1)a$ уже заменена на a , дополнительно хранится текущая информация (i, a, m_i) , где i – номер метки $Mi(1)$, m_i – текущее значение числа вхождений символов $Mi(0)$ в записи после очередного применения п.2. После каждого применения этого пункта число m_i уменьшается на единицу. При текущем значении $m_i=0$ тройка $(i, a, 0)$ исключается из списка.

Ниже, шаг за шагом, представлена работа алгоритма над схемой с приведенной записью. Пусть на её входы x_1, x_2, x_3, x_6, x_7 подан набор сигналов $(0,0,1,0,1)$. Ход применения алгоритма моделирования следующий:

- 1) $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_2\text{M11}(0)\text{nand}(2)\text{M11}(1)\text{nand}(2)\text{M3}(1)x_3x_6x_7$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_2\text{M11}(0)$
 $\text{nand}(2)\text{M11}(1)\text{nand}(2)\text{M3}(1)x_3x_61$
- 2) $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_2\text{M11}(0)\text{nand}(2)\text{M11}(1)\text{nand}(2)\text{M3}(1)x_301$
- 3) $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_2\text{M11}(0)\text{nand}(2)\text{M11}(1)\text{nand}(2)\text{M3}(1)01$
- 4) $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_2\text{M11}(0)\text{nand}(2)\text{M11}(1)\text{nand}(2)101$

- 5) $\{(3, 1, 1)\}$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_2\text{M11}(0)\text{nand}(2)\text{M11}(1)11$
- 6) $\{(3, 1, 1)\}$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_2\text{M11}(0)\text{nand}(2)11$
- 7) $\{(3, 1, 1), (11, 1, 1)\}$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_2\text{M11}(0)0$
- 8) $\{(3, 1, 1), (11, 1, 1)\}$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)x_210$
- 9) $\{(3, 1, 1), (11, 1, 0)\} = \{(3, 1, 1)\}$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)\text{nand}(2)010$
- 10) $\{(3, 1, 1)\}$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)\text{M16}(1)10$
- 11) $\{(3, 1, 1), (16, 1, 1)\}$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)\text{nand}(2)10$
- 12) $\{(3, 1, 1), (16, 1, 1)\}$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)\text{M16}(0)1$
- 13) $\{(3, 1, 1), (16, 1, 0)\} = \{(3, 1, 1)\}$
 $\text{nand}(2)\text{nand}(2)x_1\text{M3}(0)11$
- 14) $\{(3, 1, 0)\} = \emptyset$
 $\text{nand}(2)\text{nand}(2)x_111$
- 15) $\text{nand}(2)\text{nand}(2)0111$
- 16) $\text{nand}(2)111$
- 17) 01

Итак, набору входных сигналов $(0,0,1,0,1)$ соответствует набор выходных сигналов $(0,1)$.

Как видно из рассмотренного примера, функциональное моделирование комбинационной схемы осуществляется одним просмотром записи схемы практически без поисков. Поиски осуществляются только в случае определения значений меток типа $Mi(0)$ во вспомогательном списке значений этих меток. Однако объем этого списка динамически меняется за счет прибавления и удаления строк типов (i, a, m_i) ($m_i \neq 0$) и $(i, a, 0)$ соответственно, и его среднее значение для наиболее часто встречающихся на практике схем, является незначительным.

Аналогичным образом рассматривается функциональное моделирование схем из логических и запоминающих элементов.

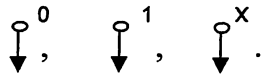
Язык Alex описания цифровых схем достаточно хорошо изучен для схем из примитивных элементов, т.е. элементов, реализующих булевы

функции типа конъюнкции, дизъюнкции, отрицания и т.д. Однако при проектировании цифровых схем часто используются и схемы, элементами которых являются более сложные объекты.

Приводится расширение языка Alex для описания схем, построенных из элементов со многими выходами.

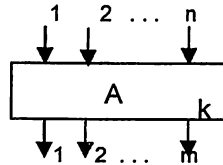
Перечислим некоторые вспомогательные объекты, которые нужны для уточнения понятия схемы из элементов со многими выходами.

1) Элементарные схемы:



Содержательный смысл этих объектов заключается в том, что в любой дискретный момент времени (время дискретизировано) на выходах этих объектов образуются, соответственно, сигналы 0, 1 и x.

2) Автоматный элемент:



Он имеет n входов, m выходов, k состояний и реализует конечный автомат. Для любого такого автомата фиксировано некоторое множество

$$M_A = \{ \langle i_r, j_r \rangle / r = 1, 2, \dots, p \}$$

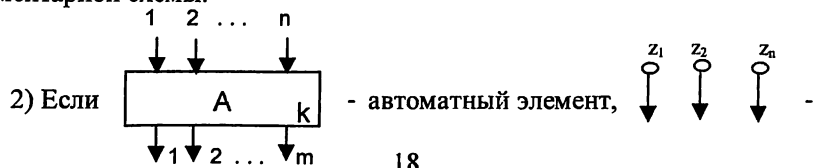
пар номеров входов и выходов.

Содержательный смысл пары $\langle i_r, j_r \rangle$ заключается в том, что в любой дискретный момент времени выход с номером j_r не зависит от входа с номером i_r (номера i_1, i_2, \dots, i_r могут повторяться).

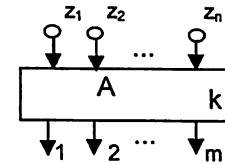
Определим схему из элементов со многими выходами следующим образом:

Определение 1.7. Базис индукции.

1) Элементарная схема \downarrow^z ($z \in \{0, 1, x\}$) является схемой. Схемы \downarrow^0 и \downarrow^1 имеют выходы и не имеют входов, а схема \downarrow^x имеет и вход и выход. Входом является сама элементарная схема \downarrow^x , выходом – выход этой элементарной схемы.



элементарные схемы, то конструкция



является схемой.

Входами этой схемы являются те элементарные схемы z_i ($1 \leq i \leq n$), которые являются элементарными схемами типа x_i . Выходами являются выходы автоматного элемента A.

Индукционный шаг. 1. Пусть определены схемы S_1 и S_2 , которые не пересекаются, т.е. не имеют общих элементов (рис. 6.a). Тогда их упорядоченное (слева-направо) объединение S также является схемой (рис. 6.b). Входами и выходами схемы S являются, соответственно, все входы и все выходы схем S_1 и S_2 .

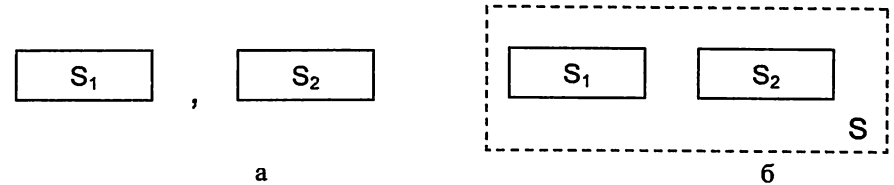


Рис.6.

2. Пусть определена схема S' , которая наряду с остальными входами имеет также вход x , а выход с номером i в любой дискретный момент времени t ($t = 1, 2, \dots$) функционально не зависит от x (рис. 7.a). Тогда, если выход схемы S' с номером i отождествить с входом x этой же схемы, получится некоторая конструкция S , которая также является схемой (рис. 7.b). Входами схемы S являются все входы схемы S' , кроме входа x , а выходами – все выходы S' , кроме выхода с номером i .

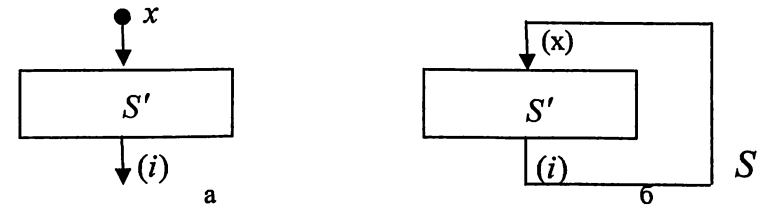


Рис. 7.

3. Других схем нет.

Описание схемы из элементов со многими выходами на языке Alex
Определение 1.8. Базис индукции.

1) Если схема S является элементарной схемой z ($z \in \{0,1,x\}$), то ее описанием будет $A(x) = z(0,1)$

2) Если S является схемой согласно п.2 базиса индукции определения схемы из элементов со многими выходами, то

$$A(S) = A(n,1)z_1(0,1)z_2(0,2) \dots z_n(0,1)A(0,2)A(0,3) \dots A(0,m)$$

Индукционный шаг.

1) Пусть определены описания $A(S_1)$ и $A(S_2)$ для схем S_1 и S_2 , соответственно, а схема S получается из схем S_1 и S_2 согласно п.1 индукционного шага определения 1.7. Тогда

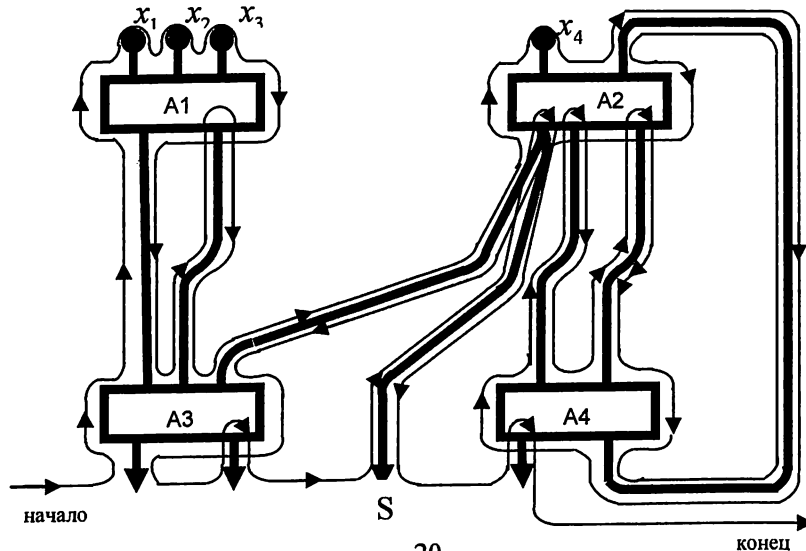
$$A(S) = A(S_1)A(S_2)$$

2) Пусть определено описание $A(S')$ схемы S' , а S получается из S' согласно п.2 индукционного шага определения 1.7. Тогда

$$A(S) = [A(S')]_{A(S',i)}^{x(0,1)}$$

где $A(S',i)$ является описанием максимальной подсхемы схемы S' относительно ее i -ого выхода (напомним, что входы этой подсхемы являются входами схемы S').

Пример. Рассмотрим схему из элементов со многими выходами:



Записью этой схемы является:

$$A(S) = A3(3,1)A1(3,1)x1(0,1)x2(0,1)x3(0,1)A1(0,2)A2(2,1)x4(0,1) \\ A4(2,2)A2(0,2)A2(0,3)A3(0,2)A2(0,1)A4(0,1)$$

Используя описание схемы на языке Alex можно предложить алгоритмы обработки схем, в частности, алгоритм синтаксического контроля, верификации, выделения подсхемы, удовлетворяющей конкретным критериям преобразования схем и т.д.

Во второй главе рассматривается ряд важных процедур над схемами и, применением языка Alex, решаются соответствующие задачи. Этими процедурами являются: эквивалентные преобразования цифровых схем, разбиение схемы на части, выделение и вставка подсхемы, переход из одного уровня детализации описания схемы к другому, выделение и удаление максимальной подсхемы с одним выходом из цифровой схемы, временная синхронизация цифровых схем и т.д.

При проектировании больших интегральных схем важной задачей является уменьшение потребляемой мощности элементов.

Актуальность задачи связана с популярностью сферы мобильных устройств, где, с одной стороны, имеет место тенденция к уменьшению размера устройств, а с другой стороны, к увеличению их функциональной сложности. Поэтому вычисление значения потребляемой мощности приобретает особую важность.

Заметим, что если некоторый пассивный элемент схемы (пассивным является элемент, имеющий малую динамическую активность) заменить константой, то это позволит уменьшить потребляемую мощность схемы. В связи с этим становится актуальной задача анализа схемы с целью выявления ее пассивных частей применительно к конкретному множеству входных сигналов.

Предложены алгоритм определения подсхем, статистическая активность выходов которых не меньше α и процедуру замены таких схем на константу.

Для проверки работы алгоритма были проведены эксперименты для схем B17, B18, B19 из набора тестовых схем ИТС'99. Были выбраны 5 случайно сгенерированных сред выполнения с 10000 входными сигналами каждая. Для каждой из них был апробирован выше представленный алгоритм. Если принять, что среднее энергопотребление элементов схемы одинаково, то в лучшем случае было получено уменьшение общего энергопотребления схемы на 34%, а в худшем - на 5%. Результаты представлены в Таблице 1. Как и следовало ожидать, чем меньше α , тем более большую подсхему удастся выделить и, наоборот, с увеличением α ,

уменьшается размер пассивной схемы. Отметим, что даже при требовании максимальной точности ($\alpha=1$), алгоритм выявляет значительные пассивные куски схем.

Таблица 1. Выделение пассивных элементов схемы.

	Количество элементов схемы	α	ε	Количество выделенных элементов
B17	39665	0.75	3	5647
		0.8	3	2589
		0.85	2	1378
		0.9	1	1045
		1.0	1	1045
B18	114621	0.75	4	39768
		0.8	4	24899
		0.85	3	15800
		0.9	2	11405
		1.0	1	10850
B19	231320	0.75	5	40895
		0.8	4	39871
		0.85	4	35489
		0.9	2	30358
		1.0	1	21522

Третья глава посвящена наилучшим приближениям булевых функций многих переменных функциями меньшим числом переменных. Рассматривается также задача наилучшего приближения цифровых схем с одним выходом. Эта процедура называется выделением активной части схемы. Вводится понятие динамической активности элементов схемы из примитивных элементов (типа "И", "ИЛИ", "НЕ" и т.д.). Разрабатывается алгоритм вычисления этих активностей.

Активность аргумента булевой функции показывает меру зависимости функции от этого аргумента. Отсюда следует, что если функция $f(x_1, \dots, x_n)$ приближается функцией $\varphi(x_{i_1}, \dots, x_{i_k})$, где $1 \leq i_1 < \dots < i_k \leq n$ и $k < n$, то аргументы x_{i_1}, \dots, x_{i_k} должны быть выбраны из наиболее активных аргументов x_1, \dots, x_n функции $f(x_1, \dots, x_n)$.

Рассматриваются два варианта формулировки задачи приближения

функции $f(x_1, \dots, x_n)$ функциями меньшим числом аргументов:

Первый вариант. Пусть заданы булева функция $f(x_1, \dots, x_n)$ и натуральное число $k < n$. Требуется найти булеву функцию k переменных $\varphi(x_{i_1}, \dots, x_{i_k})$ ($1 \leq i_1 < i_2 < \dots < i_k \leq n$) таким образом, чтобы расстояние между функциями f и φ , т.е. $\rho(f, \varphi) = \|f \oplus \varphi\|$ было минимальным.

Второй вариант. Пусть заданы булева функция $f(x_1, \dots, x_n)$ и число $\varepsilon > 0$. Требуется найти булеву функцию

$$\varphi(x_{i_1}, \dots, x_{i_k})$$

такую, что

$$\rho(f, \varphi) < \varepsilon$$

при минимальном $k \leq n$.

Ясно, что обе задачи для больших n трудно решаемые. Поэтому вычисляя (точными или статистическими методами) активности аргументов функции $f(x_1, \dots, x_n)$ для первого варианта, следует подобрать первые k наиболее активных аргументов (предположим x_1, \dots, x_k), а затем построить функцию $\varphi_k(x_1, \dots, x_k)$ следующим образом.

Для любого набора $(\alpha_1, \dots, \alpha_k)$ из нулей и единиц:

$$\varphi_k(\alpha_1, \dots, \alpha_k) = \begin{cases} 1, & \text{если } \|f(\alpha_1, \dots, \alpha_k, x_{k+1}, \dots, x_n)\| > \frac{1}{2} \\ 0, & \text{если } \|f(\alpha_1, \dots, \alpha_k, x_{k+1}, \dots, x_n)\| < \frac{1}{2} \\ 0 \text{ или } 1, & \text{если } \|f(\alpha_1, \dots, \alpha_k, x_{k+1}, \dots, x_n)\| = \frac{1}{2} \end{cases}$$

Для второго варианта подбор k выполняется таким образом, чтобы

$$\rho(f, \varphi_k) < \varepsilon,$$

а для любого $i < k$

$$\rho(f, \varphi_i) \geq \varepsilon.$$

Разработан новый эффективный метод вычисления динамических активностей элементов цифровых схем. Динамическая активность элемента имеет вероятностный смысл: она, для данного фиксированного распределения вероятностей появления наборов входных сигналов схемы,

является вероятностью того, что в моменты времени t и $t+1$ на выходе данного элемента появляются разные сигналы, т.е. в момент времени $t+1$ происходит переключение состояния выхода элемента. Предполагается, что события появления наборов входных сигналов схемы в моменты времени t и $t+1$ независимы.

Любой такой переход является большой нагрузкой для элемента, и он поглощает энергию, равную

$$\frac{CU^2}{2},$$

где C - емкость на выходе узла, перезаряжаемая в процессе переключения, U - напряжение питания. И если выход элемента в среднем переключается с частотой $n \text{ сек}^{-1}$, то он потребляет мощность

$$\frac{CU^2}{2}n.$$

Эта частота переключения выхода элемента зависит не только от частоты работы схемы, а особенно - от распределения вероятностей появления наборов входных символов.

Динамическая активность элемента e , выход которого относительно входов схемы реализует функцию $f_e(x_1, \dots, x_n)$, вычисляется по следующей формуле:

$$A_e = 2 \cdot \|f_e\| \cdot \|\bar{f}_e\|,$$

где

$$\|f_e\| = \sum_{(\alpha_1, \dots, \alpha_n)} f_e(\alpha_1, \dots, \alpha_n) P(\alpha_1, \dots, \alpha_n),$$

а $P(\alpha_1, \dots, \alpha_n)$ является вероятностью появления набора $(\alpha_1, \dots, \alpha_n)$.

Очевидно, динамическую активность элемента e можно приблизительно вычислить статистическими методами. Действительно, если схема работает с частотой $V \text{ сек}^{-1}$, а выход элемента e при этом переключается в среднем с частотой $n \text{ сек}^{-1}$, то отношение $\frac{n}{V}$ приблизительно совпадает с динамической активностью элемента.

Точность зависит от числа экспериментов при вычислении частоты переключения выхода элемента, которая оценивается методами Монте-Карло.

Предлагается динамические активности элементов вычислить, используя быстрый алгоритм моделирования схемы.

Рассматривается также задача выделения активной части цифровой схемы. Активная часть схемы с одним выходом получается из этой схемы фиксированием «пассивных» входов на такие константы, которые обеспечивают наилучшее приближение основной схемы ее активной частью. Предлагается верификацию данной схемы осуществить ее активной частью. Точность приближения регулируется выбором числа «пассивных» входов. Схемы со многими выходами рассматриваются по отдельным выходам, относительно которых она является системой подсхем с одним выходом.

Если схема не комбинационная, т.е. содержит также запоминающие элементы, то она реализует конечный автомат. Подсхема относительно данного выхода также реализует конечный автомат, но уже с одним выходом. Активности входов этой подсхемы вычисляются статистическими методами. Дальнейшая процедура выделения активной части аналогична случаю комбинационной схемы.

Ниже приводится таблица результатов построения активных частей схемы с315 серии ISCAS 85.

Таблица 2. Выделение активных частей схемы C5315.

Номер выхода	Длина записи подсхемы	Кол-во входов подсхемы	Степень приближения, % (корреляция подсхемы и ее активной части)	Длина записи активной части подсхемы	Кол-во входов активной части подсхемы
1	2	3	4	5	6
80	358	38	99,13	347	34
80	358	38	97,28	245	28
80	358	38	96,54	215	25
85	324	34	99,25	285	30
85	324	34	97,67	231	24
85	324	34	96,89	218	23

Продолжение таблицы 2.

Номер выхода	Длина записи подсхемы	Кол-во входов подсхемы	Степень приближения, % (корреляция подсхемы и ее активной части)	Длина записи активной части подсхемы	Кол-во входов активной части подсхемы
90	365	39	98,62	329	28
90	365	39	95,93	176	18
90	365	39	96,92	176	18
95	284	28	99,1	210	22
95	284	28	98,68	205	21
95	284	28	98,22	180	19
100	292	30	99,26	269	24
100	292	30	99,21	207	23
100	292	30	98,47	178	20
105	599	60	98,65	502	38
105	599	60	97,80	313	35
105	599	60	97,41	277	34
110	532	55	98,69	435	33
110	532	55	95,16	365	22
110	532	55	94,03	355	19
115	606	61	98,56	479	33
115	606	61	94,02	179	19
115	606	61	93,63	164	17

В таблице каждый выход (номер выхода указан в первом (слева)

столбце) рассматривается трижды. Во втором столбце указаны длины записи подсхемы на языке Alex. В третьем столбце указаны количество входов подсхемы. В следующих столбцах указаны соответственно – точность приближения, длины записи активных частей подсхемы и число входов активных частей. Например, в шестом столбце (для выхода с номером 80) указаны три варианта числа входов активной части: 34, 28 и 25. В первом варианте обеспечивается 99,13% степень приближения активной части подсхемы с одним выходом (с номером), которая (активная часть) имеет описание длиной 347. Во втором варианте, когда число входов активной части снизилось до 28, точность приближения несколько уменьшается (97,28%), длина описания также уменьшается до 245, а в третьем варианте указанные параметры становятся соответственно 25, 96,54% и 215. Последние три строки таблицы показывают интересные результаты для подсхемы относительно выхода с номером 115, где число входов 61 снижается до 33, 19 и 17, сохранив точности приближения соответственно 98,56%, 94,02% и 93,63%. При этих точностях число входных наборов сигналов для полной верификации от 2^{61} снижается, соответственно, до 2^{39} , 2^{19} и 2^{17} .

Четвертая глава посвящена функциональному моделированию и верификации цифровых схем. Рассматривается алгоритм быстрого моделирования схем, моделирование схем на множестве слов в алфавите $\{0,1,x,z\}$. Алгоритм реализуется относительно тестовых эталонных схем серии ISCAS-85, ISCAS-89. Результаты показывают высокую эффективность алгоритма моделирования. Рассматриваются также минимальные представления булевых функций от четырех переменных и их применение на практике.

Благодаря достижениям в области технологии изготовления интегральных схем (ИС) неуклонно уменьшаются как число дефектов, так и минимальный размер элемента кристалла. Это дает возможность разрабатывать и выпускать все более сложные и насыщенные кристаллы. Однако с ростом плотности компоновки ИС усложняются и проблемы верификации и тестирования для выявления потенциальных ошибок: кристалл сверхбольшой интегральной схемы (СБИС) может содержать миллионы внутренних схемных узлов, которые недоступны для непосредственного управления и наблюдения через контакты ввода-вывода этого кристалла. Задачи верификации и тестирования могут оказаться исключительно трудными, а затраты времени и усилий на решение проблем тестирования могут привести к значительному возрастанию себестоимости производства СБИС.

Основной составной частью верификации и тестирования является функциональное моделирование схемы. Данная глава посвящена

разработке способа быстрого моделирования схем из логических и запоминающих элементов. Метод базируется на использование языка строчного описания схем Alex.

Одним из преимуществ языка Alex является то, что в записи схемы отражена и логика моделирования. При описании схемы по уровням в любой записи встречаются или входные элементы, или элементы ветвления, описания которых уже встречались до этого, т.е. значение функции можно вычислить одним проходом справа налево.

Для сравнений результатов времени моделирования предлагаемого симулятора AlexSim с симулятором ModelSim V5.5 рассматривались несколько эталонных схем из ISCAS-85 и ISCAS-89, создавались так называемые testbench-и для их моделирования. Последние также пишутся на языке Verilog, но уже в Behavioural форме и предназначены для определения входов схемы в каждом такте симулятора и записи выходов в файл. Входные данные тоже записываются в файл, чтобы в дальнейшем можно было бы ввести их в программу и проверить идентичность результатов. Все входные значения при моделировании брались из генератора случайных чисел.

Количество наборов во всех случаях равно 1.000.000.

Таблица 3.

Сравнение работы симулятора AlexSim с симулятором ModelSim V5.5.

Имя схемы	Кол. Входов/ выходов	Кол. элемен- тов	Время симуляции программы ModelSim (сек.)	Время симуляции программы AlexSim (компиляция/ время симуляции) (сек.)	Отноше- ние времен симуля- ции
c1355	41 / 32	546	60	4 / 2.7	22,22
c1908	33 / 25	880	115	3 / 2.9	39,66
c2670	233 / 140	1193	154	28 / 16.1	9,57
c3540	50 / 22	1669	204	5 / 5.4	37,78
c5315	178 / 123	2307	393	22 / 15.4	25,52
c6288	32 / 32	2416	417	9 / 8.5	49,06
c7552	207 / 08	3512	967	26 / 22.6	42,79
s196a	14 / 14	547	57	2 / 1.9	30,00
s196b	14 / 14	547	56	3 / 1.8	31,11
s238a	14 / 14	526	57	2 / 1.9	30,00

Продолжение таблицы 3.

Имя схемы	Кол. Входов/ выходов	Кол. элемен- тов	Время симуляции программы ModelSim (сек.)	Время симуляции программы AlexSim (компиляция/ время симуляции) (сек.)	Отноше- ние времен симуля- ции
s238	14 / 14	526	56	3 / 1.9	29,47
s13207	62 / 152	8589	2106	15 / 16.0	131,63
s1423	17 / 5	731	105	2 / 3.1	33,87
s1488	8 / 19	659	30	2 / 1.5	20,00
s27	4 / 1	13	7	2 / 2.0	3,50
s298	3 / 6	133	20	2 / 0.4	50,00
s344	9 / 11	175	24	2 / 0.7	34,29
s349	9 / 11	176	25	2 / 0.7	35,71
s382	3 / 6	179	27	2 / 0.5	54,00
s386	7 / 7	165	17	1 / 0.7	24,29
s400	3 / 6	185	27	2 / 0.7	38,57
s420	18 / 1	234	30	2 / 1.4	21,43
s444	3 / 6	202	27	1 / 0.6	45,00
s510	19 / 7	217	13	2 / 1.4	9,29
s526a	3 / 6	215	26	1 / 0.6	43,33
s526	3 / 6	214	26	2 / 0.6	43,33
s5378	35 / 49	2958	288	8 / 7.5	38,40
s713	35 / 23	412	46	2 / 2.3	20,00
s820	18 / 19	294	22	2 / 1.5	14,67
s832	18 / 19	292	22	2 / 2.0	11,00
s838	34 / 1	478	58	2 / 2.6	22,31
s9234	36 / 39	5808	639	10 / 8.9	71,80
s953	16 / 23	424	53	2 / 1.7	31,18

Рассматривается также функциональное моделирование цифровых схем на уровне логических элементов, определенных на множестве слов в алфавите $\{0,1,x,z\}$, где x – неопределенный сигнал, а z – постоянный, но неизвестный сигнал, который обусловлен высоким сопротивлением,

обрывом или коротким замыканием внутри схемы. Для этого случая принимаются следующие таблицы истинности для примитивных элементов:

and	0	1	x	z	or	0	1	x	z	xor	0	1	x	z
0	0	0	0	0	0	0	1	x	x	0	0	1	x	x
1	0	1	x	x	1	1	1	1	1	1	1	0	x	x
x	0	x	x	x	x	x	1	x	x	x	x	x	x	x
z	0	x	x	x	z	x	1	x	x	z	x	x	x	x

nan	0	1	x	z	no	0	1	x	z	xno	0	1	x	z
d					г					г				
0	1	1	1	1	0	1	0	x	x	0	1	0	x	x
1	1	0	x	x	1	0	0	0	0	1	0	1	x	x
x	1	x	x	x	x	x	0	x	x	x	x	x	x	x
z	1	x	x	x	z	x	0	x	x	z	x	x	x	x

buf	
0	0
1	1
x	x
z	x

not	
0	1
1	0
x	x
z	x

Пятая глава посвящена некоторым общим теоретическим вопросам, касающихся свойств булевых функций. В частности, рассматриваются некоторые свойства пороговых функций, минимальных дизъюнктивных нормальных форм этих функций, применению булевых дифференциалов и активностей переменных булевых функций для выделения активной части схемы, одному стандартному представлению булевой функции и т.д.

Как известно, булева функция $f(x_1, x_2, \dots, x_n)$ называется пороговой, если существуют целые числа $\xi_1, \xi_2, \dots, \xi_n$ и w такие, что для любого набора $(\alpha_1, \alpha_2, \dots, \alpha_n)$ из нулей и единиц $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1$ тогда и только тогда, когда

$$\sum_{i=1}^n \xi_i \alpha_i \geq w.$$

Числа $\xi_1, \xi_2, \dots, \xi_n$ называются весами аргументов указанной функции, а w - порогом.

Функция $f(x_1, x_2, \dots, x_n)$ называется положительно-пороговой, если ее параметры $\xi_1, \xi_2, \dots, \xi_n$ и w неотрицательны. Известно, что задача распознавания пороговости функции алгоритмически весьма сложна. Однако существуют легко распознаваемые пороговые функции, с помощью которых можно построить достаточно широкие классы пороговых функций.

Заметим, что из совокупности целых чисел $\xi_1, \xi_2, \dots, \xi_n, w$ однозначно определяется соответствующая пороговая функция $f(x_1, x_2, \dots, x_n)$, в то же время эта же функция может быть определена и другим набором чисел, например, набором

$$k\xi_1, k\xi_2, \dots, k\xi_n, kw,$$

где k - любое натуральное число.

Теорема 5.1. Если функция $f(x_1, x_2, \dots, x_n)$ является пороговой, то для любого набора $(\sigma, \sigma_1, \sigma_2, \dots, \sigma_n)$ из нулей и единиц функция $f^\sigma(x_1^{\sigma_1}, x_2^{\sigma_2}, \dots, x_n^{\sigma_n})$ также является пороговой.

Следствие 1. Для любой пороговой функции $f(x_1, x_2, \dots, x_n)$ существует набор $(\sigma, \sigma_1, \sigma_2, \dots, \sigma_n)$ из нулей и единиц такой, что функция

$$f_1(x_1, x_2, \dots, x_n) = f^\sigma(x_1^{\sigma_1}, x_2^{\sigma_2}, \dots, x_n^{\sigma_n})$$

является положительно-пороговой.

Множество всех пороговых функций можно разбить на классы эквивалентности, включая в один и тот же класс те функции $f_1(x_1, x_2, \dots, x_n)$ и $f_2(x_1, x_2, \dots, x_n)$, для которых существует набор из нулей и единиц $(\sigma, \sigma_1, \sigma_2, \dots, \sigma_n)$ такой, что

$$f_2(x_1, x_2, \dots, x_n) = f_1^\sigma(x_1^{\sigma_1}, x_2^{\sigma_2}, \dots, x_n^{\sigma_n}).$$

Отсюда следует, что класс эквивалентности, порождаемый пороговой функцией $f_1(x_1, x_2, \dots, x_n)$, является

$$\{f^\sigma(x_1^{\sigma_1}, \dots, x_n^{\sigma_n}) \mid \sigma, \sigma_i \in \{0, 1\}, i = 1, \dots, n\}.$$

Следовательно, изучение пороговых функций можно свести к изучению множества представителей классов эквивалентности. В качестве таких представителей могут быть, например, положительно-пороговые функции.

Теорема 5.2. Положительно-пороговая функция является монотонной.

Определение 5.1. Функция $f(x_1, x_2, \dots, x_n)$ называется обобщенно-монотонной, если существует такая монотонная функция $\varphi(x_1, x_2, \dots, x_n)$ и набор $\tilde{\sigma} = (\sigma, \sigma_1, \dots, \sigma_n)$, что

$$f(x_1, x_2, \dots, x_n) = \varphi^\sigma(x_1^{\sigma_1}, x_2^{\sigma_2}, \dots, x_n^{\sigma_n}).$$

Следствие 2. Любая пороговая функция является обобщенно-монотонной.

Заметим, что не все монотонные функции пороговые. Контрпримером является монотонная функция $x_1 x_2 \vee x_3 x_4$.

Теорема 5.3. Для любой положительно-пороговой функции $f(x_1, x_2, \dots, x_n)$ с весами аргументов $\xi_1, \xi_2, \dots, \xi_n$ и порогом w существует равная ей положительно-пороговая функция $F(x_{i_1}, \dots, x_{i_k})$, где $k = \xi_1 + \dots + \xi_n$, $x_{i_j} \in \{x_1, \dots, x_n\}$, $j = 1, \dots, k$, с весами 1 и порогом w .

Заметим, что любую положительно-пороговую функцию $f(x_1, x_2, \dots, x_n)$ с весами аргументов 1 и порогом w можно представить дизъюнктивной нормальной формой, которая состоит из всех C_n^w элементарных конъюнкций вида $x_{i_1} x_{i_2} \dots x_{i_w}$. Эта дизъюнктивная нормальная форма осуществляет простое "голосование" по принципу преодоления порога w . Например, если для пороговой функции $f(x_1, x_2, x_3, x_4)$ $\xi_1 = \xi_2 = \xi_3 = \xi_4 = 1$ и $w = 3$, то

$$f(x_1, x_2, x_3, x_4) = x_1 x_2 x_3 \vee x_1 x_2 x_4 \vee x_1 x_3 x_4 \vee x_2 x_3 x_4.$$

Схема технической реализации пороговой функции $f(x_1, x_2, \dots, x_n)$ пороговым элементом с n входами и одним выходом показана на рис.8.

Блоки ξ_i ($i = \overline{1, n}$) осуществляют умножение ξ_i на x_i , блок

Σ образует взвешенную сумму значений x_i , а блок w сравнивает взвешенную сумму с порогом w .

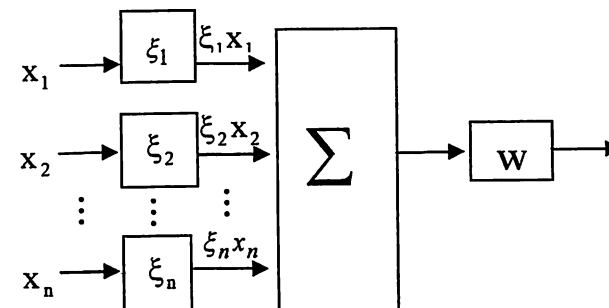


Рис.8.

Автоматическое изменение весов ξ_i и порога w порогового элемента обеспечивает значительное расширение возможностей этого элемента. Так, из схемы, указанной на рисунке 8, можно получить пороговые элементы, реализующие все пороговые функции от n переменных.

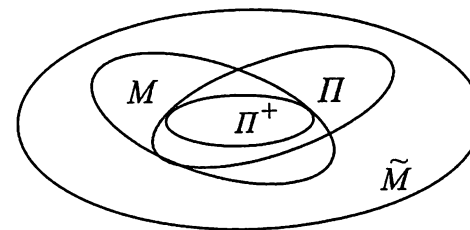


Рис. 9

Соотношение классов монотонных, обобщенно-монотонных, пороговых, положительно-пороговых функций изображено на рис.9, где M – класс монотонных функций, \tilde{M} – класс обобщенно-монотонных функций, Π – класс пороговых функций, а Π^+ – класс положительно-обобщенных функций.

В теории булевых функций важной задачей является задача построения минимальной дизъюнктивной нормальной формы функции. Как известно, сокращенная дизъюнктивная нормальная форма монотонной функции не содержит отрицаний и является ее единственной минимальной дизъюнктивной нормальной формой.

Теорема 5.4. Сокращенная дизъюнктивная нормальная форма

обобщенно-монотонной функции является единственной минимальной дизъюнктивной нормальной формой.

Следствие 3. Сокращенная дизъюнктивная нормальная форма любой пороговой функции одновременно является минимальной дизъюнктивной нормальной формой.

Определение 5.2. Набор чисел $(\omega_1, \omega_2, \dots, \omega_n)$ называется допустимым, если существует булева функция $f(x_1, x_2, \dots, x_n)$ такая, что $\omega_i = \omega_i^f$ для $i = \overline{1, n}$.

$\tilde{\sigma}$ -набором назовем набор чисел $\tilde{\sigma} = (\sigma_0, \sigma_1, \dots, \sigma_n)$, где $\sigma_i \in \{0, 1\}$, $i = \overline{0, n}$. $\tilde{\sigma}$ -процедурой назовем применение $\tilde{\sigma}$ -набора к некоторой функции $f(x_1, x_2, \dots, x_n)$, в результате чего получается новая функция $f^{\sigma_0}(x_1^{\sigma_1}, x_2^{\sigma_2}, \dots, x_n^{\sigma_n})$. Множество всех функций $f^{\sigma_0}(x_1^{\sigma_1}, x_2^{\sigma_2}, \dots, x_n^{\sigma_n})$, где $\sigma_i \in \{0, 1\}$, $i = \overline{0, n}$, называется $\tilde{\sigma}$ -семейством функции $f(x_1, x_2, \dots, x_n)$. Если две функции принадлежат одному $\tilde{\sigma}$ -семейству, то они называются родственными.

Для булевых функций существуют некоторые стандартные представления, которые стали классическими: совершенная дизъюнктивная нормальная форма, совершенная конъюнктивная нормальная форма и полином Жегалкина. Введением понятия производной для булевых функций и установлением их некоторых важных свойств появляются возможности получения новых стандартных форм представления этих функций.

Приведена новая стандартная форма, где производная булевой функции по своим аргументам занимает центральное место.

Теорема 5.5. Любая булева функция $f(x_1, \dots, x_n)$ может быть представлена в следующем виде:

$$f(x_1, \dots, x_n) = x_1 \frac{\partial f(x_1, \dots, x_n)}{\partial x_1} \oplus x_2 \frac{\partial f(0, x_2, \dots, x_n)}{\partial x_2} \oplus \dots \oplus x_i \frac{\partial f(0, \dots, 0, x_i, \dots, x_n)}{\partial x_i} \oplus x_n \frac{\partial f(0, \dots, 0, x_n)}{\partial x_n} \oplus f(0, \dots, 0, 0). \quad (2)$$

Представление функции в виде (2) в некотором смысле является обобщением полинома Жегалкина. В частности, если производные в нем являются постоянными, то оно является представлением линейной функции. Если же они являются конъюнкциям, не содержащими отрицания переменных, то (2) попросту является полиномом Жегалкина функции.

Наконец, если указанные производные представлены в любом другом виде, то заменой их своими полиномами Жегалкина, получится полином Жегалкина функции.

Таким образом, представлен новый способ построения полинома Жегалкина функции.

Булева функция характеризуется активностями совокупностей аргументов. Наиболее важным являются активности отдельных аргументов, которые играют решающую роль при анализе надежности комбинационных схем.

Приведем некоторые свойства допустимых наборов.

1. Если набор $(\omega_1, \dots, \omega_n)$ допустимый, то набор $(\omega_i, \dots, \omega_i)$, полученный из приведенного набора перемещением координат, также является допустимым.
2. Если наборы $(\omega_1, \dots, \omega_k)$ и $(\omega_{k+1}, \dots, \omega_n)$ допустимы, то набор $(\omega_1, \dots, \omega_k, \omega_{k+1}, \dots, \omega_n)$ также является допустимым.
3. Если наборы $(\omega'_1, \dots, \omega'_{n-1})$ и $(\omega''_1, \dots, \omega''_{n-1})$ допустимы, а их соответствующими функциями являются $\varphi(x_1, \dots, x_{n-1})$ и $\psi(x_1, \dots, x_{n-1})$, то набор $(\omega_1, \dots, \omega_{n-1}, \omega_n)$ также является допустимым, где

$$\omega_i = \frac{\omega'_i + \omega''_i}{2}, \quad i = 1, 2, \dots, n-1, \quad \omega_n = \|\varphi \oplus \psi\|.$$

Определение 5.3. Допустимый набор $(\omega_1, \dots, \omega_n)$ назовем составным, если его можно разбить на такие две части $(\omega_1, \dots, \omega_k)$ и $(\omega_{k+1}, \dots, \omega_n)$, что каждая часть также является допустимым набором. В противном случае он называется простым.

Приведено необходимое и достаточное условие для того, чтобы набор являлся составным.

Теорема 5.6. Допустимый набор $(\omega_1, \dots, \omega_n)$ является составным тогда и только тогда, когда соответствующую ему функцию $f(x_1, \dots, x_n)$ можно представить в виде суммы по mod2 двух функций, которые не имеют общих существенных аргументов, т.е. $f(x_1, \dots, x_n) = \varphi(x_1, \dots, x_k) \oplus \psi(x_{k+1}, \dots, x_n)$.

ЗАКЛЮЧЕНИЕ

В диссертационной работе получены следующие основные результаты:

- 1) Исследованы возможности языка Alex описания цифровых схем, выявлен ряд специфических особенностей с точки зрения расширения области его применения в автоматизации проектирования цифровых схем. Уточнены понятия цифровой комбинационной схемы с одним выходом, схемы со многими выходами, схемы из логических и запоминающих элементов, определены записи описания соответствующих схем на языке Alex. Доказаны теоремы о записи, устоявшиеся необходимые и достаточные условия для того, чтобы данная последовательность символов являлась записью некоторой схемы. Проверка условий теоремы о записи позволяет осуществить эффективный синтаксический контроль правильности описания схемы.
- 2) Язык Alex обобщен для описания схем высокого уровня, элементами которых могут быть подсхемы любого уровня. Конструктивно решены задачи «укрупнения» элементов комбинационной схемы, разбиения схемы на части, удовлетворяющие заданным техническим и технологическим ограничениям, выделения и удаления максимальной подсхемы. Разработан алгоритм перехода из одного уровня детализации описания схемы к другому (высокому или низкому), тем самым обеспечивая тождественные преобразования описания схем.
- 3) Предложена алгебра эквивалентных преобразований логических схем и эффективные алгоритмы их реализации: последовательное соединение, частично-последовательное соединение, параллельное соединение, выделение максимальной подсхемы, выделение подсхемы.
- 4) Предложен новый подход к модульно-ориентированному проектированию систем на чипах. Введено понятие динамической активности элементов схемы, увеличение которой увеличивает динамическую потребляемую мощность и, как следствие, уменьшает надежность элементов. Разработаны алгоритмы временной синхронизации цифровых схем, уменьшения потребляемой мощности цифровых схем, заменой выделенных пассивных подсхем на константы.
- 5) Разработан принцип наилучшего приближения булевых функций функциями меньшим числом аргументов и эффективный алгоритм осуществления этой процедуры. На основании этого результата введено понятие активной части цифровой схемы с одним выходом, разработан алгоритм выделения активной части схемы с одним выходом. Эффективность алгоритма показана для эталонных схем ISCAS-85. Разработан принцип существенного сокращения времени моделирования

и верификации при автоматизированном проектировании СБИС, заменой верификации двух схем верификацией активных частей этих схем.

- 6) Разработан и успешно реализован на тестовых эталонных схемах из ISCAS-85, ISCAS-89 и ITC'99 быстрый алгоритм функционального моделирования. Результаты сравнения работы программ моделирования AlexSim и ModelSim показали бесспорное преимущество программы AlexSim. Алгоритм быстрого моделирования расширен для алфавита $\{0,1,x,z\}$. Разработан алгоритм минимального представления булевых функций от четырех переменных с целью оптимизации описания цифровых схем в автоматизации проектирования схем. Апробация предложенных методов на эталонных схемах серий ISCAS-85, ISCAS-89 и ITC'99 показали состоятельность такого подхода.
- 7) Предложено новое стандартное представление булевой функции с использованием булевых дифференциалов.

Основные результаты диссертации опубликованы в следующих работах:

1. Бозоян Ш. Е., Егиазарян В.С. Некоторые процедуры над логическими схемами и их реализация на языке Alex // «Исследовано в России», -2003г. Т.6. –С. 817-824.
2. Бозоян Ш.Е., Егиазарян В.С. Новый подход к модульно-ориентированному проектированию систем на чипах // «Исследовано в России», -2003г. Т.6.. –С. 1386-1395.
3. Бозоян Ш.Е., Егиазарян В.С. Некоторые функциональные свойства пороговых функций // Вестник ГИУА. серия: «Моделирование; Оптимизация; Управление». -2004г. -Вып. 7, -Т. 2. –С. 21-25.
4. Bозоян Sh. Yeghiazaryan V. To the module-orientated approach of chip designing. 2-nd IEEE International Conference on Circuits and Systems for Communication (ICCSC'04). Moscow, Russia, 30 June-2 July, -2004г. 66 pdf.
5. Бозоян Ш.Е., Егиазарян В.С., Погосян С.П. Быстрое моделирование схем из логических и запоминающих элементов// Вестник ГИУА, серия: «Моделирование; Оптимизация; Управление», -2006г. -Вып. 9. -Т.1. –С.34-42.
6. Бозоян Ш.Е., Егиазарян В.С., Погосян С.П. Новый метод вычисления динамических активностей элементов цифровых схем // Государственный инженерный университет Армении (Политехник), Сборник материалов годичной научной конференции. -2006г. –Т. 1. –С. 18-21.
7. Бозоян Ш.Е., Егиазарян В.С., Погосян С.П. Метод быстрого моделирования цифровых схем // Математика в высшей школе. -2006г. Т. 2. -№3. -С. 24-28.
8. Бозоян Ш.Е., Егиазарян В.С. О наилучшем приближении булевых функций // Вестник ГИУА, серия: «Моделирование; Оптимизация;

- Управление», -2006г. -Вып. 9. -Т. 2. -С. 34-42.
9. Бозоян Ш.Е., Егиазарян В.С. Язык Alex описания схем // «ПРОГРАММИРОВАНИЕ». -Москва. -2007г. -№ 4. Т. 33. -С. 65-80.
 10. Бозоян Ш.Е., Егиазарян В.С., Погосян С.П. Быстрое моделирование цифровых схем с использованием их строчного описания на языке Alex // Труды VI Международной конференции «Идентификация систем и задачи управления». -Москва. 29 января-1 февраля 2007г. SICPRO'07. -С. 796-813. ISBN 5-201-14992-8.
 11. Бозоян Ш.Е., Егиазарян В.С. К существенному сокращению времени верификации при проектировании СБИС // Высокие технологии, фундаментальные и прикладные исследования, образование: Сборник трудов Третьей международной научно-практической конференции «Исследование, разработка и применение высоких технологий в промышленности». 14.03.07.-17.03.07. Санкт-Петербург. Россия. Под ред. А.П. Кудинова. СПб: Изд-во Политехнического ун-та. - 2007г. Т. 9. -С. 36-41.
 12. Егиазарян В.С., Погосян С.П. Классы эквивалентностей булевых функций P и NPN. Минимальное представление булевых функций от четырех переменных // «Известия НАН Армении и ГИУА». Серия технических наук. -2007г. -Т. LX. -№ 3. С. 544-552.
 13. Бозоян Ш.Е., Егиазарян В.С. Простые и составные наборы активностей аргументов булевых функций // Годичная научная конференция. Сборник научных статей. РАУ. -2006г. -С. 85-89.
 14. Бозоян Ш.Е., Егиазарян В.С. Ускорение процесса верификации цифровых схем с использованием активной части схемы // Высокие технологии, фундаментальные и прикладные исследования, образование. Т. 11: Сборник трудов Шестой международной научно-практической конференции «Исследование, разработка и применение высоких технологий в промышленности». 16-17.10.2008г. Санкт-Петербург. Россия. Под ред. А.П. Кудинова. СПб: Изд-во Политехнического ун-та. -2008г. -Т. 11. -С. 76-79.
 15. Бозоян Ш.Е., Егиазарян В.С. Активная часть цифровых схем и ее применение в области автоматизации проектирования схем // VII Международная конференция «Идентификация систем и задачи управления». SICPRO '08. -2009. -С. 1205-1209.
 16. Бозоян Ш.Е., Егиазарян В.С., Кочаров Д.А., Егиазарян Л.В., Гишян С.А. Выделение и вставка подсхемы с использованием языка описания Alex // Вестник ГИУА. серия: «Моделирование; Оптимизация; Управление». - 2008. -Вып. 11. -Т. 1. -С. 27-30.
 17. Егиазарян В.С. О минимальной дизъюнктивной нормальной форме пороговых функций // Сборник трудов научных статей годичной

- научной конференции РАУ. -2007г. -С. 122-125.
18. Егиазарян В.С. Об одном эквивалентном преобразовании цифровых схем // Вестник ГИУА, серия: «Моделирование; Оптимизация; Управление». -Ереван. -2008г. - Вып. 11. -Т. 1. -С. 35-40.
 19. Бозоян Ш.Е., Егиазарян В.С., Кочаров Д.А., Бадалян Н.Б. Применение булевых дифференциалов и активностей переменных булевых функций для выделения активной части схемы // Известия Национальной Академии Наук Армении и ГИУА. Серия технических наук. -2008г. - Т. LXI № 1. -С. 137-146.
 20. Егиазарян В.С. Новый алгоритм вычисления активностей входов цифровых схем // Вестник РАУ. Физико-Математические и естественные науки. -2008г. -№1. -С. 28-31.
 21. Бозоян Ш.Е., Егиазарян В.С. Переход из одного уровня детализации описания схемы к другому // VIII Международная конференция. «Идентификация систем и задачи управления». SICPRO'09. - 2010г. - С.1570-1575.
 22. Егиазарян В.С. Функциональное моделирование цифровых схем на уровне логических элементов, определенных на множестве слов в алфавите $\{0,1, x, z\}$ // Третья годичная научная конференция. Сборник научных статей. РАУ. -2008г. -С. 85-89.
 23. Бозоян Ш.Е., Егиазарян В.С. Модульно-ориентированное проектирование систем на чипах. Вестник РАУ // Физико-Математические и естественные науки. -2009г. -№1. -С. 37-44.
 24. Егиазарян В.С. О временной синхронизации цифровых схем // Вестник РАУ. Физико-Математические и естественные науки. -2009г. -№2. -С. 37-44.
 25. Егиазарян В.С. Разбиение схемы на части // Вестник РАУ. Физико-Математические и естественные науки. -2009. -№2. С. 93-99.
 26. Егиазарян В.С. Об одном стандартном представлении булевой функции // Высокие технологии, фундаментальные и прикладные исследования, образование. Сборник трудов Девятой международной научно-практической конференции «Исследование, разработка и применение высоких технологий в промышленности». 22-25.04.2010, Санкт-Петербург, Россия. Под ред. А.П. Кудинова, Г.Г. Матвиенко. СПб.: Изд-во Политехн. ун-та. -2010. -Т. 4. С.109-112.
 27. Айказян Д.А., Бозоян Ш.Е., Егиазарян В.С. Графический редактор проектирования древовидных цифровых схем // Четвертая годичная научная конференция, Сборник научных статей. РАУ. -2009г. -С. 86-90.
 28. Акопджанян В.А., Бозоян Ш.Е., Егиазарян В.С. Алгоритм уменьшения

потребляемой мощности цифровых схем// Четвертая годовая научная конференция. Сборник научных статей. РАУ. –Ереван. -2009г. –С. 91-96.

29. Бозоян Ш.Е., Егиазарян В.С. Расширение языка Alex для описания схем, построенных из элементов со многими выходами// Пятая годовая научная конференция. Сборник научных статей. РАУ. 2010г. С. 160-167.
30. Бозоян Ш.Е., Егиазарян В.С. Выделение и удаление максимальной подсхемы с одним выходом из цифровой схемы// Высокие технологии, фундаментальные и прикладные исследования, образование.: Сборник трудов Одиннадцатой международной научно-практической конференции “Исследование, разработка и применение высоких технологий в промышленности“. 27-29 апреля 2011 г. Санкт-Петербург. Россия. Под ред. А.П. Кудинова. СПб.: Изд-во Политехнического ун-та. -2011г. –Т. 1. –С. 89-93.

Անփոփում

Ատենախոսական աշխատանքում ստացվել են հետևյալ հիմնական արդյունքները.

1. Ուսումնասիրվել են թվային սխեմաների նկարագրման Alex լեզվի հնարավորությունները, բացահայտվել են մի շարք առանձնահատկություններ՝ թվային սխեմաների ավտոմատացված նախագծման ասպարեզում նրա կիրառությունների ընդլայնման տեսակետից: Ճշգրտվել են մեկ ելքով թվային կոմբինացիոն սխեմայի, շատ ելքերով սխեմայի, տրամաբանական և հիշող տարրերով սխեմայի գաղափարները, սահմանվել են համապատասխան սխեմաների գրառումները Alex լեզվով: Այսպես ստացվել են գրառման մասին թեորեմները, որոնք սահմանում են անհրաժեշտ և բավարար պայմաններ սիմվոլների տրված հաջորդականության որևէ սխեմայի գրառում հանդիսանալու համար: Գրառման մասին թեորեմի պայմանների ստուգմամբ իրականացվում է սխեմայի գրառման ճշտության շարահյուսական արդյունավետ հսկողություն:

2. Alex լեզուն ընդհանրացվել է բարձր մակարդակի այնպիսի սխեմաների նկարագրության համար, որոնք կառուցված են ցանկացած մակարդակի ենթասխեմա հանդիսացող տարրերով: Առաջարկվել են կառուցողական լուծումներ կոմբինացիոն սխեմայի տարրերի

«խոշորացման», սխեմայի՝ որոշակի տեխնիկական և տեխնոլոգիական սահմանափակումների բավարարող մասերի տրոհման, մաքսիմալ ենթասխեմայի անջատման և հեռացման խնդիրների համար: Մշակվել է սխեմայի նկարագրման մի մակարդակից այլ մակարդակի (ավելի բարձր կամ ցածր) անցնելու ալգորիթմը՝ դրանով իսկ ապահովելով սխեմաների նկարագրության նույնական ձևափոխություններ:

3. Առաջարկվել են տրամաբանական սխեմաների համարժեք ձևափոխությունների հանրահաշիվ և այդ ձևափոխությունների իրականացման արդյունավետ ալգորիթմներ. հաջորդական միացում, մասնակի-հաջորդական միացում, զուգահեռ միացում, մաքսիմալ ենթասխեմայի անջատում, սխեմայի անջատում:

4. Առաջարկվել է չիփերից կառուցված համակարգերի՝ մոդուլային կոդմոդալմամբ նախագծման նոր մոտեցում: Ներմուծվել է սխեմայի տարրերի դինամիկ ակտիվության գաղափարը: Դինամիկ ակտիվության աճը մեծացնում է օգտագործվող դինամիկ հզորությունը և, որպես հետևանք, փոքրացնում տարրի հուսալիությունը: Մշակվել են թվային սխեմաների ժամանակային սինխրոնացման, թվային սխեմայի պասիվ ենթասխեմաները հաստատուններով փոխարինելու ճանապարհով օգտագործած հզորության փոքրացման ալգորիթմներ:

5. Մշակվել է բուլյան ֆունկցիայի քիչ թվով արգումենտներով բուլյան ֆունկցիայով լավագույն մոտարկման սկզբունքը և այդ պրոցեդուրայի իրականացման արդյունավետ ալգորիթմ: Այս արդյունքի հիման վրա ներմուծվել է մեկ ելքով թվային սխեմայի ակտիվ մասի գաղափարը, մշակվել է մեկ ելքով սխեմայի ակտիվ մասի անջատման ալգորիթմը: Ալգորիթմի արդյունավետությունը ցուցադրվել է էտալոնային ISCAS-85 սխեմաների վրա: Մշակվել է գերմեծ ինտեգրալ սխեմայի ավտոմատացված նախագծման կարևորագույն փուլերից մեկի՝ մոդելավորման և վերիֆիկացման ժամանակի էական կրճատման սկզբունքը՝ երկու սխեմաների վերիֆիկացումը նրանց ակտիվ մասերի վերիֆիկացումով փոխարինելով:

6. Մշակվել և ISCAS-85, ISCAS-89, ITC'99 էտալոնային սխեմաների վրա հաջողությամբ իրականացվել է ֆունկցիոնալ մոդելավորման արագ ալգորիթմը: AlexSim և ModelSim մոդելավորման ծրագրերի աշխատանքի արդյունքների համեմատությունը ցուցադրել է AlexSim ծրագրի ակնառու առավելությունները: Արագ մոդելավորման

ալգորիթմն ընդլայնվել է $\{0,1,x,z\}$ այբուբենի համար: Մշակվել է չորս փոփոխականից բուլյան ֆունկցիաների մինիմալ ներկայացման ալգորիթմը՝ ավտոմատացված նախագծման ժամանակ թվային սխեմաների նկարագրության օպտիմալացման համար: Առաջարկված մեթոդների փորձարկումը ISCAS-85, ISCAS-89 և ITC'99 էտալոնային սխեմաների համար ցույց է տվել այդպիսի մոտեցման հիմնավորվածությունը:

7. Բուլյան դիֆերենցիալների օգտագործումով առաջարկվել է բուլյան ֆունկցիաների նոր ստանդարտ ներկայացում:

VLADIMIR SAHAK YEGHIAZARYAN

LANGUAGE AND ALGORITHMIC PROBLEMS AUTOMATED
DESIGN
OF DIGITAL CIRCUITS

CONCLUSION

Results achieved in the thesis are as follows:

1. The scope of the Alex language for digital scheme description is examined. A number of specific peculiarities of field expansion and their application in the projection of digital scheme description are revealed here. The notion of digital combination scheme with one output, many outputs and schemes from logical and memorable elements are also defined, besides circuit notations for the description of corresponding schemes in Alex language are identified. A theorem of circuit notation is proved establishing the necessary and satisfactory conditionals for the given sequence of symbols to be a circuit notation for a certain scheme. The verification of the conditions of the theorem allows us to make an efficient syntactic control of the correctness of the scheme description.

2. Alex language is generalized for the high level scheme circuit notation, elements of which may be sub schemes of any level. The problems of the combinative scheme element that is "integration", scheme segmentation that satisfies the given technical and technological limitations selection and removal of the maximal sub scheme are constructively solved.

An algorithm for the transition from one level of detailed scheme to another (high or low), providing identical transformation of scheme description is developed in the thesis.

3. Equivalent transformation algebra for logical scheme and efficient algorithms with its implementation is suggested: consistent combination, partially consistent combination, parallel combination, allocation of maximal sub scheme, allocation of sub scheme.

4. A new approach to module-oriented projection systems on chips is suggested, as well as a notion of schemes' elements dynamic activity is introduced, the expansion of which increases the dynamic power and decreases the safety of elements. Also an algorithm for temporal synchronization of digital schemes is developed to reduce the used power of digital schemes, for replacement of allocated passive sub scheme with constants.

5. In the thesis we've also worked out the principle of the best approximation of Boolean functions with the least number of arguments and efficient algorithm for this procedure implementation. On the basis of these results a notion of active part of a digital scheme with one output is introduced and an algorithm for the allocation of the scheme's active part with one output is developed. The efficiency of an algorithm is shown for standard schemes ISCAS-85. Consequently, we've developed a policy for the significant decrease of time simulation and verification while atomized projection of VLSI by verification replacement of two active parts of the schemes.

6. A quick algorithm for functional modeling is worked out and successfully realized on standard schemes ISCAS-85, ISCAS-89 and ITC'99. The comparison of the results of modeling programs AlexSim and ModelSim have proved the obvious advantage of AlexSim. The algorithm for fast modeling is expanded for the alphabet $\{0,1,x,z\}$. In this work an algorithm of minimal presentations of Boolean function from four variables is developed with the aim to optimize the description of digital schemes projection. The approbation of the suggested methods on the standard scheme of ISCAS-85, ISCAS-89 and ITC'99 types have shown the opulence of this approach.

7. A new standard implementation of Boolean function with the use of Boolean differentials is developed.



Տպագրված է «ԼԻՄՈՒՇ» ՍՊԸ-ի տպարանում:
Տպաքանակ՝ 50:

15.05.2014

Հայաստանի Ազգային գրադարան



NL0456336

