

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Пензенский государственный университет
архитектуры и строительства»
(ПГУАС)

В.Г. Камбург, О.А. Кувшинова, В.А. Елисеев

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И СЕТИ В ОТРАСЛИ

Рекомендовано Редсоветом университета
в качестве учебного пособия для студентов,
обучающихся по направлению 23.03.03 (190600.62)
«Эксплуатация транспортно-технологических машин
и комплексов»

Пенза 2014

УДК 004:[629.33+656] (075.8)

ББК 32.97:39.3я73

К18

Рецензент – кафедра «Математика и суперкомпьютерное моделирование» Пензенского государственного университета (зав. кафедрой доктор физико-математических наук, профессор Ю.Г. Смирнов)

Камбург В.Г.

К18 Вычислительная техника и сети в отрасли: учеб. пособие / В.Г. Камбург, О.А. Кувшинова, В.А. Елисеев. – Пенза: ПГУАС, 2014. – 104 с.

Даны общие теоретические и практические материалы по моделированию и проектированию цифровых комбинационных и последовательных вычислительных устройств и их типовых узлов.

Теория основывается на принципах моделирования на взаимно однозначных соответствиях алгебры логики и технических узлов, таких, как счетчики, регистры, сумматоры и др. Практическая часть посвящена примерам моделирования электронных схем в среде WORKBENCH.

Учебное пособие подготовлено на кафедре «Информационно-вычислительные системы» и предназначено для студентов, обучающихся по направлению 23.03.03 (190600.62) «Эксплуатация транспортно-технологических машин и комплексов» очной и заочной форм обучения.

© Пензенский государственный университет архитектуры и строительства, 2014

© Камбург В.Г., Кувшинова О.А., Елисеев В.А., 2014

ВВЕДЕНИЕ

Внедрение ЭВМ оказало большое влияние на многие области науки и техники, вызвало процесс их математизации и компьютеризации. Развитие вычислительной техники, а также сферы и методов ее использования – процессы взаимосвязанные и взаимообусловленные. С одной стороны, потребности народного хозяйства, науки и культуры стимулируют поиски учеными и конструкторами новых путей построения ЭВМ, а с другой – появление электронных вычислительных машин, систем и устройств с большими функциональными возможностями, с существенно улучшенными показателями по производительности, стоимости, габаритным размерам, надежности и т.п. создает предпосылки для непрерывного расширения областей и развития форм применения ЭВМ.

Первоначально сравнительно узкая сфера использования ЭВМ, главным образом для научных и технических расчетов, в короткий срок существенно расширилась и охватила почти все области науки, техники, планирования и управления технологическими процессами, все области человеческой деятельности, связанные с обработкой больших объемов информации.

Именно поэтому обучение студентов новым специальностям, сочетающим как профессиональные знания в области компьютерных технологий, проектирования и эксплуатации больших информационных систем, так и специальные знания в конкретной области науки и техники, например строительстве, является на современном этапе развития общества весьма актуальным и перспективным.

Настоящее учебное пособие охватывает вопросы моделирования цифровых вычислительных средств на дискретных элементах и состоит из 5 разделов и приложения.

Первый раздел посвящен краткой истории развития вычислительной техники

Во втором и третьем разделах подробно рассматриваются общие вопросы моделирования элементов, узлов, схем в их взаимно однозначном соответствии с модельными образами в виде логических выражений, преобразований булевой алгебры и навыков оптимизации реальных схем и сетей.

В четвертом разделе дан обзор основных технических элементов узлов и сетей.

В пятом разделе приводится пример практического использования моделирования электронных схем.

1. ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

1.1. Этапы развития вычислительной техники

Люди всегда испытывали потребность в счете. Для этого они использовали пальцы рук, камешки, которые складывали в кучки или располагали в ряд. Число предметов фиксировалось с помощью черточек, которые проводились по земле, с помощью зарубок на палках и узелков, которые завязывались на веревке.

С увеличением количества подлежащих подсчету предметов, развитием наук и ремесел появилась необходимость в проведении простейших вычислений. Самым древним инструментом, известным в различных странах, являются счеты (в Древнем Риме они назывались *calculi*). Они позволяют производить простейшие вычисления над большими числами. Счеты оказались настолько удачным инструментом, что дожили с древних времен почти до наших дней.

Историки сходятся во мнении, что их возраст составляет несколько тысяч лет, а их родиной могут быть и Древний Китай, и Древний Египет, и Древняя Греция.

С развитием точных наук появилась необходимость в проведении большого количества точных вычислений.

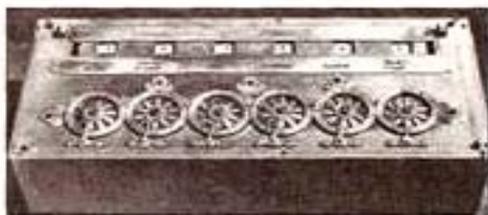


Рис. 1. Счетная машина Паскаля

делало полный оборот, в действие автоматически приводилось второе колесико (десятки); когда и оно достигало цифры 9, начинало вращаться третье колесико и т.д. Машина Паскаля могла только складывать и вычитать.

В 1642 г. французский математик Блез Паскаль сконструировал первую механическую счетную машину, известную как суммирующая машина Паскаля (рис.1). Эта машина представляла собой комбинацию взаимосвязанных колесиков и приводов. На колесиках были нанесены цифры от 0 до 9. Когда первое колесико (единицы)



Рис. 2. Счетная машина Лейбница

Когда первое колесико (единицы) делало полный оборот, в действие автоматически приводилось второе колесико (десятки); когда и оно достигало цифры 9, начинало вращаться третье колесико и т.д. Машина Паскаля могла только складывать и вычитать.

В 1694 г. немецкий математик Готфрид Вильгельм фон Лейбниц сконструировал более совершенную счетную машину (рис.2). Он был убежден, что его изобретение найдет широкое применение не только в науке, но и в быту. В отличие от

машины Паскаля Лейбниц использовал цилиндры, а не колесики и приводы. На цилиндры были нанесены цифры. Каждый цилиндр имел девять рядов выступов или зубцов. При этом первый ряд содержал

1 выступ, второй – 2 и так вплоть до девятого ряда, который

содержал 9 выступов. Цилиндры были подвижными и приводились в определенное положение оператором. Конструкция машины Лейбница была более совершенной: она была способна выполнять не только сложение и вычитание, но и умножение, деление и даже извлечение квадратного корня.

Однако уже в конце XIX в. с изобретением электромагнитного реле появились первые электромеханические счетные устройства. В 1887 г. Герман Голлерит (США) изобрел электромеханический табулятор с вводом чисел с помощью перфокарт. На идею использовать перфокарты его натолкнула пробивка компостером проездных билетов на железнодорожном транспорте. Разработанная им 80-колонная перфокарта не претерпела существенных изменений и в качестве носителя информации применялась в первых трех поколениях компьютеров. Табуляторы Голлерита использовались во время 1-й переписи населения в России в 1897 г. Сам изобретатель тогда специально приезжал в Санкт-Петербург. С этого времени электромеханические табуляторы и другие подобные им устройства стали широко применяться в бухгалтерском учете.

В начале XIX в. Чарльз Бэббидж сформулировал основные положения, которые должны лежать в основе конструкции вычислительной машины принципиально нового типа. В такой машине, по его мнению, должны быть «склад» для хранения цифровой информации, специальное устройство, осуществляющее операции над числами, взятыми со «склада». Бэббидж называл такое устройство «мельницей». Другое устройство служит для управления последовательностью выполнения операций, передачей чисел со «склада» на «мельницу» и обратно; наконец, в машине должно быть устройство для ввода исходных данных и вывода результатов вычислений. Эта машина так никогда и не была построена –

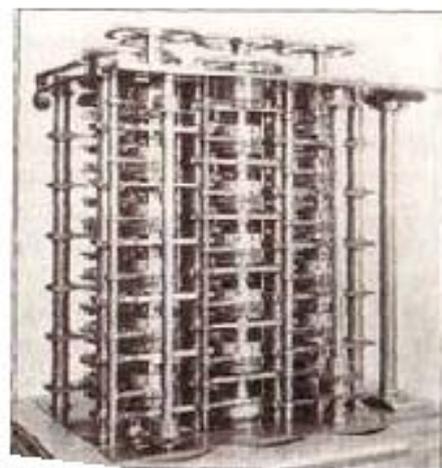


Рис. 3. Машина Бэббиджа

существовали лишь ее модели, но принципы, положенные в ее основу, были позже реализованы в цифровых ЭВМ.

Научные идеи Бэббиджа увлекли дочь известного английского поэта лорда Байрона – графиню Аду Августу Лавлейс. Она заложила первые фундаментальные идеи о взаимодействии различных блоков вычислительной машины и последовательности решения на ней задач. Поэтому Аду Лавлейс по праву считают первым в мире программистом. Многими понятиями, введенными Адой Лавлейс в описания первых в мире программ, широко пользуются современные программисты.

Началом новой эры развития вычислительной техники на базе электромеханических реле стал 1934 г. Американская фирма IBM (International Business Machines) начала выпуск алфавитно-цифровых табуляторов, способных выполнять операции умножения. В середине 30-х годов XX в. на основе табуляторов создается прообраз первой локальной вычислительной сети. В Питсбурге (США) в универсаме была установлена система, состоящая из 250 терминалов, соединенных телефонными линиями с 20 табуляторами и 15 пишущими машинками для расчетов с покупателями. В 1934–1936 гг. немецкий инженер Конрад Цузе пришел к идее создания универсальной вычислительной машины с программным управлением и хранением информации в запоминающем устройстве. Он сконструировал машину «Z-3» – это была первая программно-управляемая вычислительная машина – прообраз современных ЭВМ.

В начале 40-х годов XX в. в лабораториях IBM совместно с учеными Гарвардского университета была начата разработка одной из самых мощных электромеханических вычислительных машин. Она получила название MARK-1, содержала 760 тыс. компонентов и весила 5 т.

Переход к следующему этапу развития средств вычислительной техники и технологии программирования был бы невозможен без основополагающих научных исследований в области передачи и обработки информации. Развитие теории информации связано прежде всего с именем Клода Шеннона. Отцом кибернетики по праву считается Норберт Винер, а создателем теории автоматов является Джон фон Нейман.

Концепция кибернетики родилась из синтеза многих научных направлений: во-первых, как общий подход к описанию и анализу действий живых организмов и вычислительных машин или иных автоматов; во-вторых, из аналогий между поведением сообществ живых организмов и человеческого общества и возможностью их описания с помощью общей теории управления; и, наконец, из синтеза теории передачи информации и статистической физики, который

привел к важнейшему открытию, связывающему количество информации и отрицательную энтропию в системе. Сам термин «кибернетика» происходит от греческого слова, означающего «кормчий»; в современном смысле он впервые был применен Н.Винером в 1947 г. Книга Н.Винера, в которой он сформулировал основные принципы кибернетики, называется «Кибернетика об управлении и связи в животном и машине» [8].

Клод Шеннон – американский инженер и математик, человек, которого называют отцом современной теории информации. Он доказал, что работу переключателей и реле в электрических схемах можно представить посредством алгебры, изобретенной в середине XIX в. английским математиком Джорджем Булем. С тех пор булева алгебра стала основой для анализа логической структуры систем любого уровня сложности.

В 1946 г. блестящий американский математик венгерского происхождения Джон фон Нейман сформулировал основную концепцию хранения команд компьютера в его собственной внутренней памяти, что послужило огромным толчком к развитию электронно-вычислительной техники.

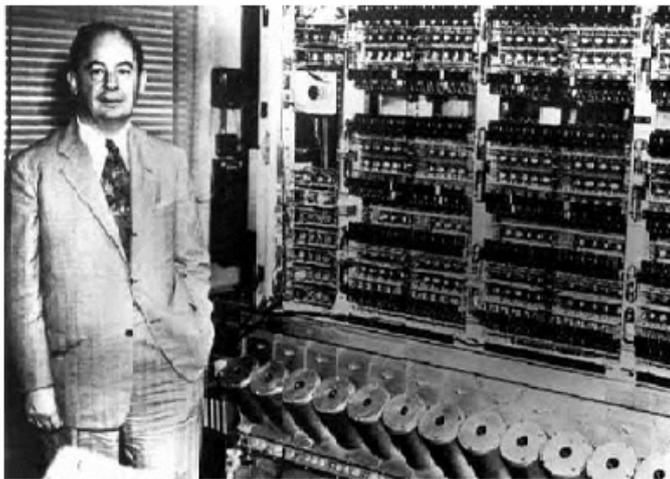


Рис. 4. Машины, построенные на принципах фон Неймана

В основу построения подавляющего большинства ЭВМ положены следующие общие принципы, сформулированные в 1945 году американским ученым венгерского происхождения ДЖОНОМ фон НЕЙМАНОМ.

Принцип двоичного кодирования:

1. Принцип программного управления

Из него следует, что программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности.

2. Принцип однородности памяти

Программы и данные хранятся в одной и той же памяти. Поэтому ЭВМ не различает, что хранится в данной ячейке памяти – число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

3. Принцип адресности

Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти так, чтобы к запомненным в них значениям можно было бы впоследствии обращаться или менять их в процессе выполнения программы с использованием присвоенных имен.

Согласно фон Нейману, ЭВМ состоит из следующих основных блоков:

- Устройства ввода/вывода информации.
- Память компьютера.
- Процессор, состоящий из устройства управления (УУ) и арифметико-логического устройства (АЛУ).

Можно и по-другому сформулировать принципы Джона фон Неймана:

1) принцип программного управления (программа состоит из набора команд, которые выполняются процессором друг за другом в определенной последовательности);

2) принцип однородности памяти (программы и данные хранятся в одной и той же памяти);

3) принцип адресности (ОП состоит из пронумерованных ячеек, и процессору в любой момент времени доступна любая ячейка).



Рис. 5. Принцип фон Неймана

Архитектура ЭВМ, построенной на принципах фон Неймана. Сплошные линии со стрелками указывают направление потоков информации, пунктирные – управляющих сигналов от процессора к остальным узлам ЭВМ.

Разработанные фон Нейманом основы архитектуры вычислительных устройств оказались настолько фундаментальными, что получили в литературе название “фон-неймановской архитектуры”. Подавляющее большинство вычислительных машин на сегодняшний день – фон-неймановские машины. Исключение составляют лишь отдельные разновидности систем для параллельных вычислений, в которых отсутствует счетчик команд, не реализована классическая концепция переменной и имеются другие существенные принципиальные отличия от классической модели (примерами могут служить потоковая и редукционная вычислительные машины).

По-видимому, значительное отклонение от фон-неймановской архитектуры произойдет в результате развития идеи машин пятого поколения, в основе обработки информации в которых лежат не вычисления, а логические выводы.

1947 г. — Экертом и Мочли начата разработка первой электронной серийной машины UNIVAC (Universal Automatic Computer). Первый образец машины (UNIVAC-1) был построен для бюро переписи США и пущен в эксплуатацию весной 1951 г.

1949 г. — английским исследователем Морнсом Уилксом построен первый компьютер, в котором были воплощены принципы фон Неймана.

1951 г. — Дж. Форрестер опубликовал статью о применении магнитных сердечников для хранения цифровой информации. В машине «Whirlwind-1» впервые была применена память на магнитных сердечниках.

1952 г. — фирма IBM выпустила свой первый промышленный электронный компьютер IBM 701, который представлял собой синхронную ЭВМ параллельного действия, содержащую 4000 электронных ламп и 12 000 диодов.

1956 г. — фирмой IBM были разработаны плавающие магнитные головки на воздушной подушке. Изобретение их позволило создать новый тип памяти — дисковые запоминающие устройства (ЗУ), значимость которых была в полной мере оценена в последующие десятилетия развития вычислительной техники.

1957 г. — группа под руководством Д. Бэкуса завершила работу над первым языком программирования высокого уровня, получившим

название ФОРТРАН. Язык, реализованный впервые на ЭВМ IBM 704, способствовал расширению сферы применения компьютеров.

1974 г. — несколько фирм объявили о создании на основе микропроцессора Intel-8008 персонального компьютера — устройства, выполняющего те же функции, что и большой компьютер, но рассчитанного на одного пользователя.

1975 г. — появился первый коммерчески распространяемый персональный компьютер Альтаир-8800 на основе микропроцессора Intel-8080. Этот компьютер имел оперативную память всего 256 байт, клавиатура и экран отсутствовали.

Конец 1975 г. — Пол Аллен и Билл Гейтс (будущие основатели фирмы Microsoft) создали для компьютера «Альтаир» интерпретатор языка Basic, позволивший пользователям просто общаться с компьютером и легко писать для него программы.

1990-е гг. — появление глобальных компьютерных сетей массового пользования.

2000-е гг. — интеграция ЭВМ и бытовой техники, встраиваемые компьютеры, развитие сетевых вычислений.

1.2. Поколение ЭВМ

Компьютерная грамотность предполагает наличие представления о пяти поколениях ЭВМ. Когда говорят о поколениях, то в первую очередь имеют в виду исторический портрет электронно-вычислительных машин (ЭВМ).

Всю историю развития электронно-вычислительной техники принято делить на поколения. Смены поколений чаще всего были связаны со сменой элементной базы ЭВМ, с прогрессом электронной техники. Это всегда приводило к росту быстродействия и увеличению объема памяти. Кроме того, как правило, происходили изменения в архитектуре ЭВМ, расширялся круг задач, решаемых на ЭВМ, менялся способ взаимодействия между пользователем и компьютером.



Рис. 6. Электровакуумные лампы

ЭВМ первого поколения были ламповыми машинами 50-х годов. Их элементная база — это электровакуумные лампы (рис. 6). Эти ЭВМ были весьма громоздкими сооружениями, содержащими в себе тысячи ламп, занимавшими иногда сотни квадратных метров территории и потреблявшими электроэнергию в сотни киловатт.

Например, одна из первых ЭВМ – ENIAC – представляла собой огромный по объему агрегат длиной более 30 метров, содержала 18 тысяч электровакуумных ламп и потребляла около 150 киловатт электроэнергии. Для ввода программ и данных применялись перфоленты и перфокарты. Не было монитора, клавиатуры и мышки. Использовались эти машины главным образом для инженерных и научных расчетов, не связанных с переработкой больших объемов данных. В 1949 году в США был создан первый полупроводниковый прибор, заменяющий электронную лампу. Он получил название транзистора.

В 1960-х годах транзисторы (рис.7) послужили элементной базой для ЭВМ второго поколения. Машины стали компактнее, надежнее, менее энергоемкими. Возросло быстродействие и объем внутренней памяти. Большое развитие получили устройства внешней (магнитной) памяти: магнитные барабаны, накопители на магнитных лентах.



Рис. 7. Транзисторы

В этот период развиваются языки программирования высокого уровня: ФОРТРАН, АЛГОЛ, КОБОЛ. Составление программы перестало зависеть от конкретной модели машины, сделалось проще, понятнее, доступнее.

В 1959 г. был изобретен метод, позволивший создавать на одной пластине и транзисторы, и все необходимые соединения между ними. Полученные таким образом схемы стали называться интегральными схемами или чипами. Изобретение интегральных схем послужило основой для дальнейшей миниатюризации компьютеров.

В дальнейшем количество транзисторов, которое удавалось разместить на единицу площади интегральной схемы, увеличивалось приблизительно вдвое каждый год.

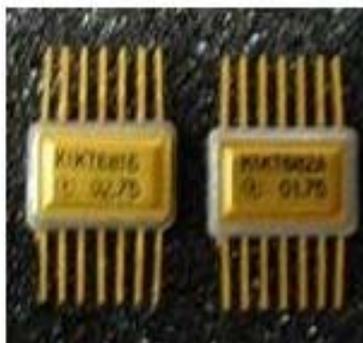


Рис. 8. Интегральная схема

Третье поколение ЭВМ создавалось на новой элементной базе – **интегральных схемах (ИС)**. ЭВМ третьего поколения начали производиться во второй половине 1960-х годов, когда американская фирма IBM приступила к выпуску системы машин IBM-360. Немного позднее появились машины серии IBM-370.

В Советском Союзе в 1970-х годах начался выпуск машин серии ЕС ЭВМ (Единая система ЭВМ) по образцу IBM 360/370. Скорость работы наиболее мощных моделей ЭВМ достигла уже нескольких миллионов операций в секунду.

ду. На машинах третьего поколения появился новый тип внешних запоминающих устройств – магнитные диски.

Успехи в развитии электроники привели к созданию **больших интегральных схем (БИС)**, где в одном кристалле размещалось несколько десятков тысяч электрических элементов.

В 1971 году американская фирма Intel объявила о создании микропроцессора. Это событие стало революционным в электронике.



Рис. 9. Микропроцессор

Микропроцессор – это миниатюрный мозг, работающий по программе, заложенной в его память. Соединив микропроцессор с устройствами ввода-вывода и внешней памяти, получили новый тип компьютера – микроЭВМ.

МикроЭВМ относится к машинам **четвертого поколения**. Широкое распространение получили персональные компьютеры (ПК). Их появление связано с именами двух американских специалистов: Стива Джобса и Стива Возняка. В 1976 году на свет появился их первый серийный ПК Apple-1, а в 1977 году – Apple-2.

Однако с 1980 года «законодателем мод» на рынке ПК становится американская фирма IBM. Ее архитектура стала фактически международным стандартом на профессиональные ПК. Машины этой серии получили название IBM PC (Personal Computer). Появление ПК по своему значению для общественного развития сопоставимо с появлением книгопечатания.

С развитием этого типа машин сформировалось понятие «информационные технологии», без которых невозможно обойтись в большинстве областей деятельности человека. Возникла новая дисциплина – информатика.

ЭВМ пятого поколения будут основаны на принципиально новой элементной базе. Основным их качеством должен быть высокий интеллектуальный уровень, в частности распознавание речи, образов. Это требует перехода от традиционной фон-неймановской архитектуры компьютера к архитектурам, учитывающим требования задач создания искусственного интеллекта. Таким образом, на данный момент создано четыре поколения ЭВМ:

- 1-е поколение: 1946 г. – создание машины ЭНИАК на электронных лампах.
- 2-е поколение: 1960-е годы. ЭВМ построены на транзисторах.
- 3-е поколение: 1970-е годы. ЭВМ построены на интегральных микросхемах (ИС).

- 4-е поколение: начало создаваться с 1971 г. с изобретением микропроцессора (МП). Построены на основе больших интегральных схем (БИС) и сверх БИС (СБИС).

Пятое поколение ЭВМ строится по принципу человеческого мозга, управляется голосом. Соответственно, предполагается применение принципиально новых технологий. Огромные усилия были приложены Японией в разработке компьютера 5-го поколения с искусственным интеллектом, но пока они не увенчались успехом. Фирма IBM тоже не намерена сдавать свои позиции мирового лидера, например, Японии. Мировая гонка за создание компьютера пятого поколения началась еще в 1981 году. С тех пор еще никто не достиг финиша.

1.3. Принципы организации вычислительных систем

Стремление удовлетворить требования разнообразных областей и форм применения электронной вычислительной техники, повысить производительность и расширить логические возможности **ЭВМ**, увеличить надежность их функционирования, облегчить контакты человека с **ЭВМ** при подготовке программ и в процессе решения задач, повысить обслуживаемость машин привело в ряде случаев к созданию таких объектов вычислительной техники, которые из-за сложности входящего в них оборудования, тесной логической взаимосвязи аппаратурных и программных средств при реализации сложных процессов функционирования, множества возможных конфигураций, территориальной рассредоточенности оборудования не укладываются в наше представление о понятии *машина*. В таких случаях вместо термина *вычислительная машина* пользуются термином *вычислительная система (ВС)*.

1.3.1. Классификация вычислительных систем

Одним из важнейших путей повышения производительности вычислительных машин и систем, их эффективности и надежности является использование различных форм реализации принципа параллелизма в функционировании вычислительного оборудования. Поэтому в основу классификации ВС следует положить в первую очередь реализуемую системой форму параллелизма. По режиму работы ВС делятся на *однопрограммные и мультипрограммные* (рис.10). В однопрограммной системе в памяти машины присутствует только одна рабочая программа (или часть ее), которая, начав выполняться, завершается до конца. При этом, даже если допускается совмещение во времени операций ввода-вывода с обработкой данных, возможен

простой оборудования (например, машина не может продолжать работу, пока не будут введены новые данные).

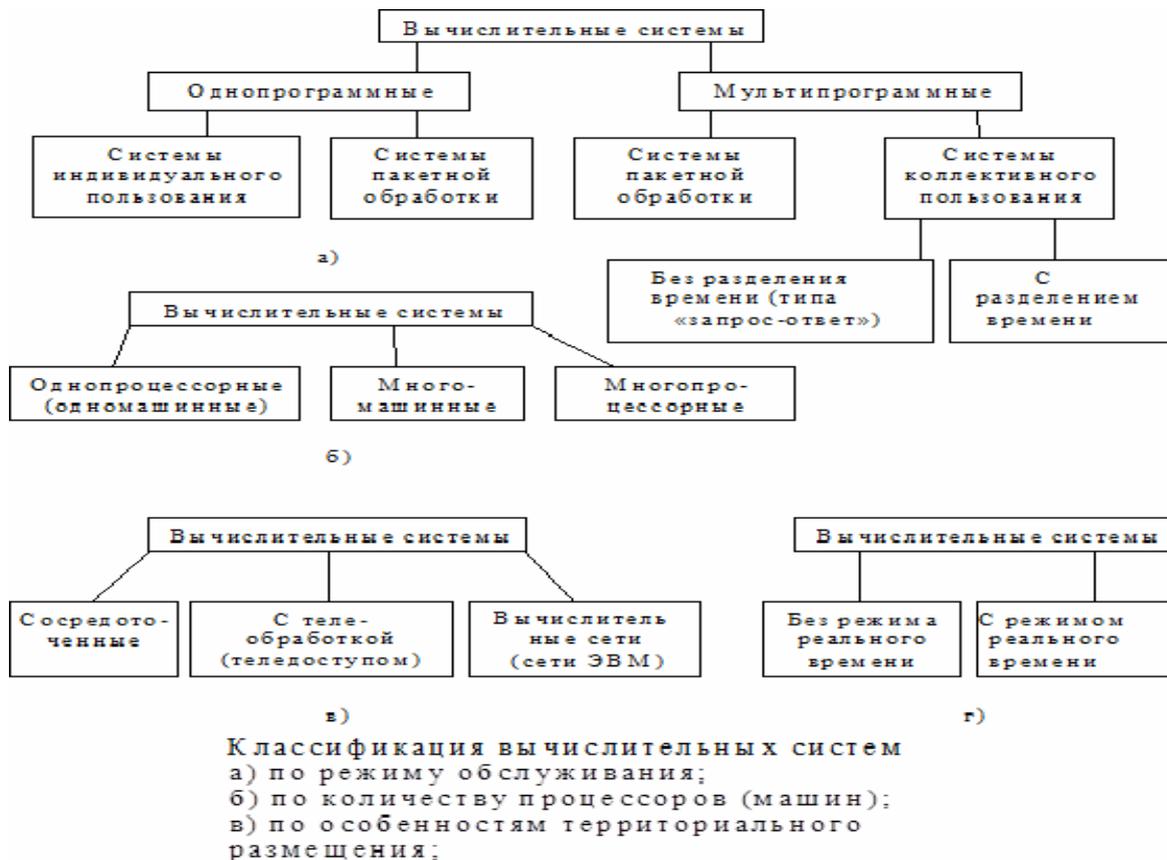


Рис. 10. Классификация ВС

Стремление повысить эффективность использования вычислительного оборудования привело к разработке мультипрограммных машин и систем, которые могут одновременно выполнять несколько программ или несколько частей одной и той же программы.

Когда говорится об одновременности выполнения программ, то подразумевается, что процессор после выполнения части одной программы может перейти к выполнению другой программы, не закончив ее, перейти к третьей и т.д., сохраняя возможность вернуться позднее к неоконченным программам и продолжить их выполнение. При этом моменты и очередность в переключении программ должны быть выбраны так, чтобы повысить общую эффективность вычислительной системы, хотя время, в течение которого решается каждая отдельная задача, по сравнению с однопрограммным режимом может даже увеличиться.

Мультипрограммный режим реализуется и по отношению к частям одной программы, что приводит к сокращению времени ее выполнения

по сравнению со случаем, когда режим мультипрограммирования не используется.

Классификация систем по режиму обслуживания. Процесс решения задачи может рассматриваться как обслуживание пользователя ВС. Рассмотрим основные режимы обслуживания.

Режим индивидуального пользования. Машина предоставляется полностью в распоряжение пользователя, по крайней мере, на время решения его задачи. Пользователь имеет непосредственный доступ к машине и может вводить информацию в оперативную память машины (или выводить из нее), используя пульт управления или устройства ввода-вывода.

Режим индивидуального пользования удобен пользователю, но плохо используется вычислительное оборудование из-за простоев, когда пользователь, получив некоторый промежуточный результат (например, при отладке программы), обдумывает, что он будет делать дальше. Этот режим применялся в ЭВМ первого поколения, а в настоящее время возродился как форма использования персональных микроЭВМ.

Режим пакетной обработки. В этом режиме пользователи не имеют непосредственного доступа к ВС. Подготовленные ими программы передаются персоналу, обслуживающему систему, и затем накапливаются во внешней памяти (на магнитных дисках, лентах и т.п.). Система последовательно либо по заранее составленному расписанию выполняет накопленный пакет программ. Пакетная обработка может производиться в однопрограммном или мультипрограммном режиме.

Мультипрограммная пакетная обработка обеспечивает высокую степень загрузки вычислительного оборудования, но при этом из-за отсутствия непосредственной связи между системой и пользователем производительность и эффективность труда самих пользователей снижаются по сравнению с индивидуальным обслуживанием. Это противоречие преодолевается в настоящее время путем создания систем коллективного пользования, содержащих высокопроизводительные ЭВМ, или, наоборот, путем применения персональных ЭВМ умеренной производительности в режиме индивидуального пользования.

Режим коллективного пользования, или мультидоступа, — форма обслуживания, при которой возможен одновременный доступ нескольких независимых пользователей к вычислительным ресурсам мощной **ВС**. Каждому пользователю предоставляется терминал, с помощью которого он устанавливает связь с системой коллективного пользования (**ВСКП**).

В наиболее простых ВСКП обработка всех запросов занимает примерно одно и то же время (системы типа «запрос – ответ»).

Обеспечение более тесного взаимодействия пользователей с вычислительными средствами в системе коллективного пользования, в которой запросы сильно разнятся по времени их обработки, требует в первую очередь сокращения времени ожидания пользователем результата выполнения коротких программ (коротких запросов), для чего применяют различные методы квантования времени, уделяемого процессором выполнению отдельных программ. Системы коллективного пользования с *квантованным обслуживанием* называются *системами с разделением времени*.

По количеству процессоров (машин) в **ВС**, определяющему возможность параллельной обработки программ, вычислительные системы делятся на *однопроцессорные (одномашинные)*, *многомашинные и многопроцессорные*.

По особенностям территориального размещения и организации взаимодействия частей системы различают следующие типы **ВС** (рис.10).

Сосредоточенные ВС. В них весь комплекс оборудования, включая терминалы пользователей, сосредоточен практически в одном месте и связь между отдельными машинами и устройствами комплекса обеспечивается, причем без существенных запаздываний, стандартными для системы внутренними интерфейсами (без использования передачи данных по каналам связи).

Вычислительные системы с телеобработкой (или теледоступом). В этих **ВС** отдельные источники и приемники информации, включая терминалы пользователей, расположены на таком значительном расстоянии от вычислительных средств, что связь их с центральными средствами **ВС** осуществляется путем передачи данных по каналам связи.

Вычислительные сети (сети ЭВМ). Они представляют собой территориально рассредоточенную многомашинную систему, состоящую из взаимодействующих **ЭВМ**, связанных между собой каналами передачи данных.

По особенностям функционирования во времени различают **ВС**, работающие не в масштабе реального времени, и *ВС реального времени*. Последние должны работать в темпе с процессом, информация о котором автоматически поступает в **ВС** и обрабатывается. Результаты обработки информации должны получаться столь быстро, чтобы можно было ими воспользоваться для воздействия на сам процесс.

Вычислительные системы различаются по назначению следующим образом:

- информационно-справочные;
- сбора и обработки данных в **АСУ**, в частности информационно-планирующие;
- управления технологическими процессами в реальном времени;
- автоматизации обработки данных при сложных экспериментах;
- автоматизации проектирования и др.

Многие виды применения **ВС**, например автоматизация проектирования, требуют, чтобы поиск решений задач осуществлялся в режиме взаимодействия человека с машиной.

Проблемная ориентация **ВС** достигается соответствующей конфигурацией аппаратурных средств и укомплектованием их соответствующим проблемно-ориентированным программным обеспечением.

1.4. Понятие об архитектуре ЭВМ

Основой любой микроЭВМ является микропроцессор (МП), который осуществляет необходимое преобразование информации внутри МП, управляет остальными компонентами микроЭВМ и обеспечивает передачу информации между компонентами микроЭВМ, а также между микроЭВМ и внешней средой.

Архитектура МП – это его логическая организация, определяемая возможностями МП по аппаратной или программной реализации функций, необходимых для построения микроЭВМ. Понятие архитектуры МП отражает его структуру, способы обращения ко всем доступным для пользователя элементам структуры, способы представления и форматы данных, набор операций, выполняемых МП, способы адресации данных, участвующих в операциях, форматы управляющих слов, поступающих в МП извне, характеристики и назначение вырабатываемых МП управляющих сигналов, реакцию МП на внешние сигналы.

Проектирование микроЭВМ обычно начинается с тщательного анализа предъявляемых к ней требований. Эти требования могут быть представлены в виде граф схемы алгоритма работы микроЭВМ, определяющей последовательность всех действий, необходимых для выполнения заданных функций. Кроме граф схемы необходимо учитывать временные ограничения на выполнение всех (или основных) действий. Эти ограничения оказывают существенное влияние на способы построения проектируемой микроЭВМ и тесно связаны с архитектурой МП. Зная архитектуру МП и всей микроЭВМ, можно составить

последовательность команд, называемую программой, которая обеспечит выполнение заданной функции.

Основой МП или устройства с программируемой логикой является арифметическо-логическое устройство (АЛУ), предназначенное для обработки информации.

Другим важным узлом МП является набор регистров общего назначения (РОН), часто именуемый СОЗУ. В РОН хранятся информационные слова, подлежащие обработке в АЛУ, результаты обработки информации в АЛУ и управляющие слова. Обращение к РОН – адресное. Эти регистры допускают считывание и запись информации, поэтому содержат входную и выходную шины, адресную шину и управляющий вход, сигнал на котором задает режим работы: запись, хранение или чтение информации. Типовая структура обрабатывающей части МП приведена на рис.11. Данная структура обладает очень широкими возможностями. Содержимое любого РОН может быть передано на буферный регистр (БР) и на регистр сдвига (Рг.сдв). АЛУ может выполнять логические и арифметические операции над содержимым обоих регистров, а результат может быть записан в любой из РОН.

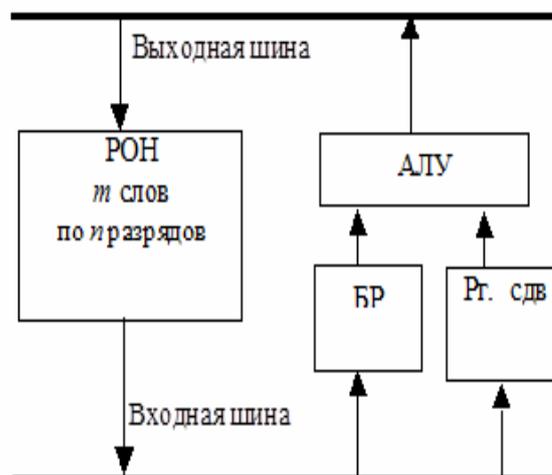


Рис. 11. Структура МП

При подаче соответствующих управляющих сигналов в этой системе, например, возможны:

- передача данных из одного РОН в другой путем пересылки выбранного из первого РОН слова транзитом через БР и АЛУ во второй РОН;
- увеличение или уменьшение на единицу содержимого любого РОН путем изменения в АЛУ выбранного из РОН значения на единицу и засылки полученного результата в тот же регистр;

- сдвиг содержимого любого РОН путем передачи выбранного числа в RгСдв, сдвига этого числа и записи через АЛУ в тот же РОН.

Тактовые сигналы могут поступать от тактового генератора, причем максимально возможная частота, а значит, и время выполнения операций будут определяться задержкой сигналов на различных компонентах схемы.

При выполнении различных действий в такой системе необходимо выполнять два требования:

- 1) обеспечение анализа логических условий и ветвление процесса;
- 2) сигнал переноса из АЛУ и выходной бит регистра сдвига должны быть сохранены, поскольку они могут понадобиться при выполнении следующей операции.

2. ОСНОВЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

2.1. Понятие информационной системы

Система – это структура, состоящая из совокупности объектов и связей между ними, предназначенная для реализации заданной совокупности функций.

Информационная система – это структура из специализированных и универсальных блоков, предназначенная для сбора, хранения, обработки и выдачи информации различного характера.

При проектировании информационных систем можно выделить три аспекта: алгоритмический, программный и технический.

В данном пособии рассматриваются основные принципы построения **технических элементов информационных систем**.

В информационной технике можно выделить следующие основные компоненты:

- средства получения, сбора, ввода информации (датчики, терминалы, читающие автоматы и т.д.);
- средства хранения информации (машинная память, базы и банки данных и т.д.);
- средства передачи информации (средства связи и передачи данных);
- средства обработки и преобразования информации (процессоры, многопроцессорные системы, интерфейсы и т.д.);
- средства представления, отображения и вывода информации (дисплеи, индикаторы, табло, графопостроители и т.д.) (рис.12).

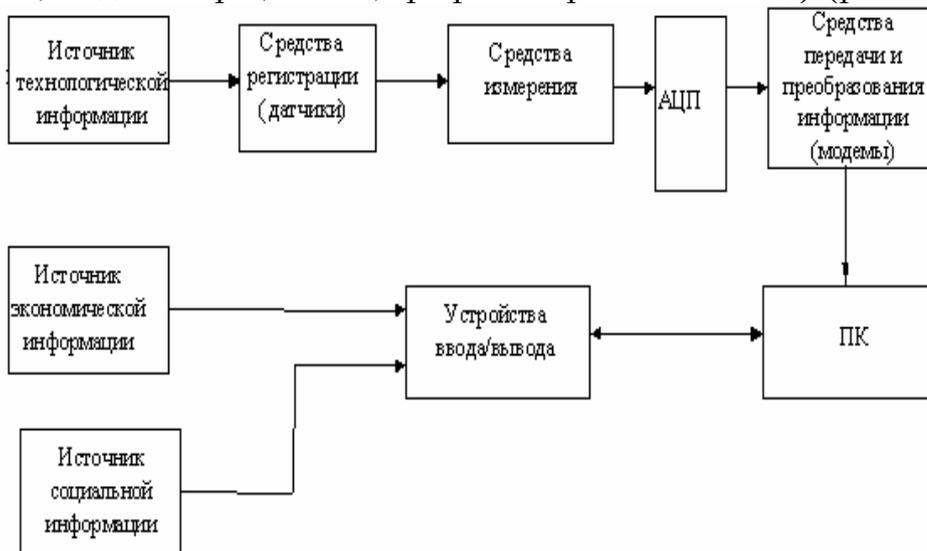


Рис. 12. Информационная система

По функциональным признакам все технические средства можно разделить на элементы, узлы (устройства) и системы.

Элемент – простейшая схема, предназначенная для одной функциональной операции с сигналом как носителем информации (получение, преобразование, хранение, усиление, передача сигнала).

Устройство (узел) – совокупность элементов, предназначенных для выполнения над совокупностью сигналов (кодом) таких функций, как запоминание кода, его преобразование, расшифровка и т.д. Устройством называют и техническое средство, состоящее из ряда узлов, например, запоминающее устройство, арифметическое устройство, устройство ввода-вывода и т.д.

Система – совокупность устройств, осуществляющих весь комплекс обработки информации, например ЭВМ, ИПС, БД и т.д.

Технические средства информационных систем могут быть построены на элементах, работающих на основе различных физических принципов (электронных, магнитных, пневматических, оптических, оптоэлектронных, химических и т.д.). В настоящее время широкое распространение получили полупроводниковые, магнитные и оптоэлектронные элементы.

2.2. Основные элементы цифровых устройств

При всем многообразии задач, решаемых цифровыми устройствами, в частности ЭВМ, функционирование их происходит в двоичной системе счисления, оперирующей только с двумя цифрами: нуль (0) и единица (1), а математическим аппаратом, на основе которого проектируются и реализуются цифровые устройства, является алгебра логики (булева алгебра).

Предметом рассмотрения алгебры логики являются высказывания (утверждения), которые могут быть либо истинными (1), либо ложными (0).

Простые высказывания, объединенные логическими операциями (связями), образуют сложное высказывание. Если простые высказывания обозначить *пропозиционными буквами или пропозиционными переменными* A, B, C, \dots (далее будем их называть *буквами или переменными*), а сложные – буквой F , то логические операции можно записать математически с помощью следующих основных логических операций.

Логическое умножение (операция И), или конъюнкция (обозначается $F = A \cap B$ или $A \wedge B$ или $F = A * B$ или $F = AB$ или $F = A \times B$). Сложное высказывание F истинно лишь тогда, когда истинны все (в

данном случае два) простые высказывания. Таблица истинности для операции конъюнкции двух переменных A и B и графическое изображение элемента И представлены на рис.13.

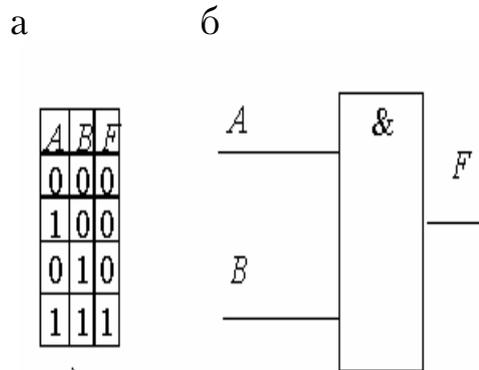


Рис. 13. Логический элемент И (конъюнкция):
а – таблица истинности; б – графическое обозначение

Логическое сложение (операция ИЛИ), или дизъюнкция, обозначаемая в виде $F = A \cup B$ или $F = A \vee B$ или $F = A + B$. Сложное высказывание истинно, если истинно хотя бы одно из простых высказываний или оба высказывания. Таблица истинности для операции дизъюнкции для двух переменных A и B и графическое изображение элемента ИЛИ приведены на рис.14. Когда хотя бы на одном входе схемы ИЛИ будет единица, на ее выходе также будет единица. Знак «1» на схеме – от устаревшего обозначения дизъюнкции как « \geq !» (т.е. значение дизъюнкции равно единице, если сумма значений операндов больше или равна 1).

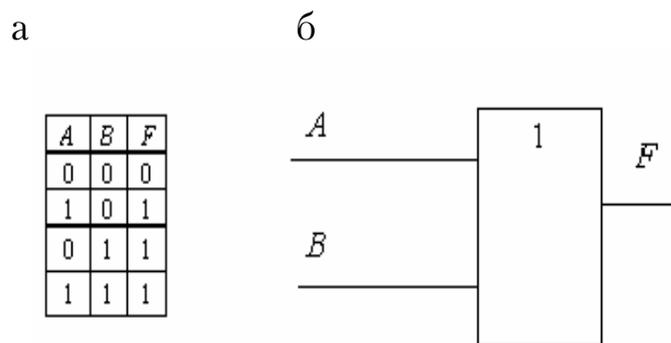


Рис. 14. Логический элемент ИЛИ (дизъюнкция):
а – таблица истинности; б – графическое обозначение

Логическое отрицание (операция НЕ), или инверсия (обозначается $F = \bar{A}$, читается «не A »). Эта операция означает, что сложное высказывание F истинно, если простое ложно, и наоборот. Таблица истинности и графическое изображение элемента НЕ приведены на рис. 15.

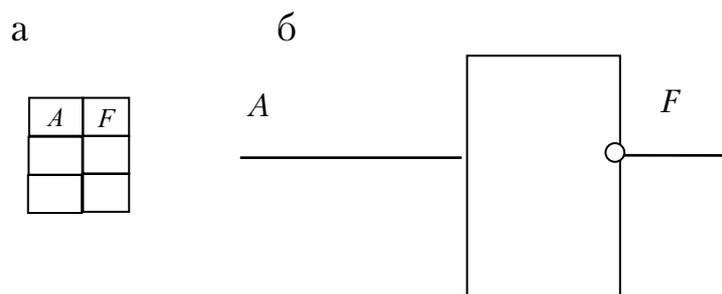


Рис.15. Логический элемент НЕ (отрицание или инверсия):
а – таблица истинности; б – графическое обозначение

Набор элементов И, ИЛИ и НЕ называют основным базисом, или *основной функционально полной системой элементов*. С помощью этих элементов можно создать схему, осуществляющую любую сколь угодно сложную логическую операцию.

В интегральной схемотехнике широко используют элементы других базисов, полученных из основного базиса:

ИЛИ – НЕ (стрелка Пирса, обозначаемая $A \downarrow B$);

И – НЕ (штрих Шеффера, обозначаемый $A \uparrow B$).

Каждый из этих элементов представляет собой функционально полную систему элементов.

Таким образом, информация в цифровых устройствах представляется в двоичной форме в виде совокупности двоичных разрядов, каждый из которых может иметь логическое состояние «1» или «0». Поэтому единицей измерения информации принят 1 бит (1 двоичный разряд). Двоичное число представляет собой совокупность нескольких двоичных разрядов. В информатике принята также единица измерения информации 1 байт = 8 бит (8 двоичных разрядов).

Наиболее распространенная форма представления информации в цифровых устройствах – позиционный двоичный код. В таком кодовом слове каждый двоичный разряд имеет свой вес (значение), который зависит от положения этого разряда в коде. Вес каждого разряда определяется числом, кратным числу 2 ($2^0, 2^1, 2^2$ и т.д.). Так, в 8-разрядном коде вес младшего разряда равен 1 (2^0), а вес старшего разряда составляет 128 (2^7). В общем случае на вход цифрового устройства поступает множество двоичных переменных $X (X_1, X_2, \dots, X_n)$, а с выхода снимается множество двоичных переменных $Y (Y_1, Y_2, \dots, Y_m)$. При этом устройство осуществляет определенную связь (логическую функцию) между входными и выходными переменными.

В последовательностных устройствах значения Y определяются значениями X , как наблюдаемыми в течение рассматриваемого такта, так и существовавшими в ряде предыдущих тактов. Для этого в

последовательностных устройствах кроме логических элементов должны быть и запоминающие.

Булевой (логической или переключательной) функцией называют двоичную переменную, значения которой зависят от значений других двоичных переменных.

Булеву функцию можно задать тремя способами: содержательно (путем словесного описания), таблично и алгебраически.

Логическое проектирование (синтез) цифрового устройства состоит из двух этапов – абстрактного и структурного синтеза.

Задача, которую должно решать устройство, обычно задается содержательно. На этапе абстрактного синтеза осуществляют переход от содержательного описания задачи к формализованному в виде таблиц или формул булевых функций. Затем составленные булевы функции минимизируются. На этапе структурного синтеза минимизированную булеву функцию реализуют, т.е. выбирают тип элементов и создают схему их электрических соединений между собой так, чтобы устройство решало поставленную задачу.

2.3. Элементы булевой алгебры (алгебра логики)

Основным математическим аппаратом, используемым для анализа и синтеза цифровых вычислительных устройств, является булева алгебра, названная по имени Джорджа Буля, разработавшего ее для изучения законов мышления [8].

Булева алгебра определяет в терминах конечного множества K две двоичные операции – логическое сложение («+», \vee , ИЛИ) и логическое умножение («*», \times , \wedge , И): $A+B = A \vee B = (A \text{ или } B)$, а также $A * B = A \times B = A \wedge B = AB = A \& B = (A \text{ и } B)$. Для группирования выражений используются круглые скобки, в отсутствие которых операция логического умножения выполняется раньше операции логического сложения (подобно арифметическим действиям).

Основные постулаты булевой алгебры:

1. Для всех $X, Y \in K$ а) $X + Y \in K$; б) $X \times Y \in K$.
2. а) для всех $X \in K$ существует элемент «0» такой, что $X + 0 = X$,
б) для всех $X \in K$ существует элемент «1» такой, что $X \times 1 = X$.
3. Коммутативность: для всех $X, Y \in K$
а) $X + Y = Y + X$; б) $X \times Y = Y \times X$.
4. Дистрибутивность: для всех $X, Y, Z \in K$
а) $X + Y \times Z = (X + Y) \times (X + Z)$; б) $Y \times (Y + Z) = X \times Y + X \times Z$.

5. Для каждого $X \in K$ существует элемент $\bar{X} \in K$ (дополнение к X), такой, что а) $X + \bar{X} = 1$; б) $X \times \bar{X} = 0$.

6. Существуют по крайней мере два элемента $X, Y \in K$ такие, что $X \neq Y$.

Свойства булевой алгебры. Некоторые свойства булевой алгебры могут быть получены непосредственно с помощью приведенных выше постулатов. Основное свойство булевой алгебры – свойство дуальности. Это свойство означает, что каждый постулат пары постулатов может быть получен из другого постулата этой же пары взаимной заменой операций ИЛИ и И, а также элементов 0 и 1.

Свойство 1. (Идемпотентность). Для всех $X \in K$

а) $X + X = X$; б) $X \times X = X$.

Свойство 2. Для каждого $X \in K$

а) $X + 1 = 1$; б) $X \times 0 = 0$.

Свойство 3. (Поглощение). Для всех $X, Y \in K$

а) $X + X \times Y = X$; б) $X \times (X + Y) = X$.

Свойство 4. Для всех $X, Y \in K$, если $X \times Y = Y$ и $X + Y = Y$, то $X = Y$.

Свойство 5. (Ассоциативность). Для всех $X, Y, Z \in K$

а) $X + (Y + Z) = (X + Y) + Z$; б) $X \times (Y \times Z) = (X \times Y) \times Z$.

Свойство 6. (Законы Де Моргана). Для всех $X, Y \in K$

а) $\overline{X + Y} = \bar{X} \times \bar{Y}$; б) $\overline{X \times Y} = \bar{X} + \bar{Y}$.

Докажем, например, пункт «а» свойства 6. Если $F = \overline{\bar{X} + \bar{Y}}$, то $\bar{F} = X + Y$.

Используем принцип дуальности, т.е. заменяем все величины их отрицаниями, а операцию логического сложения – операцией логического умножения. Получим: $\bar{F} = \bar{X} \times \bar{Y}$.

Следствие 1. (Склеивание). Для логического сложения:

| Таблица истинности | | | | |
|--------------------|---|-----|--------|---|
| A | B | A&B | не A&B | F |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |

$$(A \& B) \vee (\bar{A} \& B) = B$$

Следствие 2. (Склеивание). Для логического умножения:

| Таблица истинности | | | | | |
|--------------------|---|-----|------|--------|---|
| A | B | A∨B | не A | не A∨B | F |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

$$(A \vee B) \& (\bar{A} \vee B) = B$$

**2.4. Дизъюнктивная нормальная форма (ДНФ),
совершенная дизъюнктивная нормальная форма (СДНФ),
конъюнктивная нормальная форма (КНФ),
совершенная конъюнктивная нормальная форма (СКНФ)**

Простой конъюнкцией называется конъюнкция одной или нескольких переменных, при этом каждая переменная встречается не более одного раза (либо сама, либо ее отрицание). Например, $x\bar{y}z$ является простой конъюнкцией.

Дизъюнктивной нормальной формой (ДНФ) называется дизъюнкция простых конъюнкций. Например, выражение $x\bar{y} \vee yz$ является ДНФ.

Совершенной дизъюнктивной нормальной формой (СДНФ) называется такая дизъюнктивная нормальная форма, у которой в каждую конъюнкцию входят все переменные данного списка (либо сами, либо их отрицания), причем в одном и том же порядке. Например, выражение $x \vee yz$ является ДНФ, но не СДНФ. Выражение $x\bar{y}z \vee x\bar{y}\bar{z} \vee x\bar{y}z$ является СДНФ.

Аналогичные определения (с заменой конъюнкции на дизъюнкцию и наоборот) верны для КНФ и СКНФ. Приведем точные формулировки.

Простой дизъюнкцией называется дизъюнкция одной или нескольких переменных, при этом каждая переменная входит не более одного раза (либо сама, либо ее отрицание). Например, выражение $x \vee \bar{y} \vee z$ – простая дизъюнкция.

Конъюнктивной нормальной формой (КНФ) называется конъюнкция простых дизъюнкций (например выражение $(x \vee \bar{y} \vee z)(x \vee z)(y \vee z)$ – КНФ).

Совершенной конъюнктивной нормальной формой (СКНФ) называется такая КНФ, у которой в каждую простую дизъюнкцию входят все переменные данного списка (либо сами, либо их отрицания), причем в одинаковом порядке. Например, выражение $(x \vee y \vee \bar{z})(x \vee z)(y \vee \bar{z})$ является СКНФ.

Приведем алгоритмы переходов от одной формы к другой. Естественно, что в конкретных случаях (при определенном творческом подходе) применение алгоритмов бывает более трудоемким, чем выполнение простых преобразований, использующих конкретный вид данной формы:

имеется ДНФ $A = xy \vee \bar{y}z \vee \bar{y}\bar{z}$, после применения к ней правила Блейка можно прийти к ДНФ, равносильной данной):

$$A = xy \vee \bar{y}z \vee \bar{y}\bar{z} \vee xz = \bar{y}z \vee \bar{y}\bar{z} \vee xz$$

г) Переход от ДНФ к СДНФ.

Если в какой-то простой конъюнкции недостает переменной, например z , вставляем в нее выражение $Z \vee \bar{Z} = 1$, после чего раскрываем скобки (при этом повторяющиеся дизъюнктивные слагаемые не пишем). Например:

$$\begin{aligned} x \vee \bar{y}z &= x(y \vee \bar{y})(z \vee \bar{z}) \vee (x \vee \bar{x})\bar{y}z = xyz \vee x\bar{y}z \vee xy\bar{z} \vee x\bar{y}\bar{z} \vee x\bar{y}z \vee x\bar{y}\bar{z} = \\ &= xyz \vee x\bar{y}z \vee xy\bar{z} \vee x\bar{y}\bar{z} \vee xyz \end{aligned}$$

(в последнем равенстве из двух одинаковых слагаемых оставлено одно), т.е. из ДНФ получили СДНФ.

д) Переход от КНФ к СКНФ.

Этот переход осуществляется способом, аналогичным предыдущему: если в простой дизъюнкции не хватает какой-то переменной, например z , то добавляем в нее выражение $Z \cdot \bar{Z} = 0$ (это не меняет самой дизъюнкции), после чего раскрываем скобки с использованием распределительного закона:

$$(x \vee y)(x \vee \bar{y} \vee \bar{z}) = (x \vee y \vee z\bar{z})(x \vee y \vee \bar{z}) = (x \vee y \vee z)(x \vee y \vee \bar{z})(x \vee \bar{y} \vee \bar{z}).$$

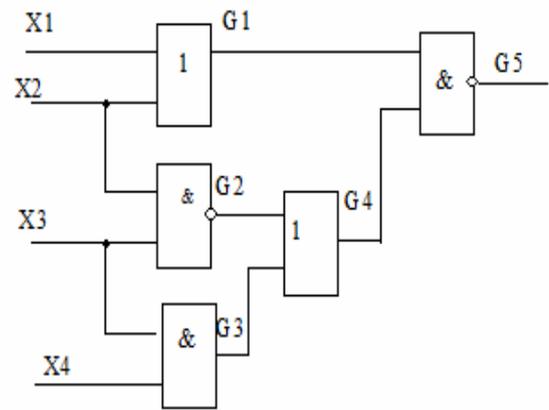
Таким образом, из КНФ получена СКНФ.

Заметим, что минимальную, или сокращенную, КНФ обычно получают из соответствующей ДНФ.

3. АНАЛИЗ И СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ

3.1. Анализ комбинационных схем

Задача анализа комбинационных схем состоит в нахождении функции, реализованной схемой, а задача синтеза, наоборот, в построении схемы, реализующей данную функцию. Функция может быть представлена либо в форме таблицы истинности, связывающей ее выходные величины с каждой возможной комбинацией входных сигналов, либо в виде булевого выражения. Процедуру анализа рассмотрим на следующем примере:



Комбинационная схема

По данной схеме легко записать: $G1 = X1 + X2$ $G2 = \overline{X2}X3$
 $G3 = X3X4$ $G4 = G2 + G3$ $G5 = \overline{G1}G4$

Преобразуем полученное выражение и, подставив одно в другое, получим:

$$G2 = \overline{X2} + \overline{X3} \quad G4 = \overline{X2} + \overline{X3} + X3X4$$

$$G5 = \overline{G1} + \overline{G4} = \overline{X1X2} + X2X3(\overline{X3X4}) = \overline{X1X2} + X2X3(\overline{X3X4}) = \overline{X1X2} +$$

$$+ X2X3(\overline{X3} + \overline{X4}) = \overline{X1X2} + X2X3\overline{X3} + X2X3\overline{X4} = \overline{X1X2} + X2X3\overline{X4}$$

3.2. Синтез комбинационных схем

Задача синтеза комбинационных цепей несколько сложнее. При этом предположим, что каноническая функция задана в виде таблицы истинности. Если количество входных переменных равно N , то количество возможных комбинаций равно 2^N . Каждая комбинация рассматривается как точка в n -кубе (n -мерное пространство). В этой точке функция f может принимать значение, равное либо 1 (1-точка), либо 0 (0-точка), либо быть неопределенной (безразличная точка). Неопределенность выходной переменной для конкретной входной комбинации означает, что эта комбинация никогда не будет иметь место либо что выходная величина может быть произвольной (0 или 1).

Схема реализует функцию f , если ее выход равен 1 для каждой комбинации, соответствующей 1-точке, равен 0 для каждой 0-точки и безразличны – для безразличной точки.

Каждая точка n -куба, определенного переменными X_1, X_2, \dots, X_n , может быть представлена в виде *терма полного произведения*, т. е. произведения, содержащего переменные или их дополнения. Такой терм равен 1 только для этой точки и называется **минтермом**.

Строки таблицы истинности удобно пронумеровать от 0 до 2^N-1 , считая X_1 наибольшим значащим разрядом. Тогда, например, минтермы для двух переменных будут определяться следующим образом:

$$m_0 = \bar{X}_1\bar{X}_2 \quad m_1 = \bar{X}_1X_2, \quad m_2 = X_1\bar{X}_2, \quad m_3 = X_1X_2.$$

Для трёх переменных

$$m_0 = \bar{X}_1\bar{X}_2\bar{X}_3 \quad m_1 = \bar{X}_1\bar{X}_2X_3 \quad m_2 = \bar{X}_1X_2\bar{X}_3 \quad m_3 = X_1\bar{X}_2\bar{X}_3$$

$$m_4 = \bar{X}_1X_2\bar{X}_3 \quad m_5 = X_1\bar{X}_2X_3 \quad m_6 = X_1X_2\bar{X}_3 \quad m_7 = X_1X_2X_3$$

Если задана комбинационная функция f , то ее простая схемная реализация может быть получена представлением каждой 1-точки соответствующим минтермом и суммированием последних. Такая реализация называется канонической суммой минтермов. Каждый минтерм реализуется логическим элементом И, на входы которого могут подаваться как X_i , так и \bar{X}_i . Суммирование выполняется с помощью логического элемента ИЛИ.

Например, пусть задана функция f трех переменных X_1, X_2, X_3 в виде следующей таблицы истинности:

Т а б л и ц а 1

| X_1 | X_2 | X_3 | f_1 | f_2 |
|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | - |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |

Тогда каноническая сумма минтермов функции f будет равна:

$$f_1 = \bar{X}_1\bar{X}_2X_3 + \bar{X}_1X_2\bar{X}_3 + X_1X_2\bar{X}_3 + X_1X_2X_3 = m_1 + m_4 + m_6 + m_7$$

Данная каноническая реализация называется двухступенчатой, т.к. максимальное число логических элементов, через которые проходит входной сигнал, равно двум (без учета инверторов).

Так как обычно каждый логический элемент характеризуется запаздыванием, то быстродействие схемы обратно пропорционально числу ступеней.

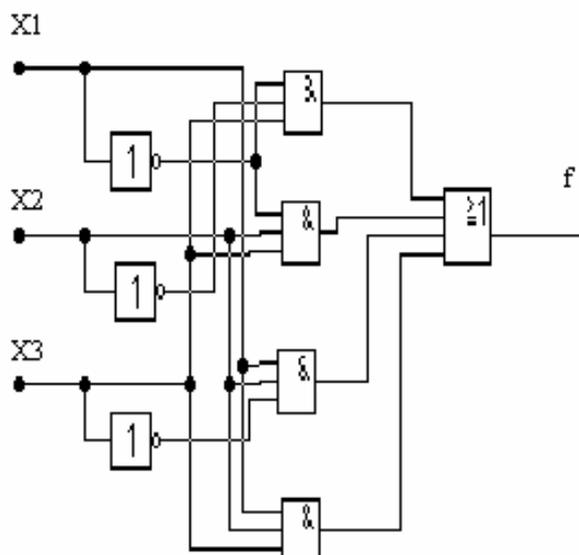


Рис. 16. Схемная реализация функции f минтермами

С помощью основных законов булевой алгебры канонические двухступенчатые реализации могут быть упрощены. Например, каноническая сумма минтермов функции, заданной следующей таблицей истинности, после упрощения может быть представлена одним минтермом:

$$f1 = X1\overline{X2} + X1X2 = X1(\overline{X2} + X2) = X1 \times 1 = X1$$

Т а б л и ц а 2

| X_1 | X_2 | f |
|-------|-------|-----|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

СДНФ (Совершенная Дизъюнктивная Нормальная Форма) — это такая ДНФ, которая удовлетворяет трём условиям:

- в ней нет одинаковых элементарных конъюнкций;
- в каждой конъюнкции нет одинаковых пропозициональных букв;
- каждая элементарная конъюнкция содержит каждую пропозициональную букву из входящих в данную ДНФ пропозициональных букв, причем в одинаковом порядке.

Метод Квайна — способ представления функции в ДНФ или КНФ с минимальным количеством членов и минимальным набором переменных.

Преобразование функции можно разделить на два этапа:

- на первом этапе осуществляется переход от канонической формы (СДНФ или СКНФ) к так называемой **сокращённой форме**;
- на втором этапе — переход от сокращённой формы к **минимальной форме**.

Первый этап (получение сокращённой формы)

Представим, что заданная функция f представлена в СДНФ. Для осуществления первого этапа преобразование проходит два действия:

1. *Операция склеивания/*
2. *Операция поглощения.*

Операция склеивания сводится к нахождению пар членов, соответствующих виду $w \cdot x$ или $w \cdot \bar{x}$, и преобразованию их в следующие выражения: $w \cdot x \vee w \cdot \bar{x} = w \cdot (x \vee \bar{x}) = w$.

Результаты склеивания w теперь играют роль дополнительных членов.

Потом выполняется *операция поглощения*. Она основана на равенстве $w \vee w \cdot z = w \cdot (1 \vee z) = w$ (член w поглощает выражение $w \cdot z$).

Вследствие этого действия из логического выражения вычёркиваются все члены, поглощаемые другими переменными, результаты которых получены в *операции склеивания*.

Обе операции первого этапа выполняются, пока это может быть осуществимо.

Применение этих операций продемонстрировано в табл.3.

Т а б л и ц а 3

| | | | | | | | | |
|--------------------|---|---|---|---|---|---|---|---|
| x_1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| x_2 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| x_3 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| $f(x_1, x_2, x_3)$ | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

СДНФ выглядит так:

$$f(x_1, x_2, x_3) = \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \cdot x_3 \vee x_1 \cdot x_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \cdot x_3$$

Результат операции склеивания нужен для преобразования функции на втором этапе (поглощения)

$$f(x_1, x_2, x_3) = \cancel{\bar{x}_1 \cdot x_2 \cdot \bar{x}_3} \vee \cancel{x_1 \cdot \bar{x}_2 \cdot \bar{x}_3} \vee \cancel{x_1 \cdot \bar{x}_2 \cdot x_3} \vee \cancel{x_1 \cdot x_2 \cdot \bar{x}_3} \vee \cancel{x_1 \cdot x_2 \cdot x_3} \vee x_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \vee x_1 \cdot \bar{x}_3 \vee x_1 \cdot x_3 \vee x_1 \cdot x_2$$

Членами результата склеивания являются

$$x_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \vee x_1 \cdot \bar{x}_3 \vee x_1 \cdot x_3 \vee x_1 \cdot x_2$$

Член $x_2 \cdot \bar{x}_3$ поглощает те члены исходного выражения, которые содержат $x_2 \cdot \bar{x}_3$, то есть первый и четвёртый. Эти члены вычёркиваются. Член $x_1 \cdot \bar{x}_2$ поглощает второй и третий, а член $x_1 \cdot x_3$ — пятый член исходного выражения.

Повторение обеих операций приводит к следующему выражению:

$$f(x_1, x_2, x_3) = x_2 \cdot \bar{x}_3 \vee \cancel{x_1 \cdot \bar{x}_2} \vee \cancel{x_1 \cdot \bar{x}_3} \vee \cancel{x_1 \cdot x_3} \vee \cancel{x_1 \cdot x_2} \vee x_1$$

Здесь склеивается пара членов $x_1 \cdot \bar{x}_2$ и $x_1 \cdot x_2$ (склеивание пары членов $x_1 \cdot \bar{x}_3$ и $x_1 \cdot x_3$ приводит к тому же результату), результат склеивания x_1 поглощает 2-, 3-, 4-, 5-й члены выражения. Дальнейшее проведение операций склеивания и поглощения оказывается невозможным, сокращённая форма выражения заданной функции (в данном случае она совпадает с минимальной формой)

$$f(x_1, x_2, x_3) = x_2 \cdot \bar{x}_3 \vee x_1$$

Члены сокращённой формы (в нашем случае это $x_2 \cdot \bar{x}_3$ и x_1) называются **простыми импликантами** функции. В итоге мы получили наиболее простое выражение, если сравнивать его с начальной версией — СДНФ. Структурная схема такого элемента показана на рис.17.

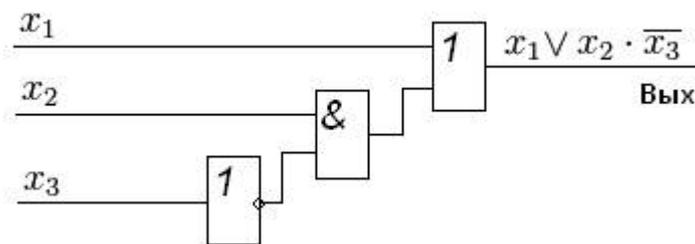


Рис. 17. Структурная схема функции

Второй этап (табличный) (получение минимальной формы)

Как и на первом этапе, в полученном равенстве могут содержаться члены, устранение которых никаким образом не повлияет на конечный результат. Следующий этап минимизации — удаление таких переменных. Таблица, представленная ниже, содержит значения истинности функции, по ней будет собрана следующая СДНФ.

Таблица 4

| | | | | | | | | | | | | | | | |
|--------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| x_1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| x_2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| x_3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| x_4 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| $f(x_1, x_2, x_3)$ | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

СДНФ, собранная по табл.4, выглядит следующим образом:

$$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 \vee \\ \vee \bar{x}_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4 \vee x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4 \vee x_1 \cdot x_2 \cdot x_3 \cdot x_4$$

Конечное выражение достигается за счёт повторного использования операций склеивания и поглощения:

$$\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_4 \vee \bar{x}_1 \cdot x_3 \cdot \bar{x}_4 \vee x_2 \cdot x_3 \cdot \bar{x}_4 \vee x_1 \cdot x_2 \cdot x_3$$

Члены этого выражения являются **простыми импликантами** выражения. Переход от сокращённой формы к минимальной осуществляется с помощью импликантной матрицы.

Члены СДНФ заданной функции вписываются в столбцы, а в строки — простые импликанты, то есть члены сокращённой формы. Отмечаются столбцы членов СДНФ, которые поглощаются отдельными простыми импликантами. В табл.5 простая импликанта $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3$ поглощает члены $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4$ и $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4$ (в первом и во втором столбцах поставлены крестики).

Вторая импликанта поглощает первый и третий члены СДНФ (указано крестиками) и т. д. Импликанты, не подлежащие исключению, образуют *ядро*. Такие импликанты определяются по вышеуказанной матрице. Для каждой из них имеется хотя бы один столбец, перекрываемый только этой импликантой.

В нашем примере ядро составляют импликанты $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3$ и $x_1 \cdot x_2 \cdot x_3$ (ими перекрываются второй и шестой столбцы). Исключение из сокращённой формы одновременно всех импликант, не входящих в ядро, невозможно, так как исключение одной из импликант может превратить другую в уже нелишний член.

Для получения минимальной формы достаточно выбрать из импликантов, не входящих в ядро, такое минимальное их число с минимальным количеством букв в каждом из этих импликант, которое обеспечит перекрытие всех столбцов, не перекрытых членами ядра. В рассматриваемом примере необходимо импликантами, не входящими в ядро, перекрыть третий и четвёртый столбцы матрицы. Это может быть достигнуто различными способами, но так как необходимо выбирать минимальное число импликант, то, очевидно, для перекрытия этих столбцов следует выбрать импликанту $\bar{x}_1 \cdot x_3 \cdot \bar{x}_4$.

Таблица 5

Импlicantная матрица

| Простая импликанта | $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4$ | $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot x_4$ | $\bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4$ | $\bar{x}_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4$ | $x_1 \cdot x_2 \cdot x_3 \cdot \bar{x}_4$ | $x_1 \cdot x_2 \cdot x_3 \cdot x_4$ |
|---|---|---|---|---|---|-------------------------------------|
| $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3$ | x | x | | | | |
| $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_4$ | x | | x | | | |
| $\bar{x}_1 \cdot x_3 \cdot \bar{x}_4$ | | | x | x | | |
| $x_2 \cdot x_3 \cdot \bar{x}_4$ | | | | x | x | |
| $x_1 \cdot x_2 \cdot x_3$ | | | | | x | x |

Минимальная дизъюнктивная нормальная форма (МДНФ) заданной функции:

$$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \cdot x_3 \vee \bar{x}_1 \cdot x_3 \cdot \bar{x}_4. \quad (1)$$

Структурная схема, соответствующая этому выражению, приведена на рис. 18. Переход от сокращённой схемы к МДНФ был осуществлён путём исключения лишних членов — импликант $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_4$ и $x_2 \cdot x_3 \cdot \bar{x}_4$. Покажем допустимость подобного исключения членов из логического выражения.

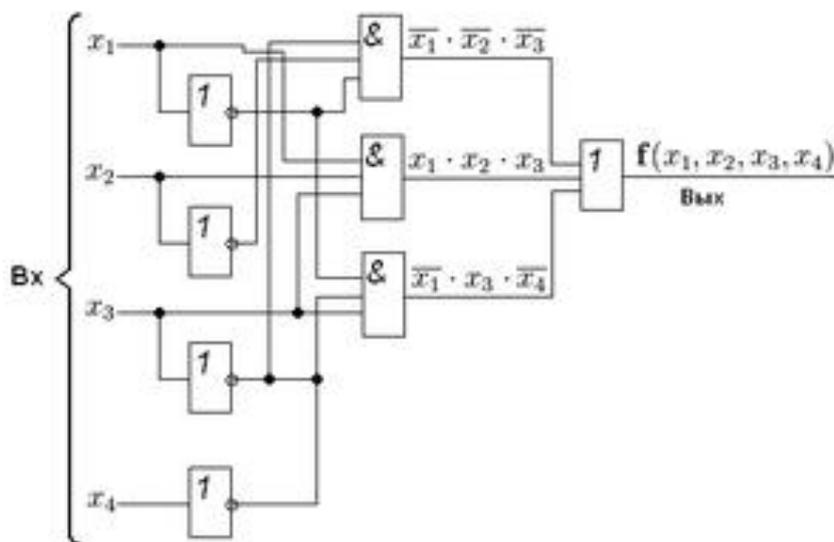


Рис. 18. Структурная схема функции

Импликанты $\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_4$ и $x_2 \cdot x_3 \cdot \bar{x}_4$ становятся равными лог. 1 соответственно при следующих наборах значений аргументов: $x_1=0, x_2=0, x_4=0$ и $x_2=1, x_3=1, x_4=0$.

Роль этих импликант в выражении сокращённой формы функции заключается лишь в том, чтобы на приведённых наборах значений аргументов присваивать функции $f(x_1, x_2, x_3, x_4)$ значение 1. Однако при этих наборах функция равна 1 из-за остальных импликант выражения. Действительно, подставив полученный набор значений переменных в формулу для $f(1)$, имеем:

- при $x_1=0, x_2=0, x_4=0$

$$f(0,0,x_3,0) = 1 \cdot 1 \cdot \bar{x}_3 \vee 0 \cdot 0 \cdot x_3 \vee 1 \cdot x_3 \cdot 1 = \bar{x}_3 \vee x_3 = 1;$$

- при $x_2=1, x_3=1, x_4=0$

$$f(x_1,1,1,0) = \bar{x}_1 \cdot 0 \cdot 0 \vee x_1 \cdot 1 \cdot 1 \vee \bar{x}_1 \cdot 1 \cdot 1 = x_1 \vee \bar{x}_1 = 1.$$

3.3. Табличный метод получения простых импликант

В терм-произведении могут отсутствовать некоторые переменные. В этом случае данное терм-произведение P определяет несколько точек в n -кубе. Говорят, что эти точки покрыты P . Например, для переменных X_1, X_2 и X_3 терм $X_1\overline{X_3}$ покрывает две точки: $(1,0,0)$ и $(1,1,0)$.

Вообще терм-произведение K переменных покрывает в n -кубе 2^{n-k} точек и обычно называется $(n-k)$ -куб.

В соответствии с этим минтерм представляет собой 1-куб.

Терм-произведение P , покрывающее только 1-точки и, возможно, неопределенные точки, называется импликантой функции f . В последнем примере импликантами являются минтермы m_1, m_3, m_6 и m_7 . Импликанта P_1 функции f называется простой импликантой, если не существует другая импликанта P_2 , покрывающая все точки импликанты P_1 . Например, если f_2 функция задана таблицей истинности (табл.6), то термпроизведение $P_1 = \overline{X_1}\overline{X_2}$ покрывает две точки $(0,0,0)$ и $(0,0,1)$. Так как для обеих точек $f_2=1$, то P_1 есть импликанта f_2 . Однако P_2 не является простой импликантой f_2 , поскольку существует импликанта $P_2 = \overline{X_1}$, которая также перекрывает обе эти точки.

Известно, что в минимальной двухступенчатой реализации вида И-ИЛИ комбинационной функции f выход каждого логического элемента И есть терм-произведение P_1 , являющееся простой импликантой f . Поэтому, в минимальной двухступенчатой реализации вида И-ИЛИ комбинационной функции f выход каждого логического элемента И является простой импликантой и каждая 1-точка функции f покрывается хотя бы одной из этих простых импликант.

Отсюда получается следующий алгоритм решения задачи получения минимальной двухступенчатой реализации вида И-ИЛИ функции f :

1. Найти полное множество простых импликант функции f .
2. Из этого множества выбрать подмножество минимальной стоимости, покрывающее все 1-точки этой функции.

Понятие «стоимость» простых импликант связано с критерием оптимизации. Если критерием оптимизации является минимизация числа конечных элементов, то стоимость каждой простой импликанты принимается равной 1, а стоимость импликанты с одним литером равна нулю (не требуется элемент И). В задачах минимизации общего числа входов конечных элементов стоимость простой импликанты P_k на

единицу больше количества литералов, образующих P_k (в случае одного литерала стоимость равна 1, поскольку не требуется логический элемент И для реализации P_k). Стоимость множества простых импликант равен сумме стоимостей каждой из них.

Квайном и Мак-Класки предложен следующий алгоритм получения простых импликант:

1. Составить таблицу для всех 1-точек и неопределенных точек функции f , разбитых на классы $S_0, S_1, S_2, \dots, S_n$, где S_i содержит все 1-точки, для которых i входных переменных равны 1, а $(n-i)$ равен 0.

2. Сравнить каждую строку в S_i с каждой строкой в S_{i+1} для всех i . Для пар, различающихся одним литералом, образовать новые импликанты, покрывающие обе точки. Поместить новые импликанты в класс S_i и пометить использованные для их формирования термы меткой (\vee).

3. Повторить шаг 2, используя S'_i и S'_{i+1} для образования S''_i . Продолжать до тех пор, пока новые комбинации не окажутся невозможными. Простыми импликантами будут термы, не помеченные меткой.

Пример. Пусть f есть функция входных переменных X_1, X_2, X_3 , представленных в виде таблицы истинности 1-точек:

Таблица 6

| | X_1 | X_2 | X_3 | f | метка |
|---|-------|-------|-------|-----|--------|
| 1 | 0 | 0 | 1 | 1 | \vee |
| 2 | 0 | 1 | 0 | 1 | \vee |
| 3 | 1 | 0 | 0 | 1 | |
| 4 | 0 | 1 | 1 | 1 | \vee |
| 5 | 1 | 1 | 1 | D | \vee |
| 6 | 0 | - | 1 | 1 | |
| 7 | 0 | 1 | - | 1 | |
| 8 | - | 1 | 1 | 1 | |

6-я строчка образована слиянием 1-й и 4-й строчек;
7-я строчка – соответственно слиянием 2-й и 4-й строчек;
8-я строчка – слиянием 4-й и 5-й строчек.

Дальнейшее слияние точек невозможно. Следовательно, получено 4 простые импликанты (3, 6, 7 и 8 строчки): $X_1\bar{X}_2\bar{X}_3$, \bar{X}_1X_3 , \bar{X}_1X_2 , X_2X_3 .

3.4. Получение простых импликант с помощью карт Карно

Другой метод нахождения простых импликант, удобный для функций с малым числом переменных, основан на использовании карт Карнау (Карно). Карта Карнау для n переменных содержит 2^n ячеек, каждая из которых соответствует одной из 2^n возможных комбинаций входных переменных. Соседние ячейки карты Карнау различаются

только на одну переменную. В этих картах ровно половина ячеек относится к значению переменной $X_i = 1$, а другая половина – к $X_i = 0$ (\bar{X}_i). Разбиение пополам карты Карно другим способом дает возможность учитывать другую переменную и т.д.

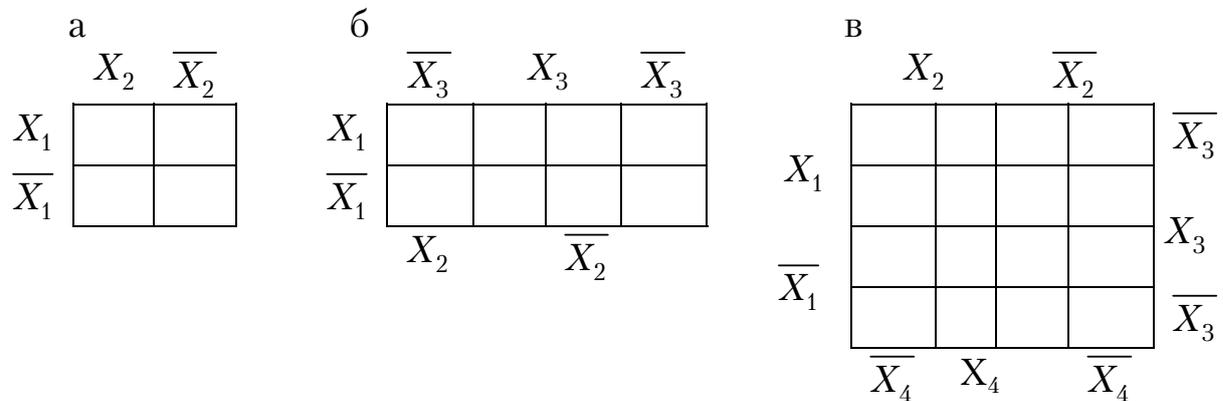


Рис.19. Карты Карно для:
а – двух переменных; б – трех переменных; в – четырех переменных

Ячейки карты Карно заполняются либо единицами, либо нулями (нули можно не изображать), либо безразличным состоянием (символ D), в зависимости от значения функции в точке, соответствующей данной ячейке (данному терму). Например, для выражения $F = \bar{X}_1 X_2 \bar{X}_3 \wedge \bar{X}_1 X_2 X_3$ карта Карно будет иметь следующий вид:

| | | | |
|-------------|-------------|-------------|-------------|
| | X_2 | \bar{X}_2 | |
| X_1 | D | | 1 |
| \bar{X}_1 | 1 | 1 | |
| | \bar{X}_3 | X_3 | \bar{X}_3 |

Пустые ячейки соответствуют 0-точкам.

Импликанты функции образуются комбинированием 1-точек и неопределенных точек, которые различаются только одной входной переменной. На карте Карно такие термы представляются множеством смежных 1-точек (или неопределенных точек). В приведенном примере можно объединить, например, точки $\bar{X}_1 X_2 \bar{X}_3$ и $\bar{X}_1 X_2 X_3$ с образованием новой импликанты $\bar{X}_1 X_2$, точки $\bar{X}_1 X_2 X_3$ и $\bar{X}_1 \bar{X}_2 X_3$ – с образованием импликанты $X_2 X_3$. Невозможно объединить ни с одной единичной точкой точку $X_1 \bar{X}_2 \bar{X}_3$, которая не покрыта предыдущими импликантами.

Карту Карно можно свернуть в цилиндр вокруг как горизонтальной, так и вертикальной оси, поскольку ячейки верхней и нижней строк или левого и правого столбцов также различаются только одной переменной и являются смежными ячейками, которые можно объединить.

3.5. Анализ и синтез комбинационных схем с разветвленным выходом

При реализации нескольких комбинационных функций часто удается получить схему с разветвленным выходом, требующую меньше логических элементов, чем отдельные схемы, реализующие каждую из этих функций. Это достигается распределением логики среди участков цепи, реализующих различные функции. Такое распределение предполагает, что выход логического элемента может быть связан с входами более чем одного логического элемента.

Число логических элементов, соединенных с выходом некоторого логического элемента, называется коэффициентом разветвления по выходу. Если этот коэффициент равен единице, то задача получения минимальной двухступенчатой схемы для реализации множества функций не отличается от задачи реализации каждой индивидуальной функции минимальной двухступенчатой реализации.

Для двухступенчатых схем вида И-ИЛИ совместное использование логических элементов возможно только при наличии общей импликанты для этих функций.

Например, для двух функций f_1 и f_2 заданы карты Карно (рис. 20).

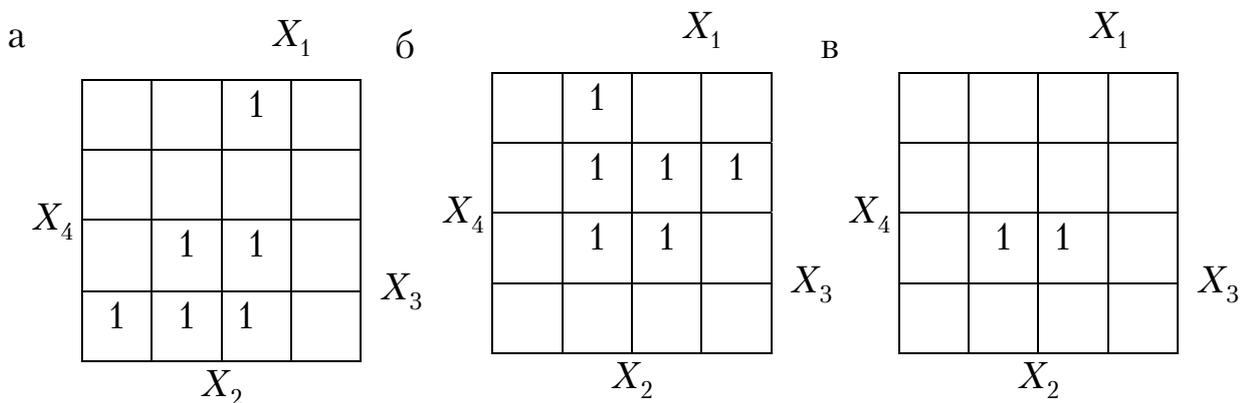


Рис. 20. Карты Карно:
а – для функции f_1 ; б – для функции f_2 ; в – для функции $f_1 f_2$

Здесь импликанта $X_2X_3X_4$ является простой импликантой произведения функций $f_1 f_2$.

$$f_1 = X_2X_3X_4 + \bar{X}_1X_3\bar{X}_4 + X_1X_2\bar{X}_4; \quad f_2 = X_2X_3X_4 + \bar{X}_1X_2\bar{X}_3 + X_1\bar{X}_3X_4$$

$$f_1f_2 = X_2X_3X_4$$

Минимальная двухступенчатая реализация двух функций f_1 и f_2 приведена на рис. 21.

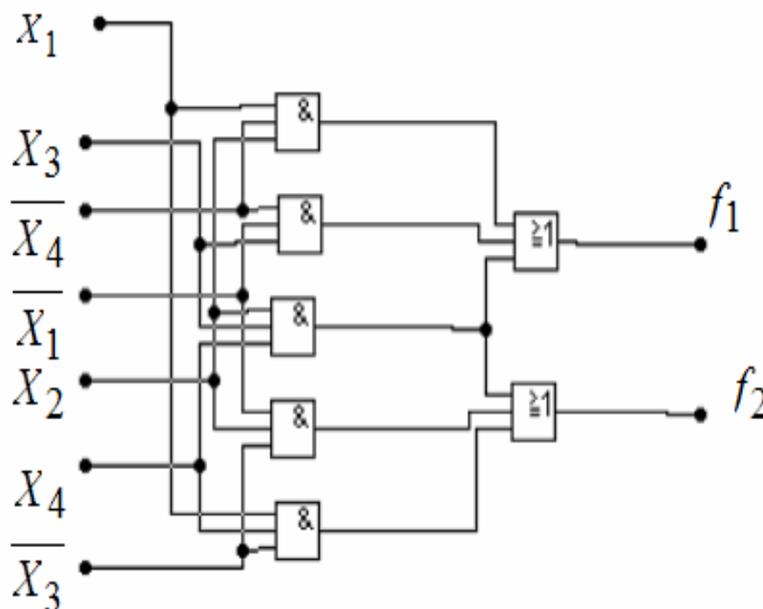


Рис. 21. Двухступенчатая реализация функций f_1 и f_2

Дадим понятие многовыходовой импликанты. Пусть $F = \{f_1, f_2, \dots, f_n\}$ – множество комбинационных функций, а F' – подмножество F . Термпроизведение P называется простой многовыходовой импликантой подмножества F' , если оно является простой импликантой функции $G' = \prod f'$ (произведение всех функций подмножества F') $f' \in F$. Так как F' может содержать одну функцию f'_i , то любая простая импликанта функции f'_i является простой многовыходовой импликантой множества F . Эти импликанты используются для построения минимальных двухступенчатых реализаций функции множества F , причем некоторые импликанты используются для реализации только одной функции, а другие – для нескольких.

Известно, что для множества комбинационных функций $F = \{f_1, f_2, \dots, f_n\}$ существует минимальная двухступенчатая реализация вида И-ИЛИ, в которой выход каждого логического элемента И представляет собой многовыходовую импликанту.

Отсюда минимальную двухступенчатую реализацию вида И-ИЛИ для множества комбинационных функций можно получить по следующему алгоритму.

1. Получить множество простых многовыходовых импликант (простые импликанты отдельной функции f также включаются в это множество).

2. Выбрать множество простых многовыходовых импликант минимальной стоимости для реализации множества комбинационных функций. Если выбирается простая многовыходовая импликанта множества функций $F' \in F$, то она используется для реализации каждой $f \in F'$.

3.6. Получение простых многовыходовых импликант табличным методом

Табличный метод получения простых импликант отдельных функций можно использовать и для получения многовыходовых импликант. Для этого необходимо выполнить следующие действия:

1. Составить таблицу из всех 1-точек и неопределенных точек для полного множества функций f_1, f_2, \dots, f_n . Разделить это множество на подмножества $S_0, S_1, S_2, \dots, S_i$, где множество содержит все такие 1-точки с i переменными, равными 1. Добавить в таблицу множество n столбцов, в которых в каждой строке записать единицу, если при данной комбинации входных переменных функции $f_i = 1$, и нуль в противоположном случае.

2. Составить комбинацию строк этой таблицы и записать в подмножества S'_i . При этом следует учитывать следующие правила:

- объединяются только те строки, в которых имеет различное значение только одна из входных переменных (во вновь образуемой строке в этом столбце ставится прочерк – безразличное состояние);

- результат объединения 2-х строк для каждой из функций f_i равен логическому умножению значений этой функции в этих строках (1 – только если в обеих строках в данном столбце стоит 1);

- помечаются знаком «v» только те из используемых для слияния строк, в которых значения функций f_1, \dots, f_n совпадают со значениями в результирующей строке;

- если после образования новой строки все функции приняли значение нуль, то такая строка вычеркивается.

3. Продолжить, если возможно, объединение строк тем же способом.

Строки, не участвующие в образовании новых строк, являются простыми многовыходовыми импликантами.

В качестве примера рассмотрим задачу нахождения простых многовыходовых импликант для двух функций f_1 и f_2 (табл.7). После преобразований таблицы по вышеприведенной методике получим многовыходовые импликанты для функции f_1 : $\overline{X_1}X_3\overline{X_4}$, $X_1X_2\overline{X_4}$, $X_2X_3X_4$, X_2X_3 , а для функции f_2 : $\overline{X_1}X_2\overline{X_3}$, $X_1\overline{X_3}X_4$, $X_2X_3X_4$, X_2X_4 .

Т а б л и ц а 7

| Подмножество | N | Переменные | | | | f_1 | f_2 | Пометка использования | Слитые строки |
|--------------|---|------------|-------|-------|-------|-------|-------|-----------------------|--------------------------------|
| | | X_1 | X_2 | X_3 | X_4 | | | | |
| S_0 | | | | | | | | | |
| S_1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | ✓ | |
| | 2 | 0 | 1 | 0 | 0 | 0 | 1 | ✓ | |
| S_2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | ✓ | |
| | 2 | 0 | 1 | 1 | 0 | 1 | 0 | ✓ | |
| | 3 | 1 | 0 | 0 | 1 | 0 | 1 | ✓ | |
| | 4 | 0 | 1 | 0 | 1 | 0 | 1 | ✓ | |
| S_3 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | ✓ | |
| | 2 | 0 | 1 | 1 | 1 | 1 | 1 | ✓ | |
| | 3 | 1 | 1 | 0 | 1 | 0 | 1 | ✓ | |
| S_4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ✓ | |
| S'_1 | 1 | 0 | - | 1 | 0 | 1 | 0 | | $S_{1-1}+S_{2-2}$ |
| | 2 | 0 | 1 | 0 | - | 0 | 1 | | $S_{1-2}+S_{2-4}$ |
| S'_2 | 1 | 1 | 1 | - | 0 | 1 | 0 | | $S_{2-1}+S_{3-1}$ |
| | 2 | - | 1 | 1 | 0 | 1 | 0 | ✓ | $S_{2-2}+S_{3-1}$ |
| | 3 | 0 | 1 | 1 | - | 1 | 0 | ✓ | $S_{2-2}+S_{3-2}$ |
| | 4 | 1 | - | 0 | 1 | 0 | 1 | | $S_{2-3}+S_{3-3}$ |
| | 5 | 0 | 1 | - | 1 | 0 | 1 | ✓ | $S_{2-4}+S_{3-2}$ |
| | 6 | - | 1 | 0 | 1 | 0 | 1 | ✓ | $S_{2-4}+S_{3-3}$ |
| S'_3 | 1 | 1 | 1 | 1 | - | 1 | 0 | ✓ | $S_{3-1}+S_{4-1}$ |
| | 2 | - | 1 | 1 | 1 | 1 | 1 | | $S_{3-2}+S_{4-1}$ |
| | 3 | 1 | 1 | - | 1 | 0 | 1 | ✓ | $S_{3-3}+S_{4-1}$ |
| S'_2 | 1 | - | 1 | 1 | - | 1 | 0 | | $S'_{2-2}+S'_{3-2}$ |
| | 2 | - | 1 | 1 | - | 1 | 0 | | $S'_{2-3}+S'_{3-1}$ |
| | 3 | - | 1 | - | 1 | 0 | 1 | | $S'_{2-5}+S'_{3-3}(+S'_{2-6})$ |

Простые многовыходовые импликанты можно также получить с помощью карт Карно индивидуальных функций и произведений всех подмножеств этого множества функций.

3.7. Принципы минимизации

Основным методом минимизации логических функций, представленных в виде СДНФ или СКНФ, является операция попарного неполного склеивания и элементарного поглощения. Операция попарного склеивания осуществляется между двумя термами (членами), содержащими одинаковые переменные, входящие которых (прямые и инверсные) совпадают для всех переменных, кроме одной. В этом случае все переменные, кроме одной, можно вынести за скобки, а оставшиеся в скобках прямое и инверсное входящие одной переменной подвергнуть склейке. Например:

$$X_1X_2X_3X_4 \vee X_1X_2X_3\bar{X}_4 = X_1X_2X_3(X_4 \vee \bar{X}_4) = X_1X_2X_3.$$

Аналогично для КНФ:

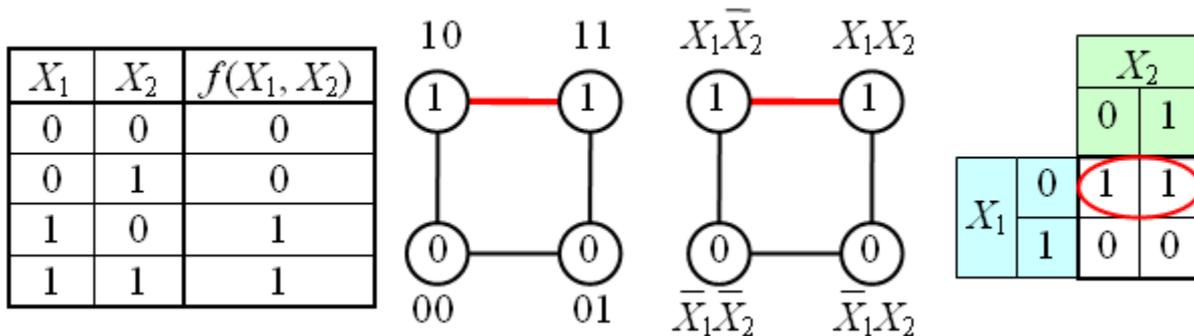
$$(\bar{X}_1X_2X_3X_4)(\bar{X}_1X_2\bar{X}_3X_4) = \bar{X}_1X_2X_3X_4X_3\bar{X}_3 = \bar{X}_1X_2X_4.$$

Возможность поглощения следует из очевидных равенств

$$A \vee \bar{A} = 1; \quad A\bar{A} = 0$$

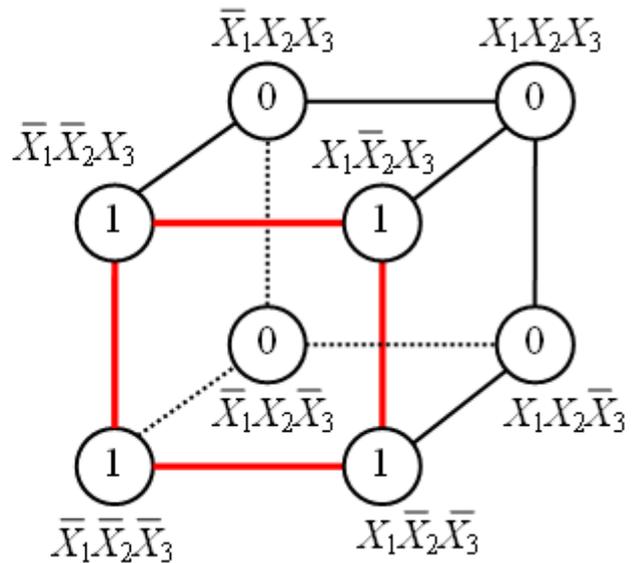
Таким образом, главной задачей при минимизации СДНФ и СКНФ является поиск термов, пригодных к склейке с последующим поглощением, что для больших форм может оказаться достаточно сложной задачей. Карты Карно предоставляют наглядный способ отыскания таких термов.

Как известно, булевы функции N переменных, представленные в виде СДНФ или СКНФ, могут иметь в своём составе $2N$ различных термов. Все эти члены составляют некоторую структуру, топологически эквивалентную N -мерному кубу, причём любые два терма, соединённые ребром, пригодны для склейки и поглощения. На рисунке изображены простая таблица истинности для функции из двух переменных и соответствующий этой таблице 2-мерный куб (квадрат), а также 2-мерный куб с обозначением членов СДНФ и эквивалентная таблица для группировки термов:



В случае функции трёх переменных приходится иметь дело с трёхмерным кубом. Это сложнее и менее наглядно, но технически возможно. На рисунке в качестве примера показаны таблица истинности для булевой функции трёх переменных и соответствующий ей куб.

| X_1 | X_2 | X_3 | $f(X_1, X_2, X_3)$ |
|-------|-------|-------|--------------------|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

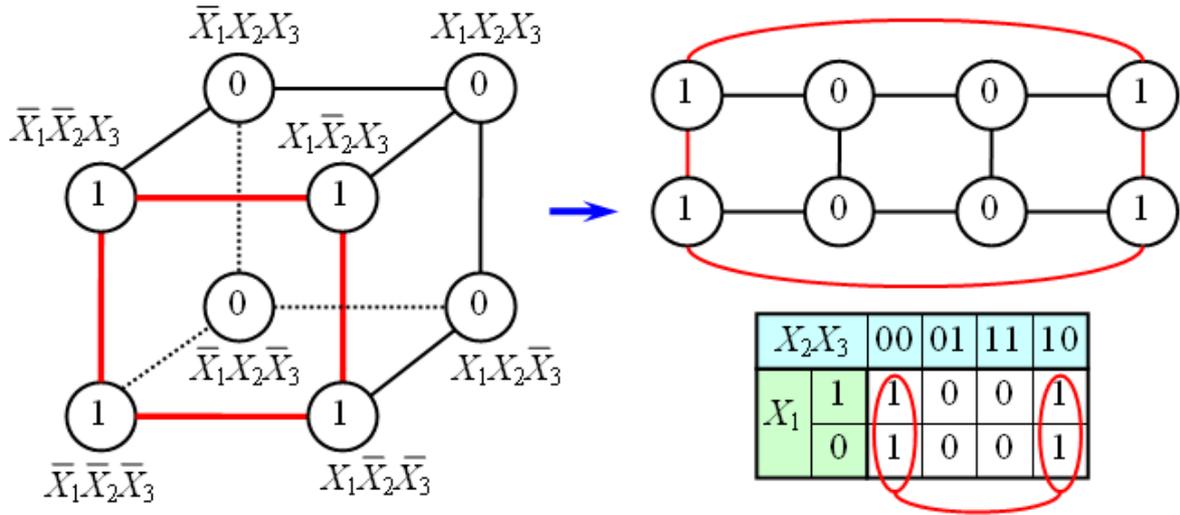


Как видно из рисунка, для трёхмерного случая возможны более сложные конфигурации термов. Например, четыре терма, принадлежащие одной грани куба, объединяются в один терм с поглощением двух переменных:

$$\begin{aligned} \overline{X_1 X_2 X_3} \vee X_1 \overline{X_2 X_3} \vee \overline{X_1 X_2} X_3 \vee X_1 \overline{X_2} X_3 &= \overline{X_1} (\overline{X_1 X_3} \vee \overline{X_1} X_3 \vee X_1 \overline{X_3} \vee X_1 X_3) = \\ &= \overline{X_2} (\overline{X_1} \vee X_1) (\overline{X_3} \vee X_3) = \overline{X_2} \end{aligned}$$

В общем случае можно сказать, что $2K$ термов, принадлежащие одной K -мерной грани гиперкуба, склеиваются в один терм, при этом поглощаются K переменных.

Для упрощения работы с булевыми функциями большого числа переменных был предложен следующий удобный приём. Куб, представляющий собой структуру термов, разворачивается на плоскость как показано на следующем рисунке. Таким образом появляется возможность представлять булевы функции с числом переменных больше двух в виде плоской таблицы. При этом следует помнить, что порядок кодов термов в таблице (00 01 11 10) не соответствует порядку следования двоичных чисел, а клетки, находящиеся в крайних столбцах таблицы, соседствуют между собой.



Аналогичным образом можно работать с функциями четырёх, пяти и более переменных. Примеры таблиц для $N=4$ и $N=5$ приведены на рис.22. Для этих таблиц следует помнить, что соседними являются клетки, находящиеся в соответственных клетках крайних столбцов и соответственных клетках верхней и нижней строк. Для таблиц пяти и более переменных нужно учитывать также, что квадраты 4×4 виртуально находятся друг над другом в третьем измерении; поэтому соответственные клетки двух соседних квадратов 4×4 являются соседними, и соответствующие им термы можно склеивать.

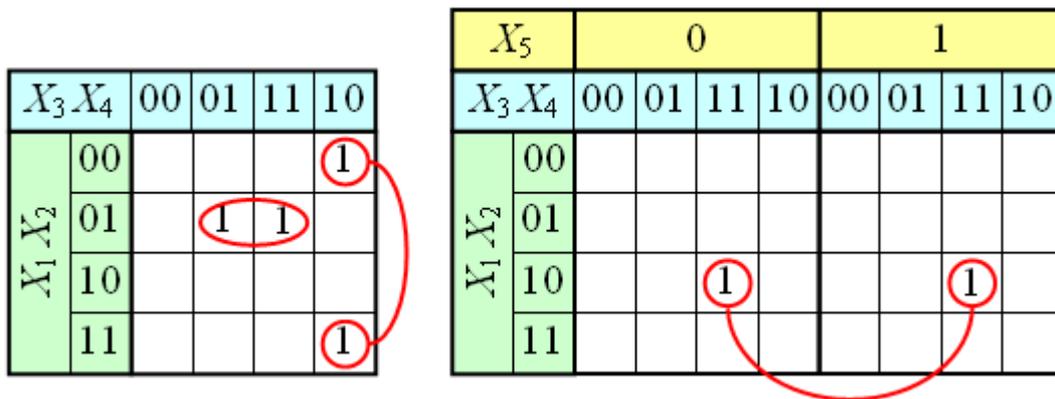


Рис. 22. Таблицы 4-х и 5-ти переменных

Описание

Карта Карно может быть составлена для любого количества переменных, однако удобно работать при количестве переменных не более пяти. По сути, карта Карно — это таблица истинности, составленная в 2-мерном виде. Благодаря использованию кода Грея в ней верхняя строка является соседней с нижней, а правый столбец — соседним с левым, т.о. вся Карта Карно сворачивается в фигуру тор (бублик). На пересечении строки и столбца проставляется соответ-

ствующее значение из таблицы истинности. После того как Карта заполнена, можно приступать к минимизации.

Если необходимо получить минимальную ДНФ, то в Карте рассматриваем только те клетки, которые содержат единицы; если нужна КНФ, то рассматриваем те клетки, которые содержат нули. Сама минимизация производится по следующим правилам (на примере ДНФ):

1. Объединяем смежные клетки, содержащие единицы, в область так, чтобы одна область содержала 2^n (n целое число = $0 \dots \infty$) клеток (помним про то, что крайние строки и столбцы являются соседними между собой), в области не должно находиться клеток, содержащих нули; карта Карно 4.

2. Область должна располагаться симметрично оси(ей) (оси располагаются через каждые четыре клетки).

3. Несмежные области, расположенные симметрично оси(ей), могут объединяться в одну.

4. Область должна быть как можно больше, а количество областей как можно меньше.

5. Области могут пересекаться.

6. Возможно несколько вариантов покрытия.

Далее берём первую область и смотрим, какие переменные не меняются в пределах этой области, выписываем конъюнкцию этих переменных; если неменяющаяся переменная нулевая, проставляем над ней инверсию. Берём следующую область, выполняем то же самое, что и для первой, и т.д. для всех областей. Конъюнкции областей объединяем дизъюнкцией.

Например (для карт на 2 переменные):

| | | | | | |
|-----------------------|------------------|------------------|------------------|------------------|------------------|
| X_2 | X_2 | X_2 | X_2 | X_2 | X_2 |
| $X_1 \backslash$ | $X_1 \backslash$ | $X_1 \backslash$ | $X_1 \backslash$ | $X_1 \backslash$ | $X_1 \backslash$ |
| 0 0 1 | 0 0 1 | 0 0 1 | 0 0 1 | 0 1 0 | 0 1 1 |
| 0 1 0 | 0 0 1 | 0 0 0 | 0 0 0 | 0 1 0 | 0 1 1 |
| 1 0 0 | 1 0 0 | 1 0 1 | 1 1 0 | 1 1 0 | 1 0 0 |
| $\bar{X}_1 \bar{X}_2$ | $\bar{X}_1 X_2$ | $X_1 X_2$ | $X_1 \bar{X}_2$ | \bar{X}_2 | \bar{X}_1 |

| | | | | | |
|------------------|------------------|------------------|------------------|------------------|------------------|
| X_2 | X_2 | X_2 | X_2 | X_2 | X_2 |
| $X_1 \backslash$ |
| 0 1 0 | 0 0 1 | 0 0 1 | 0 1 0 | 0 1 1 | 0 1 1 |
| 0 1 0 | 0 0 1 | 0 0 1 | 0 1 0 | 0 1 1 | 0 1 1 |
| 1 0 1 | 1 1 0 | 1 1 1 | 1 1 1 | 1 1 0 | 1 0 1 |
| S_1 | S_1 | S_2 | S_1, S_2 | S_1, S_2 | S_1, S_2 |

| | | | | | |
|----------------------------|------------------------|------------------|------------------------|------------------------------|------------------------|
| $S_1 \vee S_2 =$ | $S_1 \vee S_2 =$ | $S_1 \vee S_2 =$ | $S_1 \vee S_2 =$ | $S_1 \vee S_2 =$ | $S_1 \vee S_2 =$ |
| $= X_1 X_2 \vee$ | $= X_1 \bar{X}_2 \vee$ | $= X_1 \vee X_2$ | $= X_1 \vee \bar{X}_2$ | $= \bar{X}_1 \vee \bar{X}_2$ | $= X_2 \vee \bar{X}_1$ |
| $\vee \bar{X}_1 \bar{X}_2$ | $\vee \bar{X}_1 X_2$ | | | | |

Для КНФ всё то же самое, только рассматриваем клетки с нулями, неменяющиеся переменные в пределах одной области объединяем в дизъюнкции (инверсии проставляем над единичными переменными), а дизъюнкции областей объединяем в конъюнкцию. На этом минимизация считается законченной. Так, для карты Карно на рис.22 выражение в формате ДНФ будет иметь вид:

$$f(X_1, X_2, X_3, X_4) = S_1 \vee S_2 \vee S_3 = \bar{X}_1 \bar{X}_4 \vee X_1 X_2 X_4 \vee \bar{X}_2 \bar{X}_4$$

В формате КНФ:

$$f(X_1, X_2, X_3, X_4) = (S_1)(S_2)(S_3) = (X_1 \vee \bar{X}_4)(X_2 \vee \bar{X}_4)(\bar{X}_1 \vee \bar{X}_2 \vee X_4)$$

Так же из ДНФ в КНФ и обратно можно перейти, используя законы де Моргана.

Примеры

У мальчика Коли есть мама, папа, дедушка и бабушка. Коля пойдёт гулять на улицу, если ему разрешат хотя бы двое родственников.

Для краткости обозначим родственников Коли через буквы:

мама — x_1

папа — x_2

дедушка — x_3

бабушка — x_4

Условимся обозначать согласие родственников единицей, несогласие — нулём. Возможность пойти погулять обозначим буквой f , Коля идёт гулять — $f = 1$, Коля гулять не идёт — $f = 0$.

Составим таблицу истинности:

| x_1 | x_2 | x_3 | x_4 | f |
|-------|-------|-------|-------|-----|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Перерисуем таблицу истинности в 2-мерный вид:

| | | $x_3 \ x_4$ | | | |
|-------------|----|-------------|----|----|----|
| | | 00 | 01 | 10 | 11 |
| $x_1 \ x_2$ | 00 | | | | |
| | 01 | | | | |
| | 10 | | | | |
| | 11 | | | | |

Переставим в ней строки и столбцы в соответствии с кодом Грея. Получим карту Карно (рис. 23).

| | | X3 X4 | | | | | |
|-------|----|-------|----|----|----|--|--|
| | | 00 | 01 | 11 | 10 | | |
| X1 X2 | 00 | | | | | | |
| | 01 | | | | | | |
| | 11 | | | | | | |
| | 10 | | | | | | |

Рис. 23. Карта Карно

Заполним её значениями из таблицы истинности:

| | | X3 X4 | | | | | |
|-------|----|-------|----|----|----|--|--|
| | | 00 | 01 | 11 | 10 | | |
| X1 X2 | 00 | 0 | 0 | 1 | 0 | | |
| | 01 | 0 | 1 | 1 | 1 | | |
| | 11 | 1 | 1 | 1 | 1 | | |
| | 10 | 0 | 1 | 1 | 1 | | |

Минимизируем в соответствии с правилами:

1.1. Все области содержат 2^n клеток.

2.2. Так как Карта Карно на четыре переменные, оси располагаются на границах Карты и их не видно (подробнее смотри пример Карты на 5 переменных).

3.3. Так как Карта Карно на четыре переменные, все области симметрично осей — смежные между собой (подробнее смотри пример Карты на 5 переменных).

4.4. Области S_3, S_4, S_5, S_6 максимально большие.

5.5. Все области пересекаются (необязательное условие).

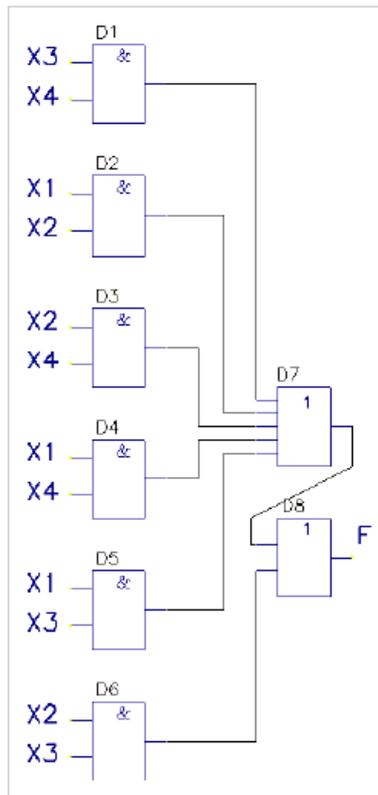
6.6. В данном случае рациональный вариант только один.

$$\begin{aligned}
 f(X_1, X_2, X_3, X_4) &= S_1 \vee S_2 \vee S_3 \vee S_4 \vee S_5 \vee S_6 = \\
 &= X_3 X_4 \vee X_1 X_2 \vee X_2 X_4 \vee X_1 X_4 \vee X_1 X_3 \vee X_2 X_3
 \end{aligned}$$

| | | X3 X4 | | | | | |
|-------|----|-------|----|----|----|-------|--|
| | | 00 | 01 | 11 | 10 | | |
| X1 X2 | 00 | 0 | 0 | 1 | 0 | S_3 | |
| | 01 | 0 | 1 | 1 | 1 | S_6 | |
| | 11 | 1 | 1 | 1 | 1 | S_2 | |
| | 10 | 0 | 1 | 1 | 1 | S_5 | |

S_4 S_1

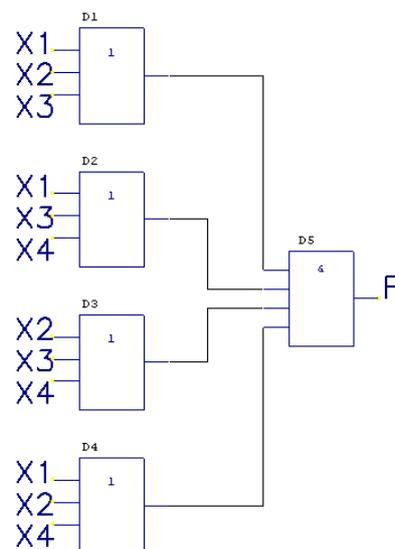
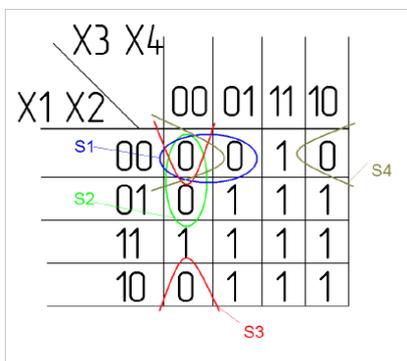
Теперь по полученной минимальной ДНФ можно построить логическую схему:



Из-за отсутствия в наличии шестивходового элемента ИЛИ, реализующего функцию дизъюнкции, пришлось каскадировать пяти и двухвходовые элементы (D7, D8).

Составим мин. КНФ:

$$f(X_1, X_2, X_3, X_4) = (S_1)(S_2)(S_3)(S_4) = (X_1 \vee X_2 \vee X_3)(X_1 \vee X_3 \vee X_4)(X_2 \vee X_3 \vee X_4)(X_1 \vee X_2 \vee X_4)$$



4. СИНТЕЗ ТИПОВЫХ УЗЛОВ ЦИФРОВЫХ СХЕМ

При проектировании цифровых вычислительных устройств можно выделить ряд типовых узлов, которые в той или иной мере могут присутствовать в устройстве. К таким типовым узлам следует отнести триггеры разнообразных типов, регистры, счётчики, сумматоры, дешифраторы, шифраторы и преобразователи кодов, распределители и мультиплексоры, устройства сравнения кодов и др.

4.1. Синтез триггеров

Синтез триггеров, как конечных автоматов, имеющих 2 устойчивых состояния, осуществляется с помощью таблиц состояний и таблиц возбуждений.

Триггер – это логическое устройство с двумя устойчивыми состояниями 0 и 1, имеющее несколько входов и два выхода (один – прямой, а другой – инверсный).

Триггер – элементарная ячейка оперативной памяти – один из базовых (основных) элементов цифровой техники.

RS-триггер получил название по названию своих входов. Вход S (англ. *Set* – установить) позволяет устанавливать выход триггера Q в единичное состояние. (Устанавливать означает записывать логическую единицу.) Вход R (англ. *Reset* – сбросить) позволяет сбрасывать выход триггера Q (англ. *Quit* – выход) в нулевое состояние.

Схемная реализация RS -триггера приведена на рис. 24.

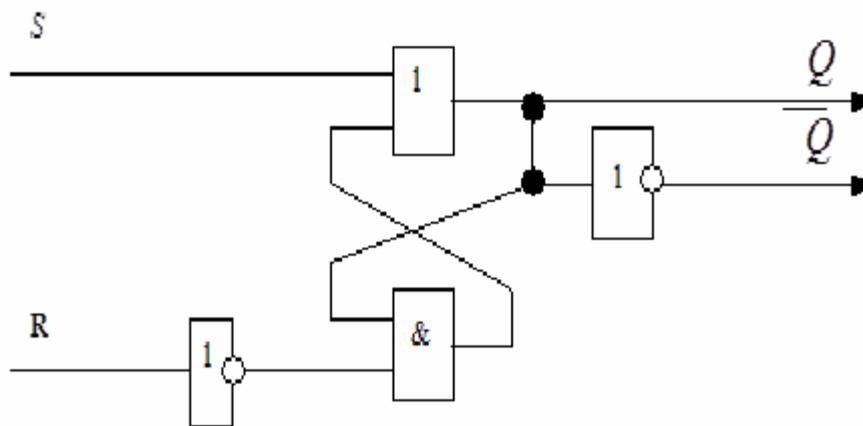


Рис. 24. Схема реализации RS -триггера

Схема реализации RS -триггера на элементах И-НЕ приведена на рис.25. Здесь входы установки триггера R и S в нулевое и единичное состояния являются инверсными.

Данная реализация соответствует логическому выражению RS -триггера, преобразованного следующим образом:

$$Q_{t+1} = S + Q_t \bar{R} = \overline{\overline{S + Q_t \bar{R}}} = \overline{\bar{S} \cdot \overline{Q_t \bar{R}}} = \overline{\bar{S} \cdot Q_t \cdot \bar{R}}$$

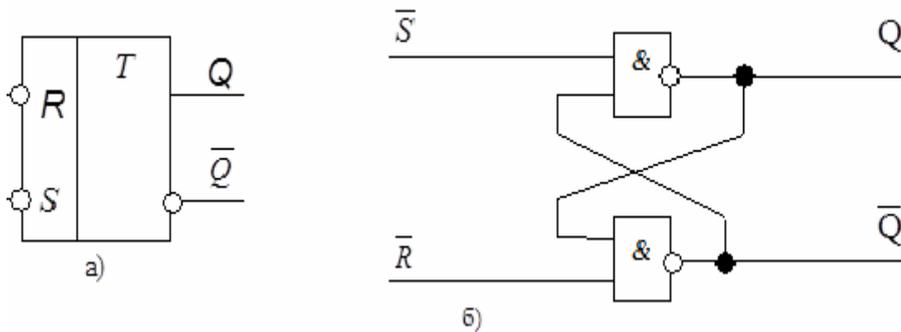


Рис.25. RS -триггер на элементах И-НЕ:
а – условное обозначение; б – схемная реализация

Существуют другие виды триггеров:

- **D -триггер:**
 - асинхронный;
 - синхронный;
- **триггер задержки;**
- **JK -триггер.**

Асинхронный D -триггер не применяют, так как его выход просто повторяет входной сигнал после окончания переходного процесса. В синхронном D -триггере, при отсутствии синхроимпульса, его состояние не меняется, а при подаче единичного сигнала на вход синхронизации триггер передаёт на выход сигнал, поступивший на его вход в предыдущий такт.

Легко убедиться, что D -триггер можно представить как RS -триггер, на информационные входы которого подаются сигналы: $S = CD$ и $R = C\bar{D}$.

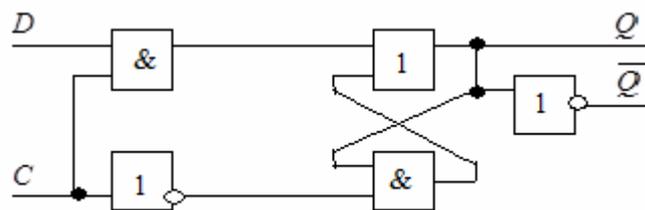


Рис.26. Синхронный D -триггер

В случае, если инверсный выход D -триггера соединяется с D -входом, он превращается в T -триггер.

JK -триггер функционирует в соответствии с таблицами состояний и возбуждений.

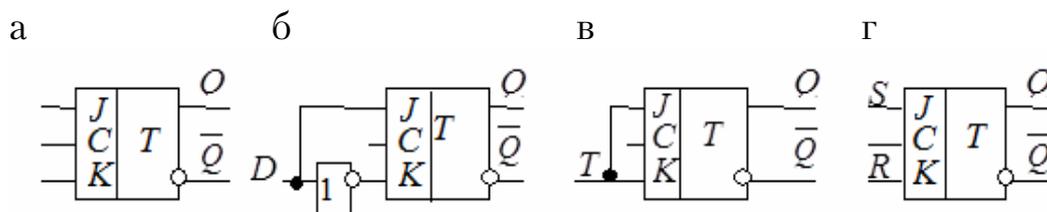


Рис. 27. JK -триггер:
 а – условное обозначение и его использование в качестве:
 б – D -триггера; в – T -триггера; г – RS -триггера

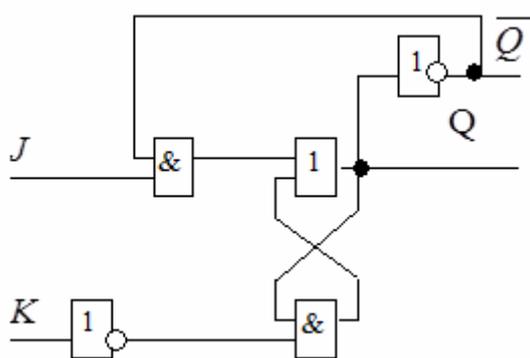


Рис. 28. Схема JK -триггера

Схема JK -триггера приведена на рис.28. В интегральной схемотехнике распространены синхронные JK -триггеры с динамическим входом синхроимпульсов C .

Триггеры путём несложной коммутации внешних выводов можно превратить в схемы, выполняющие функции RS -, D - и T -триггера. Так, если в выражении JK -триггера принять $J = D$ и $K = T$, получим

$$Q_{t+1} = D\bar{Q} + DQ = D(\bar{Q} + Q) = D, \text{ т.е. логическое выражение } D\text{-триггера.}$$

Для получения D -триггера достаточно соединить входы J и K через инвертор, причём J -вход будет выполнять роль D -входа.

T -триггер получается путём объединения J - и K -входов друг с другом и использование этого входа в качестве T -входа.

Если, на вход J подавать сигнал S , а на вход K – сигналы R , то JK -триггер будет выполнять функции RS -триггера.

Наибольшее распространение получили синхронные двухступенчатые триггеры.

4.2. Синтез пересчётных схем (счётчиков)

Счётчиком называется узел ЭВМ для подсчёта числа входных сигналов и хранения двоичного кода числа подсчитанных сигналов. Счётчики используются в ЭВМ для осуществления последовательного выполнения команд программы, подсчёта числа циклов выполненных операций, образования адресов при обращении к запоминающим устройствам и т.д. Простые счётчики описываются таблицами состоя-

ний с одним входным столбцом. В более сложных схемах используются управляющие входы для задания режима работы.

Счётчики подразделяют на суммирующие, вычитающие и реверсивные, т.е. позволяющие как прибавлять, так и вычитать очередную, пришедшую на вход единицу.

Для построения счётчиков используются триггеры одного из перечисленных типов. Их количество зависит от коэффициента пересчёта и определяется из выражения $m = \log_2 K_{сч}$, где m – ближайшее, большее целое число. Если возможное количество внутренних состояний счётчика 2^m окажется больше модуля счёта, то часть состояний ($2^m - K_{сч}$) является избыточной.

4.3. Синтез регистров

Регистром называется устройство, осуществляющее прием, хранение, некоторые логические преобразования и выдачу двойных чисел в определенном коде.

В отличие от запоминающих устройств в регистрах информация хранится не более нескольких тактов. Запоминающими элементами в регистрах служат триггеры, число которых равно числу разрядов хранимых чисел. Комбинационные схемы регистров служат для ввода и вывода хранимых чисел, преобразования их кодов, сдвига кодов на определенное число разрядов.

Различают параллельные (регистры памяти), последовательные (регистры сдвига) и параллельно-последовательные (ввод – в параллельном коде, а вывод – в последовательном и наоборот) регистры.

По способу ввода-вывода различают регистры однофазного и парафазного типов. В парафазных регистрах возможен ввод и вывод информации как в прямом, так и в инверсном коде, а в однофазных – только либо в прямом, либо инверсном коде.

В информационной технике регистры сдвига применяют для преобразования последовательного кода в параллельный и наоборот, для умножения и деления многоразрядных чисел, построения распределителей импульсов, преобразователей кодов и т.д.

Как правило, операция передачи кода с регистра объединяется с операцией приема этого кода в другой регистр.

Операция сдвига кода – это перемещение в регистре всех разрядов слова на одинаковое число разрядов влево или вправо. При сдвиге информации в ту или другую сторону возможна потеря информационных разрядов. Например, если сдвиг осуществляется вправо в сторону старших разрядов, то самый старший разряд кода, хранящийся

в крайнем справа триггере, сдвигается и пропадает. При этом в младший разряд записывается либо очередной символ, либо нуль. Особый тип регистра сдвига представляет собой кольцевой регистр, в котором выход старшего разряда соединяется со входом младшего разряда, и наоборот, в зависимости от направления сдвига.

Кольцевые регистры, например, используются для формирования циклических кодов. Наиболее удобными для построения регистров сдвига являются *D*-триггеры, поскольку в них непосредственно заложен принцип перемещения кода входного сигнала на его выход по сигналу синхронизации.

4.4. Синтез сумматоров

Как известно, все многообразие математических операций (вычитание, умножение, деление, вычисление различных функций) можно свести к единственной операции сложения прямых и обратных сдвинутых влево или вправо на то или иное число разрядов кодов чисел. Поэтому одним из основных узлов арифметических устройств ЭВМ является сумматор. Сумматоры многоразрядных чисел составляются из одноразрядных сумматоров, выполняющих операцию сложения двух или трех одноразрядных чисел.

Одним из разновидностей одноразрядных сумматоров являются полусумматор и сумматор по модулю два, которые представляют собой устройства только с двумя входами (*a* и *b*), на выходе у которых появляется сигнал высокого уровня (1) только при противоположных уровнях входных сигналов. Полусумматор отличается от сумматора по модулю два наличием дополнительного выхода *P* переноса, сигнал высокого уровня на котором появляется только при наличии единичных сигналов на обоих его выходах.

Полный сумматор представляет собой устройство для сложения трех одноразрядных чисел *a*, *b* и *c*, где *c* – сигнал переноса из предыдущего младшего разряда. На выходе полного сумматора формируются два сигнала: *S* – сумма по модулю два и *P* – сигнал переноса в старший разряд. Обозначения всех типов сумматоров приведены на рис. 29 (а, б, в).

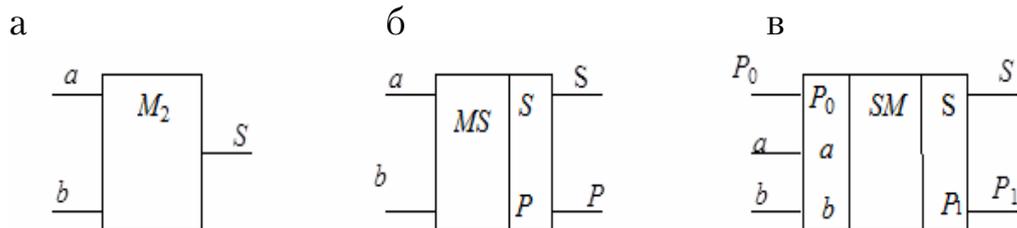


Рис. 29. Условное обозначение сумматоров:
а – сумматор по модулю два; б – полусумматор; в – полный сумматор

Полусумматор может быть построен на основе сумматора по модулю два с добавлением схемы И (рис. 30,а), а полный сумматор – на основе двух полусумматоров и схемы ИЛИ (рис. 30,б).

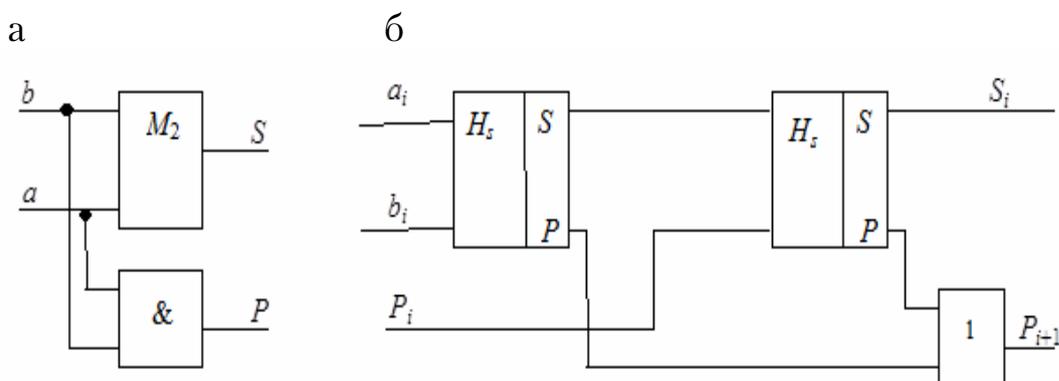


Рис. 30. Одноразрядный полусумматор:
а – на основе сумматора по модулю 2; б – на основе полусумматора

Достоинством последовательных сумматоров является простота схемы, требующей минимального количества оборудования, а недостаток – низкое быстродействие, так как для сложения кодов n -разрядных чисел требуется $(n+1)$ такт работы устройства.

Длительность формирования результата такого сумматора определяется временем установления выходных сигналов (как суммы, так и переноса) в каждом из одноразрядных сумматоров после установления сигнала на его входах.

4.5. Синтез дешифраторов, шифраторов и преобразователей кодов

Дешифратор (decoder) – это комбинационное устройство, позволяющее распознавать числа, представленные позиционным n -разрядным кодом. Рассмотрим дешифратор двоичного кода с n входами. Так как возможное количество чисел, закодированных n -разрядным двоичным кодом, равно количеству наборов из n аргументов ($P=2^n$), то наибольшее число выходов дешифратора равно 2^n . Такой дешифратор называется полным. Если число выходов $S < P$, дешифратор называют неполным, так как $m=P-S$ наборов являются запрещенными.

Достоинство линейных дешифраторов – их высокое быстродействие, недостаток – большие аппаратные затраты. Более экономичными с точки зрения аппаратных затрат являются каскадные дешифраторы. Шифратор (coder) – это комбинационное устройство, выполняющее функции, обратные дешифратору (преобразование унитарного кода в двоичный). Унитарный вход часто получается при

преобразовании аналоговой информации в цифровую форму. Если число входов шифратора 2^n , то число выходов соответственно равно n .

Различают три типа преобразователей.

В преобразователях первого типа элементы входного кода соответствуют элементам выходного кода. К этому типу относятся параллельно-последовательные регистры, преобразующие параллельный код в последовательный и обратно.

В преобразователях второго типа выходной код содержит, кроме всех элементов входного кода, еще и дополнительные элементы. Например, устройство проверки входного кода на четность с добавлением дополнительного проверочного разряда в выходной код.

К третьему типу преобразователей относятся преобразователи, у которых элементы выходного кода не совпадают с элементами входного кода. Чаще всего преобразователи кодов строятся на основе таблиц истинности в виде комбинационной схемы. Такие преобразователи обладают высоким быстродействием, однако сложны в аппаратной реализации.

4.6. Синтез распределителей и мультиплексоров

Распределитель – это устройство, передающее импульс, поступивший на его вход X , на один из S выходов в зависимости от управляющего сигнала YC , заданного двоичным кодом (рис.31).

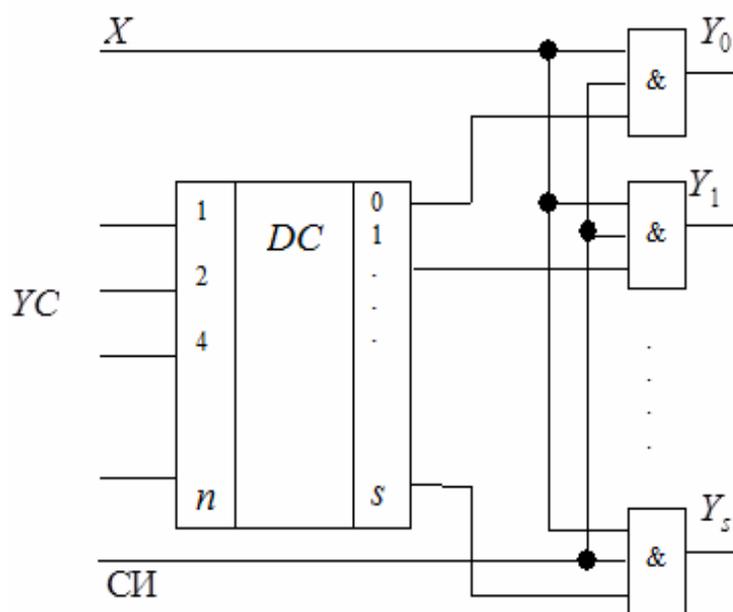


Рис. 31. Распределитель

Управляющий сигнал в виде параллельного кода попадает на вход дешифратора DC , выходы которого соединены с одним из трех входов конъюнкторов, составляющих выходную часть схемы. На второй вход всех конъюнкторов поступает входной сигнал X , а на третий – импульс синхронизации СИ.

Входной сигнал X пройдет только на один из выходов устройства в зависимости от кода YC при наличии импульса синхронизации СИ.

Мультиплексор – это устройство, подключающее единственную общую выходную шину к одному из входов в зависимости от управляющего сигнала, заданного в двоичном коде. Разрядность n управляющего сигнала определяет количество входов (2^n), с которых мультиплексор позволяет осуществлять прием информации. На рис.32 приведена схема мультиплексора на четыре входа D_0, \dots, D_3 , управляемая двухразрядным кодом X_0X_1 и синхронизируемая тактовыми импульсами СИ.

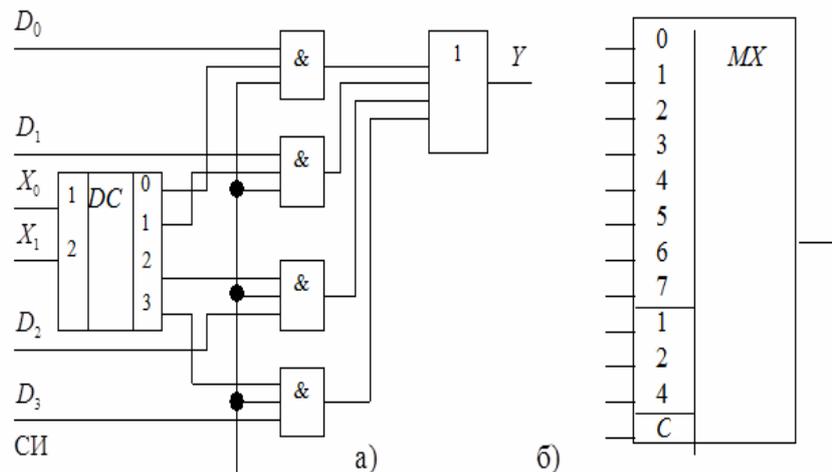


Рис. 32. Мультиплексор:
а – функциональная схема; б – условное обозначение

В случае необходимости построения мультиплексора с большим числом входов используются каскадные схемы включения мультиплексоров. Так, например, на основе использования четырехходовых мультиплексоров первого каскада и одного четырехходового мультиплексора второго каскада легко построить мультиплексор на 16 входов.

Замечательной особенностью мультиплексоров является возможность построения практически любой булевой функции путем суперпозиций.

5. ПРИНЦИПЫ МОДЕЛИРОВАНИЯ ЭЛЕКТРОННЫХ СХЕМ В СРЕДЕ WORKBENCH

Система **Electronics Workbench** представляет собой комплекс программ, функционирующих на **ПЭВМ IBM PC** под управлением **WINDOWS**, и предназначена для проектирования аналоговых и цифровых электронных схем с визуализацией исходных данных и результатов проводимых анализов.

Процесс проектирования и анализа происходит во взаимодействии разработчика с диалоговым интерфейсом программы. Объекты моделирования имитируют свои реальные прототипы как по внешнему виду, так и по способу манипуляции их органами управления. Так, создаваемая с помощью встроенного графического редактора принципиальная схема собираемого из радиодеталей устройства уже является достаточной информацией для ее моделирования. Номиналы и типы радиоэлементов (компонентов схемы) выбираются в процессе создания схемы и автоматически проставляются рядом с их условными графическими обозначениями (**УГО**). Для визуализации результатов анализа схемы на монтажном столе – рабочем поле экрана – размещаются нужные виртуальные (не существующие реально) измерительные приборы: осциллограф, вольтметр и т.д. Виртуальные приборы воспроизводят свои реальные прототипы.

В результате моделирование как таковое скрыто от пользователя программы, а процесс создания схемы сводится к привычной для лабораторной практики сборке испытываемого макета схемы, подключению и настройке приборов и, наконец, проведению измерений. Отличие состоит в том, что работа выполняется не на физических, а на виртуальных объектах на экране монитора **ПЭВМ**. Таким образом, экран представляет собой как бы поверхность монтажного стола с размещенными на ней макетом и набором подключенных к нему измерительных приборов.

5.1. Компьютерный монтажный стол. Принципы компьютерного макетирования

После запуска программы **Electronics Workbench WEWB32** на экране появляется изображение монтажного стола (рис.33).

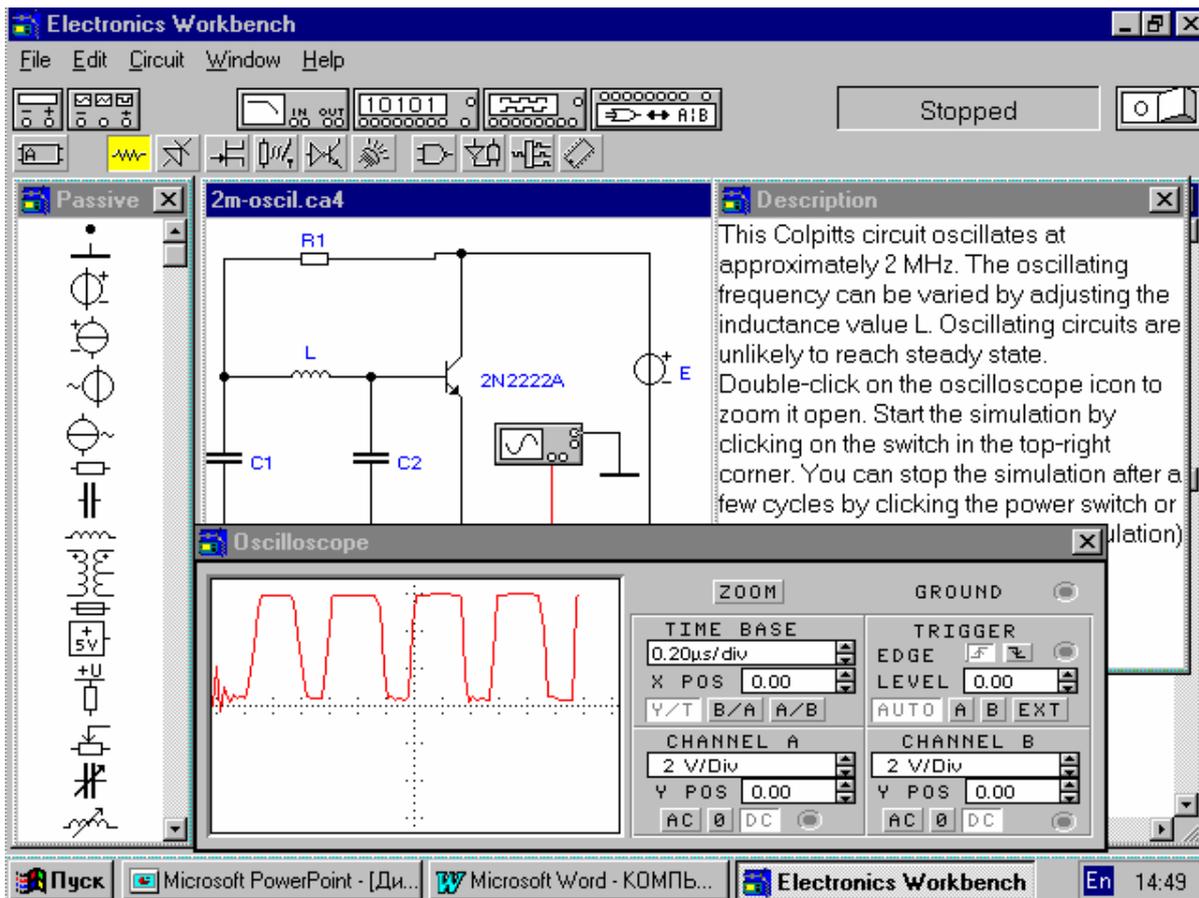


Рис.33. Общий вид монтажного стола Electronics Workbench

Монтажный стол содержит все, что может понадобиться для построения макета схемы и его испытания. Основные составные части стола:

- рабочее поле – большая центральная часть экрана, на которой размещаются компоненты схемы и измерительные приборы;
- магазин компонентов, расположенный сбоку от рабочего поля;
- разделы магазина компонентов;
- панель измерительных приборов, расположенная над рабочим полем, с изображениями имеющихся в наличии приборов;
- строка меню в верхней части экрана, предназначенная для выбора операций со схемой (загрузка, сохранение, удаление, правка и т.д.);
- клавиша включения питания в правом верхнем углу экрана, которой запускается процесс моделирования.

Рабочее поле монтажного стола используется для сборки макета схемы и подключения приборов. Рабочее поле представляет собой окно, которое можно перемещать, изменять его размеры и просматривать подобно другим окнам операционной среды **WINDOWS**.

5.2. Магазин компонентов

Магазин компонентов появляется после запуска программы в виде вертикальной полосы с условными графическими обозначениями (**УГО**) компонентов электрических схем. Кроме того, непосредственно над рабочим полем монтажного стола появляются клавиши с пиктограммами разделов магазина, один из которых является активным. Активизация раздела осуществляется щелчком на клавише. Всего предусмотрено одиннадцать разделов:

1) заказные компоненты (**Customs**), которые могут формироваться пользователем, например, путем создания подсхем (функциональные узлы, состоящие из компонентов других разделов и изображаемые в виде единого блока);

2) пассивные компоненты (**Passive**) (резисторы, конденсаторы, катушки индуктивности и пр.);

3) активные компоненты (**Active**) (диоды, транзисторы, операционные усилители и пр.);

4) полевые транзисторы (**FETs**);

5) управляемые компоненты (**Control**), значения параметров которых можно изменять в процессе моделирования (потенциометры, управляемые переключатели и пр.);

6) гибридные микросхемы (**Hybrid**), главным образом цифро-аналоговые и аналого-цифровые преобразователи;

7) элементы индикации (**Indic**);

8) логические элементы (**Gates**) цифровых схем;

9) комбинационные цифровые узлы (**Comb'l**) (мультиплексоры, дешифраторы и пр.);

10) последовательностные цифровые узлы (**Seq'l**) (триггеры, счетчики, регистры и пр.);

11) микросхемы (**IC**). Данный раздел позволяет выбрать конкретные микросхемы из типовых серий микросхем.

Раздел *Passive* содержит главным образом пассивные компоненты электронных схем. Основные представители раздела:

- *соединитель* в виде круглой точки используется для соединения проводников и создания контрольных точек в схеме. Соединитель появляется автоматически при выполнении соединения, когда перемещаемый конец линии связи совмещается с изображением уже имеющегося проводника. Соединитель можно также перетащить мышью из магазина компонентов на рабочее поле. Каждый соединитель обеспечивает подключение до четырех проводников, по одному с каждой из ортогонально ориентированных сторон;

- *источники* постоянных и переменных тока и напряжения. Параметры выходных сигналов источников назначаются после двойного щелчка на их изображении. При установке параметров источников переменного тока и напряжения назначаются среднеквадратические значения тока и напряжения I и U , связанные с амплитудными значениями I_a , U_a отношением

$$U = \frac{U_a}{\sqrt{2}}$$

Следует иметь в виду, что все источники в схеме, включая внешний измерительный генератор, должны иметь одну и ту же частоту. Поэтому назначение частоты одного источника приводит к автоматической установке частоты остальных источников;

- *резистор, конденсатор и катушка индуктивности* являются наиболее распространенными пассивными компонентами схем. Сопротивление резистора (Ом), емкость конденсатора (Ф) и индуктивность катушки (Гн) назначаются после двойного щелчка на их изображениях;

- *трансформатор* представляет собой пару связанных между собой общим магнитопроводом катушек индуктивности. Коэффициент n трансформации напряжения характеризует отношение амплитуд переменных напряжений в выходной и входной обмотках и назначается после двойного щелчка на изображении трансформатора. Направления намотки провода на катушках трансформатора отмечаются точками рядом с началом обмотки. Если обмотки трансформатора ориентированы согласовано, то его коэффициент трансформации положителен, в противном случае коэффициент трансформации отрицателен;

- *плавкий предохранитель* предназначен для защиты схемы, как правило, по входу, выходу и цепи питания, от токовой перегрузки.

Раздел *Passive* содержит также стандартный источник питания 5 В для питания цифровых микросхем и переменные резистор, конденсатор и катушку индуктивности.

Раздел *Active* включает распространенные полупроводниковые приборы, используемые для построения аналоговых схем:

- *диод*, представляющий собой *p-n*-переход, пропускающий ток только в одном направлении. По умолчанию предполагается идеальный диод, однако после двойного щелчка можно назначить модель реального диода из числа имеющихся моделей в библиотеке программы;

- *стабилитрон*, или диод Зенера, использующий участок стабильного напряжения на обратной ветви вольт-амперной характеристики

диода. Напряжение стабилизации можно назначить в пределах от 2,4 до 200 В или же выбрать модель реального стабилитрона;

- *светодиод*, который применяется для излучения света при протекании через него тока. Для идеального светодиода назначается пороговый ток, после превышения которого начинается свечение светодиода. С помощью диалогового окна можно также задать модель реального светодиода;

- *биполярные транзисторы n-p-n и p-n-p* типов. По умолчанию программа предлагает модели идеальных транзисторов, выбор модели реального транзистора из числа имеющихся в библиотеке осуществляется посредством диалогового окна после двойного щелчка на его изображении;

- *операционные усилители* – это усилители с гальванической связью выхода со входами. В магазине компонентов содержатся два вида моделей операционного усилителя, вторая из которых учитывает цепи его питания. Входным сигналом операционного усилителя служит разность его входных напряжений, т.е.

$$U_{\text{вых}} = k \times (U_+ - U_-) \dots$$

По умолчанию программа предлагает модель идеального операционного усилителя, однако посредством диалогового окна можно назначить также модель реального прибора из числа имеющихся в библиотеке. Основными параметрами модели, которые можно корректировать, являются: коэффициент усиления, напряжение смещения нуля, разность и абсолютные значения входных токов, частота единичного усиления (частота входного синусоидального напряжения усилителя без обратной связи, при которой коэффициент усиления уменьшается до единицы).

Раздел *FET* включает модели полевых транзисторов с каналами n- и p-типов с управляющим p-n переходом и с изолированным затвором. Конкретный тип полевого транзистора назначается посредством диалогового окна. По умолчанию программа предлагает модели идеальных транзисторов.

Раздел *Control* содержит модели управляемых компонентов:

- *механические переключатели* на два направления, способные переключаться нажатием клавиши на клавиатуре ПЭВМ. Клавиша, управляющая переключателем, назначается с помощью вызываемого двойным щелчком диалогового окна;

- *однополюсные переключатели, управляемые напряжением или током*, могут использоваться для сопряжения аналоговых схем с цифровыми схемами, а также для управления исполнительными

органами электромеханических систем. Для данных переключателей посредством диалогового окна назначаются пороги включения (V_{on} или I_{on}) и выключения (V_{off} или I_{off}). Порог включения определяет условие замыкания переключателя, а порог выключения – условие его размыкания. Порог включения выбирается, как правило, выше порога выключения, что обеспечивает гистерезисную передаточную характеристику и надежность работы переключателя;

➤ *источники напряжения и тока, управляемые напряжением или током*, позволяют построить макромодели сложных электронных узлов с привлечением минимального числа компонентов их эквивалентных схем. Посредством диалогового окна для каждого из четырех разновидностей управляемых источников задается коэффициент пропорциональности значений выходного и входного параметров:

- для источника тока, управляемого током, назначается коэффициент усиления по току

$$F = \frac{I_{\text{ВЫХ}}}{I_{\text{ВХ}}} \left[\frac{\text{А}}{\text{А}} \right];$$

- для источника тока, управляемого напряжением, задается крутизна

$$G = \frac{I_{\text{ВЫХ}}}{V_{\text{ВХ}}} \left[\frac{\text{А}}{\text{В}} \right];$$

- для источника напряжения, управляемого током, устанавливается трансрезистивный коэффициент

$$H = \frac{V_{\text{ВЫХ}}}{I_{\text{ВХ}}} \left[\frac{\text{В}}{\text{А}} \right];$$

- источник напряжения, управляемый напряжением, характеризуется коэффициентом усиления по напряжению

$$E = \frac{V_{\text{ВЫХ}}}{V_{\text{ВХ}}} \left[\frac{\text{В}}{\text{В}} \right]$$

Коэффициенты F и E – безразмерные, G имеет размерность **сименс**, а H – **Ом**;

- **источник тока, управляемый полиномом**, позволяет создать источник со сложной функциональной зависимостью от нескольких аргументов.

Остальные разделы магазина компонентов содержат элементы и узлы цифровых схем, а также аналого-цифровые и цифроаналоговые преобразователи для сопряжения аналоговых и цифровых схем.

5.3. Панель приборов

Панель измерительных приборов содержит семь позиций (рис.34):

- мультиметр (multimeter);
- функциональный генератор (functional generator);
- осциллограф (oscilloscope);
- частотный характериограф (Bode plotter);
- генератор двоичных слов (word generator);
- логический анализатор (logic analyzer);
- логический преобразователь (logic converter).

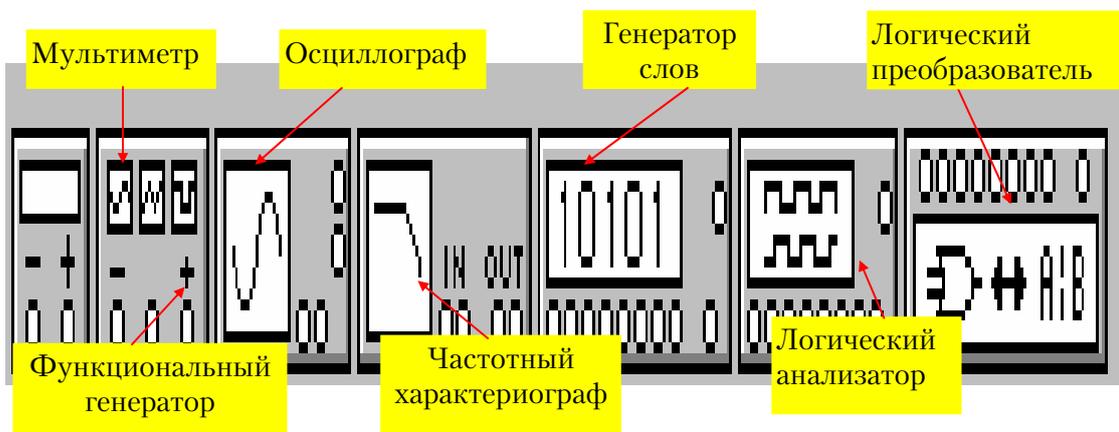


Рис. 34. Панель приборов Electronics Workbench

Кроме того, имеются вольтметры и амперметры, расположенные в магазине компонентов.

Мультиметр предназначен для измерения напряжения, тока, сопротивления или ослабления сигнала в децибелах между двумя контрольными точками схемы.

Функциональный генератор является источником напряжения, который вырабатывает аналоговый сигнал синусоидальной, прямоугольной или треугольной формы. Органами управления этого прибора можно настроить частоту, коэффициент заполнения (отношение длительности сигнала к периоду), амплитуду и постоянную составляющую выходного напряжения.

Осциллограф используется для визуального наблюдения электрических сигналов в функции времени, измерения их амплитудных и временных параметров – длительности, времени задержки, периода повторения и пр. Осциллограф снабжен двумя входными зажимами (каналы А и В) и позволяет наблюдать одновременно два сигнала в двух различных узлах схемы.

Частотный характерограф позволяет измерять амплитудно- и фазочастотные характеристики (**АЧХ** и **ФЧХ**) схемы (получать **диаграммы Боде**). **АЧХ** представляет собой отношение напряжений в двух контрольных точках в функции изменяющейся частоты, причем отношение напряжений выражается в логарифмическом масштабе (в децибелах). **ФЧХ** представляет собой зависимость от частоты разности фаз двух напряжений в градусах.

Генератор слов применяется для исследования цифровых схем. Он вырабатывает восемь последовательностей цифровых двоичных сигналов, периодически повторяющихся каждые шестнадцать тактов, а также тактовые импульсы, которые можно использовать для синхронизации проверяемой схемы. Информационное содержание последовательностей можно задавать по желанию пользователя.

Логический анализатор служит для развертки во времени восьми цифровых сигналов при исследовании цифровых схем. Этот прибор подобен 8-канальному осциллографу для одновременного наблюдения цифровых сигналов в восьми различных узлах схемы.

Логический преобразователь, входящий в состав панели приборов, в зависимости от назначенного его органами управления режима способен преобразовывать логические функции из одной формы представления в другую и синтезировать схемы комбинационной логики по заданной булевой функции.

5.4. Опорное меню

Строка опорного меню Electronics Workbench расположена в верхнем левом углу монтажного стола. Опорное меню включает пункты:

| | |
|-----------------|----------------|
| <u>F</u> ile | Файл |
| <u>E</u> dit | Редактирование |
| <u>C</u> ircuit | Схема |
| <u>W</u> indow | Окно |
| <u>H</u> elp | Помощь |

Выбрать пункт из меню можно либо щелчком на его имени, либо одновременным нажатием клавиши ALT и подчеркнутого символа в имени пункта (например, чтобы открыть пункт Circuit, следует ввести ALT+C). Выбор пункта приводит к появлению бокса команд этого пункта, выбор команды осуществляется аналогично. *Если имя команды тусклое, то такая команда недоступна.*

Пункт File включает команды для работы с файлами:

| | |
|-------------------|--------------------|
| New (Ctrl+N) | Создать новый файл |
| Open ... (Ctrl+O) | Открыть файл ... |
| Save (Ctrl+S) | Сохранить |

| | |
|---------------------|----------------------|
| Save As ... | Сохранить как ... |
| Revert to Saved ... | Восстановить |
| Print ... (Ctrl+P) | Печать ... |
| Print Setup ... | Настройка печати ... |
| Exit (Alt+F4) | Выход |

Перечисленные команды являются обычными командами WINDOWS. Команда Print Setup дает дополнительную возможность назначения альтернативного принтера, не указанного в панели управления WINDOWS, а также выбора ориентации печатаемого изображения, размера бумаги и насыщенности распечатки с помощью диалогового окна.

Пункт Edit содержит команды, применяемые для редактирования схемы:

| | |
|---------------------|----------------------------|
| Cut (Ctrl+X) | Вырезать |
| Copy (Ctrl+C) | Копировать |
| Paste (Ctrl+Y) | Вставить |
| Delete (DEL) | Удалить |
| Select All (Ctrl+A) | Выделить все |
| Copybits (Ctrl+I) | Скопировать в буфер |
| Show Clipboard | Показать буфер изображения |

Команда Cut удаляет выделенные компоненты или текст в буфер. Выделение компонентов выполняется щелчком левой клавишей мыши после вывода указателя на изображение компонента. Выделение более одного компонента в схеме достигается выводом указателя в точку, расположенную выше и левее верхнего левого компонента в группе, и перемещением его по диагонали вниз и вправо при нажатой клавише мыши. По мере перетаскивания указателя появляется и увеличивается в размерах прямоугольник, который должен охватить все объекты выделяемой группы, после чего клавиша мыши отпускается. Командой Cut выделенные компоненты удаляются в буфер вырезанного изображения (Clipboard), откуда могут быть вставлены в другую схему командой Paste. Команда Copy работает аналогично Cut, однако выделенный фрагмент схемы не удаляется, а копируется в буфер.

Пункт Circuit открывает команды, с помощью которых выполняется построение схемы и задаются директивы моделирования:

| | |
|--------------------|----------------------------|
| Activate (Ctrl+G) | Запуск (включение питания) |
| Stop (Ctrl+T) | Стоп |
| Pause/Resume (F9) | Пауза/Продолжение |
| Label ... (Ctrl+L) | Обозначение |
| Value ... (Ctrl+U) | Значение параметра |
| Model ... (Ctrl+M) | Модель |

| | |
|-------------------------------|---------------------------|
| Zoom (Ctrl+Z) | Трансфокация (увеличение) |
| Rotate (Ctrl+R) | Вращение |
| Fault ... (Ctrl+F) | Дефект |
| Subcircuit ... (Ctrl+B) | Подсхема |
| Wire color ... | Цвет провода |
| Preferences ... (Ctrl+E) | Предустановки |
| Analysis Options ... (Ctrl+Y) | Директивы анализа |
| Restrictions | Ограничения |

Использование большинства из этих команд рассматривается в разд. 2, поэтому ниже дается лишь краткая их расшифровка.

Команда **Activate** запускает программу моделирования, «включая» питание схемы. Запуск можно также осуществить щелчком на выключателе питания в верхнем правом углу монтажного стола.

Команда **Stop** используется для останова моделирования. К тому же результату приводит щелчок на выключателе питания.

Команда **Pause/Resume** предназначена для временного прерывания моделирования и его возобновления с точки останова.

Команда **Label** применяется для позиционного обозначения предварительно выделенного компонента посредством диалогового окна. После введения позиционного обозначения диалоговое окно закрывается щелчком на клавише **ОК**.

Команда **Value** предназначена для указания значения параметра, предварительно выделенного щелчком на его изображении компонента с помощью появляющегося диалогового окна. После того, как значение будет напечатано, необходимо щелкнуть на клавише **Accept**.

Команда **Model** устанавливает тип модели компонента посредством диалогового окна. Обычно тип компонента выбирается из числа имеющихся в библиотеке моделей, однако имеется возможность создания собственных моделей.

Команда **Zoom** предназначена для увеличения размеров изображения предварительно выделенного прибора или раскрытия структуры подсхемы. Команде **Zoom** соответствует также двойной щелчок на изображении прибора.

С помощью команды **Rotate** можно изменять ориентацию компонента, поворачивая его изображение на 90° по часовой стрелке. Вольтметр и амперметр этой командой не вращаются, однако изменяется ориентация их выводов.

Команда **Fault** создает дефектный компонент для обучения поиску неисправностей в схеме. После выбора этой команды следует указать посредством диалогового окна характер дефекта: **Open circuit**

(холостой ход – разрыв цепи) или Short circuit (короткое замыкание). Устранение дефекта достигается выбором режима None (Нет) из меню данной команды.

Команда Subcircuit предназначена для объединения части собранной схемы в самостоятельный функциональный узел, подобный интегральной микросхеме, который будет размещен в разделе Customs магазина компонентов.

Для создания подсхемы сначала необходимо собрать ее из имеющихся компонентов на рабочем поле монтажного стола, выделить компоненты подсхемы, а затем выбрать команду Subcircuit. В появившемся диалоговом окне печатается имя подсхемы и одновременно указывается режим копирования компонентов в подсхему (с изменением или сохранением взаимного расположения компонентов), после чего изображение подсхемы появляется в разделе Customs магазина, а изображения входящих в подсхему компонентов на рабочем поле заменяются изображением подсхемы. При этом все внешние подключения компонентов подсхемы становятся выводами подсхемы.

В дальнейшем подсхема может корректироваться после двойного щелчка на ее УГО совершенно аналогично редактированию основной схемы на рабочем поле. Новые выводы подсхемы образуются при перетаскивании проводника от внутреннего компонента подсхемы к краю подсхемы, при этом клавиша мыши отпускается после появления на краю подсхемы маленького кружочка.

Команда Wire color устанавливает цвет предварительно выделенного проводника.

Команда Preferences позволяет настроить рабочее поле монтажного стола и внешний вид схемы. Данная команда предусматривает следующие директивы:

Show grid – щелчок на этой опции показывает точечную координатную сетку рабочего поля;

Use grid – щелчок на этой опции включает использование координатной сетки для выравнивания по ней выводов компонентов и проводников;

Show labels – включает отображение на рабочем поле позиционных обозначений компонентов, которые появляются рядом с их УГО;

Show values и Show models применяются для отображения на рабочем поле значений параметров и типов моделей компонентов. Следует отметить, что данные опции не выполняются, если в наборе команд Restrictions пункта Circuit назначено «Hide values and models restrictions» («Скрыть значения параметров и модели»).

Пункт Analysis options включает команды, устанавливающие виды выполняемых анализов, точность расчетов и особенности отображения результатов.

Пункт Restrictions содержит команды ограничения:

Hide values fnd models – скрыть значения параметров и типы моделей компонентов. Данный прием может использоваться для постановки заданий на идентификацию скрытых величин путем тестирования схемы;

Hide subcircuits – скрыть все подсхемы. Данная команда превращает подсхемы в «черные ящики» с неизвестной структурой, восстановление которой может составить суть задания студенту;

Hide parts bin – скрыть магазин компонентов – ограничивает доступ к магазину компонентов, предлагая студенту для выполнения задания ограничиться набором компонентов, уже имеющимся на рабочем поле;

Hide unused instruments – скрыть неиспользуемые приборы. Команда не позволяет доставать с панели приборов дополнительные объекты и предлагает ограничиться при проведении измерений теми из них, которые уже размещены на рабочем поле;

Hide faults – скрыть дефекты. Команда отменяет выделение ранее «испорченного» компонента, после чего внешне этот компонент не будет отличаться от исправных;

Password – пароль. Команда предотвращает несанкционированную отмену ограничений кем-либо, кроме автора ограничений. После первого же сохранения схемы доступ к диалоговому окну Restrictions становится возможным только посредством ввода пароля.

Пункт Window опорного меню предлагает команды управления окнами программы Electronics Workbench:

| | |
|-------------------------------|---------------------------|
| Arrange (Ctrl+W) | Упорядочивание |
| <u>1</u> Circuit | Схема |
| <u>3</u> Description (Ctrl+D) | Описание |
| Custom (Subcircuit) | Заказная (Подсхема) |
| Passive | Пассивные |
| Active | Активные |
| Field Effect Transistors | Полевые транзисторы |
| Control | Управляемые компоненты |
| Hybrid | Гибридные схемы |
| Indicators | Индикаторы |
| Gates | Логические элементы |
| Combinational | Комбинационные узлы |
| Sequential | Последовательностные узлы |
| Integrated Circuits | Интегральные схемы |

Команда Arrange позволяет аккуратно расположить на рабочем поле монтажного стола открытые окна программы (раскрытые приборы, описание). Для этого окна перемещаются примерно туда, где их хотелось бы видеть, после чего выбирается команда Arrange.

Команда Circuit выводит рабочее поле на передний план.

Команда Description позволяет разместить текстовый комментарий к схеме на английском языке в окне описания.

Остальные команды этого раздела относятся к разделам магазина компонентов. Выбор любой из них активизирует соответствующий раздел.

Пункт Help предоставляет быструю контекстно-зависимую справку по использованию программы.

5.5. Размещение и обозначение компонентов

Манипуляции с виртуальными компонентами осуществляются с помощью мыши и клавиатуры. Количество однотипных компонентов, извлекаемых из магазина компонентов, не ограничено.

Для размещения на рабочем поле монтажного стола нужного компонента указатель подводится мышью к пиктограмме соответствующего раздела магазина компонентов и выполняется щелчок левой клавишей мыши. Это приводит к активизации раздела и появлению его содержимого в магазине компонентов слева от рабочего поля. После этого указатель подводится к **УГО** нужного компонента и нажимается левая клавиша мыши. Далее, не отпуская клавиши, **УГО** компонента переносят на рабочее поле, где клавиша отпускается, когда компонент окажется в желательной позиции. Каждый вновь введенный компонент выделяется красным цветом. Для выделения более одного компонента при нажатой клавише **SHIFT** на клавиатуре выполняется щелчок на всех выделяемых компонентах. К такому же результату приводит щелчок правой клавишей мыши без использования клавиатуры. Для выделения группы объектов можно также, установив указатель слева и выше самого левого объекта в группе, переместить указатель по диагонали вправо и вниз. По мере перемещения появится расширяющийся прямоугольник. Когда прямоугольник захватит все объекты группы, левую клавишу мыши можно отпустить. Выделение компонентов используется для указания их типономиналов, удаления, перемещения.

После размещения компонента в желаемой позиции его изображение можно ориентировать с помощью команды вращения для того, чтобы получить читаемую конфигурацию схемы без излишних пересечений проводников. Вращение выделенного объекта или группы

компонентов на 90° выполняется командой **Rotate** (Вращение) из меню **Circuit** (Схема). Этой же команде соответствует нажатие двух клавиш – **Ctrl** и **R** (**Ctrl+R**).

Для указания позиционного обозначения и типономинала выделенного компонента используются команды **Model** (Модель), **Label** (Обозначение) и **Value** (Значение) из меню **Circuit**. Командам **Model** и **Value** эквивалентны двойные щелчки на изображении выделенного компонента или команды с клавиатуры **Ctrl+M** и **Ctrl+U** соответственно. Если обозначения компонентов до указанных команд не были заданы, то сначала появляется диалоговое окно **Label**. Для компонентов, имеющих сложные модели (например для транзисторов), по умолчанию устанавливаются идеальные модели. Замена модели компонента производится с помощью диалогового окна моделей, которое появляется после команды **Model** (**Ctrl+M**). Диалоговое окно предлагает перечень моделей компонентов, из которого выбирается нужный и нажимается клавиша **OK**.

После назначения типономиналов компонентов рядом с их **УГО** на рабочем поле монтажного стола появляются их позиционные обозначения и номиналы (или типы для активных компонентов). Если эти данные на рабочем поле не обязательны, то их можно скрыть командой «**Hide values and models**» («Скрытие значений и моделей») в диалоговом окне **Restrictions** (Ограничения) из меню **Circuit**.

Выделенный компонент можно удалить с рабочего поля, или нажав клавишу **DEL**, или перетащив **УГО** компонента назад в магазин компонентов.

5.6. Выполнение межсоединений

Когда все нужные компоненты окажутся на рабочем поле монтажного стола, их выводы необходимо соединить друг с другом в соответствии с принципиальной электрической схемой. Соединение выводов двух компонентов выполняется указателем с помощью мыши. Для этого указатель подводится к выводу одного компонента (при этом вывод выделится), нажимается левая клавиша мыши и при нажатой клавише указатель подводится по кратчайшему пути к нужному выводу другого компонента. По мере перемещения указателя за ним следует линия подводимого проводника. Когда указатель окажется на нужном выводе, последний также выделится. Тогда клавиша отпускается, и линия связи автоматически трассируется между двумя выделенными выводами компонентов. При необходимости, например если линии располагаются слишком близко друг к другу или проходят над

УГО компонентов, проводник можно переместить, указав проводник щелчком мыши и перетащив его после появления на месте указателя двунаправленной стрелки.

Для соединения двух проводников и создания контрольных точек в схеме используется **УГО** соединителя (точки), представляющее собой зачерненный кружок в разделе пассивных компонентов. Соединитель имеет четыре ортогонально ориентированных вывода, которые можно использовать для подключения четырех проводников, по одному с каждой стороны. Соединители создаются автоматически при перетаскивании проводника от какого-либо вывода компонента к данному проводнику. Клавиша мыши отпускается после появления на изображении проводника кружочка в точке подключения.

Проводники можно выполнить линиями разного цвета командой **Wire color** (Цвет проводника) из меню **Circuit** либо выбором цвета после двойного щелчка на изображении проводника. Если данным проводником к контрольной точке подключается вход виртуального осциллографа или логического анализатора, то этим же цветом будет окрашена соответствующая сигналограмма на экране прибора.

Если необходимо удалить проводник, то следует указать вывод компонента, куда подключен этот проводник, так, чтобы этот вывод выделился, и, нажав левую клавишу мыши, оттащить проводник от вывода и отпустить клавишу. Если убрать компонент или прибор с рабочего поля в магазин компонентов или на панель приборов, то все линии связи, подключенные к этому объекту, также исчезнут.

Линия связи может получиться неровной из-за неточного взаимного позиционирования компонентов и приборов по горизонтали и вертикали. Для выравнивания линий связи можно выделить объект и скорректировать его позицию с помощью мыши.

Один из узлов схемы, потенциал которого принимается равным нулю, должен быть заземлен путем подключения к нему **УГО Ground** (Земля – ) из раздела пассивных компонентов магазина.

5.7. Подключение и настройка измерительных приборов

Измерительные приборы, точнее, их пиктограммы хранятся на панели приборов выше рабочего поля монтажного стола. Для подключения нужного прибора его пиктограмма с помощью мыши переносится на рабочее поле. Зажимы прибора подключаются к контрольным точкам схемы аналогично остальным соединениям в схеме. Каждый из приборов имеется на панели в единственном экземпляре; поэтому при переносе его на рабочее поле место на панели остается свободным.

Кроме того, в магазине компонентов имеются вольтметры и амперметры, количество которых не ограничено.

После подключения каждого прибора выполняется его настройка под конкретный режим анализа. С этой целью изображение прибора увеличивают путем выделения прибора и выполнения команды **Zoom** (Лупа) из меню **Circuit**. К тому же результату приводит двойной щелчок на изображении прибора. На увеличенном изображении прибора устанавливают в необходимые положения органы его управления. Все приборы, перечисленные в разд. 1, имеют собственные органы управления, однако для их настройки используются одни и те же принципы:

1) для выбора органа управления, выполненного в виде клавиши или кнопки, и его установки используется щелчок мышью после вывода указателя на клавишу или кнопку;

2) чтобы изменить значение и единицу размерности вырабатываемой или измеряемой величины, производится щелчок на стрелках «вверх» и «вниз», расположенных сбоку от обозначения единицы размерности или ее значения. Иногда можно щелкнуть в свободном поле и нажать стрелочные клавиши «вверх» или «вниз» на клавиатуре для изменения значения величины. На некоторых приборах можно просто напечатать новое значение величины.



Рис. 35. Мультиметр

Мультиметр (рис. 35) применяется для измерения напряжения, тока, сопротивления или ослабления сигнала в децибелах между двумя контрольными узлами схемы. Режим работы мультиметра задается клавишами. Выбор клавиши осуществляется щелчком мыши на клавише верхнего ряда:

A – измерение тока (режим амперметра). Для измерения тока мультиметр должен быть включен своими зажимами «+» и «-» в разрыв цепи, где измеряется ток. Щелчком на клавишах «~» или «-» задается режим измерения либо переменного (его среднего значения), либо постоянного тока. По умолчанию мультиметр в режиме амперметра обладает внутренним сопротивлением 1 МОм (10^3 Ом), однако сопротивление можно изменить щелчком на клавише **SETTINGS** (Установки) мультиметра. Следует иметь в виду, что включение низкоомного прибора в высокоомную схему может привести к снижению математической точности моделирования;

V – измерение напряжения (режим вольтметра). Для измерения напряжения между двумя точками схемы к ним необходимо подключить соответствующие зажимы прибора. Режим измерения и внутреннее сопротивление прибора (по умолчанию 1 Мом = 10⁶ Ом) можно задать щелчком на клавишах «~», «-» и SETTINGS;

Ω – измерение сопротивления (режим омметра) выполняется между двумя точками схемы, к которым подключены зажимы прибора. Правильный результат измерения получается, если задан режим «-», а измеряемая цепь не заземлена и не соединена с источником напряжения;

dB – измерение ослабления напряжения в децибелах между двумя точками схемы в соответствии с выражением

$$dB = 20 \lg \frac{V_n - V_m}{V_{ст}}$$

где V_n и V_m – напряжения в точках **n** и **m**;

$V_{ст}$ – стандартное напряжение (по умолчанию 1 В).

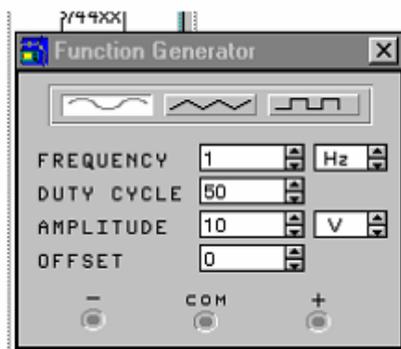


Рис. 36. Функциональный генератор

Функциональный генератор (рис.36)

является источником напряжения специальной формы. Форма сигнала задается клавишами верхнего ряда. Частота сигналов, коэффициент заполнения, амплитуда и смещение базовой линии (постоянной составляющей напряжения) устанавливаются клавишами. Изменение значений указанных параметров производится стрелочными кнопками сбоку от задаваемых значений:

FREQUENCY (Частота) выходных сигналов может устанавливаться в пределах от 1 Гц (Hz) до 999 МГц;

DUTY CYCLE (Коэффициент заполнения) задает для прямоугольного сигнала отношение в % длительности импульса с высоким рабочим уровнем к периоду, а для треугольного сигнала – отношение длительности положительного линейно нарастающего фронта к периоду. Коэффициент заполнения 50 % соответствует симметричной форме сигнала. Форма синусоидального сигнала не изменяется с изменением данного коэффициента;

AMPLITUDE (Амплитуда) задает максимальное значение выходного напряжения, отсчитываемое от базовой линии (смещения по постоянному току) в предположении, что выходной сигнал снимается с

зажимов **COM** (Общий) и «+» (или **COM** и «-»). Если выходной сигнал снимается с зажимов «+» и «-», то амплитуда удваивается;

OFFSET (Смещение) задает величину постоянного смещения выходного сигнала – его базовой линии в пределах от -999 до +999 кВ.

Отметим, что режим источника напряжения, реализуемый функциональным генератором, предполагает, что генерируемый сигнал свободен от искажений, шумов и пульсаций, свойственных реальным генераторам с реальной нагрузкой. Параметры генерируемого сигнала в режиме источника напряжения не зависят от величины подключенной нагрузки. Один из выходных зажимов генератора обычно подключается к земле.

Осциллограф имеет два канала **A** и **B** и применяется для визуального наблюдения двух сигналов одновременно. Осциллограф можно также использовать в режиме, когда один сигнал откладывается в качестве аргумента по оси абсцисс, а второй – в качестве функции по оси ординат. Кроме входных зажимов каналов **A** и **B** осциллограф снабжен клеммой заземления (**GROUND**) и входным зажимом запуска развертки **TRIGGER** (Запуск) (рис.37).

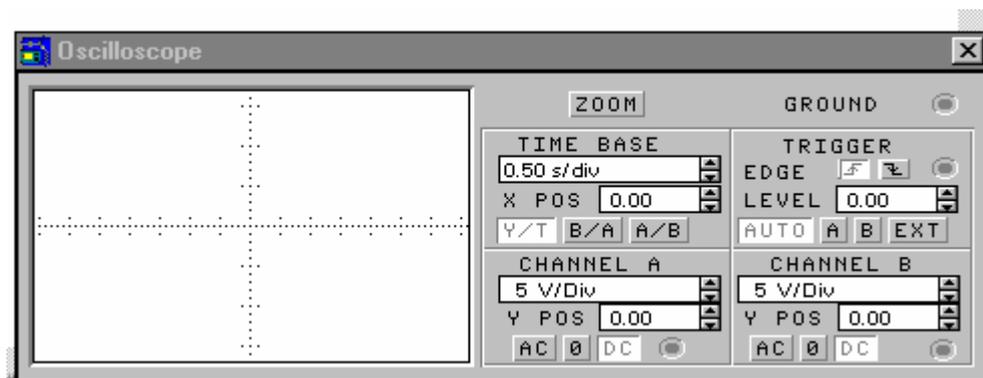


Рис.37. Осциллограф

Управление осциллографом заключается в установке режимов его работы с помощью клавишных переключателей, на которых для их выбора выполняется щелчок мышью.

Щелчок на одной из кнопок **Y/T**, **B/A**, **A/B** определяет оси **X** и **Y**. В режиме **Y/T** по оси **X** откладывается время, а по оси **Y** – напряжения сигналов **A** и **B**. Масштабирование оси **X** осуществляется в этом режиме кнопками «↑» и «↓» сбоку от табло текущего масштаба **TIME BASE** (s/div – с/дел). Скорость развертки можно изменять от 0,1 нс до 0,5 с на одно деление шкалы. Масштаб по оси **Y** устанавливается для каждого канала отдельно соответствующими кнопками сбоку от указателей текущего масштаба и может изменяться от 0,01 мВ/дел до 50 В/дел. Можно также задать смещение начальной точки развертки по

осям X и Y, причем смещение по оси X определяется кнопками **X POS** для обоих каналов одновременно. Если X POS установлено 0.00, то развертка начинается от начала экрана осциллографа, положительное ненулевое значение X POS сдвигает начало развертки вправо, а отрицательное – влево. Смещение сигналов по оси Y задается органами **Y POS** обоих каналов в пределах от -3 до 3 делений и используется для того, чтобы раздвинуть по вертикали изображения сигналов A и B. В режимах **B/A** и **A/B** шкала по оси X определяется установленным масштабом по каналам A и B соответственно.

Исследуемые входные сигналы могут подаваться в каналы A и B через открытые для постоянной составляющей входы и через закрытые входы. Выбор типа входа осуществляется щелчком на одной из клавиш **AC**, **0** или **DC**:

AC (Alternating Current – переменный ток) игнорирует постоянную составляющую сигнала. Данный режим эквивалентен подаче сигнала на вход канала через конденсатор большой емкости;

0 (нуль) заземляет вход канала. Этот режим используется для коррекции положения луча на экране осциллографа в отсутствие входного сигнала;

DC (Direct Current – постоянный ток) показывает на экране сигнал с учетом его постоянной составляющей. Следует иметь в виду, что схемное подключение разделительного конденсатора между контрольной точкой и входом осциллографа является некорректным, так как конденсатор воспринимается программой как компонент с неподключенным выводом.

Блок **TRIGGER** осциллографа определяет режим запуска горизонтальной развертки. Кнопки **EDGE** (фронт) «**┌**» и «**└**» устанавливают запуск развертки соответственно по фронту и спаду запускающего сигнала. **LEVEL** (уровень) определяет пороговое напряжение, превышение которого приводит к запуску развертки. Кнопки **AUTO**, **A**, **B** и **EXT** указывают, от какого сигнала производится запуск развертки:

AUTO – автоматический запуск;

A, **B** – запуск от входных сигналов A и B соответственно;

EXT – запуск от внешнего сигнала, подаваемого на зажим **TRIGGER**.

При подключении осциллографа его зажим **GROUND** следует подключать к общей точке (земле) схемы.

Частотный характериограф (Bode plotter – построитель диаграмм Боде) служит для вывода результатов моделирования в частотной области в виде амплитудно-частотной характеристики (АЧХ) и фазо-

частотной характеристики (ФЧХ). Прибор генерирует гармоническое напряжение с линейно изменяющейся частотой на своем зажиме **IN** и измеряет отношение напряжений на подключенных к схеме зажимах **OUT** (Выход) и **IN** (Вход) или же разность их фаз. Частоты всех источников переменного тока в схеме в процессе расчета игнорируются, однако схема должна включать хотя бы один такой источник.

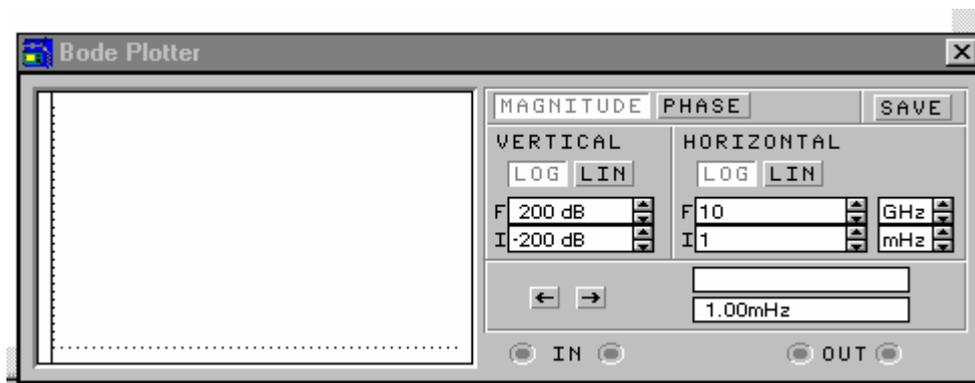


Рис. 38. Частотный характериограф

Характериограф выводит на свой экран в функции частоты либо отношение амплитуд, выраженное в относительных единицах или децибелах, либо разность их фаз в градусах. Режим работы прибора задается клавишами и стрелочными кнопками «↑», «↓» (рис.38):

- щелчок на клавише **MAGNITUDE** или **PHASE** устанавливает режим измерения АЧХ или ФЧХ соответственно;
- щелчок на клавишах **LOG** или **LIN** устанавливает логарифмический или линейный масштаб вертикальной **VERTICAL** и горизонтальной **HORIZONTAL** осей графика на экране прибора;
- шкалы осей задаются начальными (**I**) и конечными (**F**) значениями величин. По оси X всегда откладывается частота в Гц, а по оси Y – безразмерная величина (**LIN**) или децибелы (**LOG**) для отношения напряжений и градусы для разности фаз;
- для точного отсчета результата измерений можно использовать визир прибора, управляя его позицией с помощью кнопок с изображениями горизонтальных стрелок «←», «→» в нижней части лицевой панели характериографа. Визир можно также перемещать с помощью мыши, для чего визир (в виде крестика в левом нижнем углу экрана прибора) следует перетащить в нужную точку на графике АЧХ или ФЧХ. Отсчет измеряемой величины появляется на цифровом табло в нижней правой части прибора.

Генератор двоичных слов (Word generator) (рис.39) используется для испытаний цифровых схем и вырабатывает последовательно 16 циклически повторяющихся байтов – восьмибитовых двоичных

слов – в сопровождении синхроимпульсов, снимаемых с зажима **CLK**. В левой части раскрытого изображения генератора расположено информационное поле, представляющее собой матрицу из 16 строк и 8 колонок. При активизации моделирования включением питания информация выводится построчно на восемь выходных зажимов, расположенных в нижней части прибора. Текущее выходное слово отображается в шестнадцатеричном коде в правом нижнем углу прибора и в обычном двоичном коде непосредственно на изображениях зажимов.

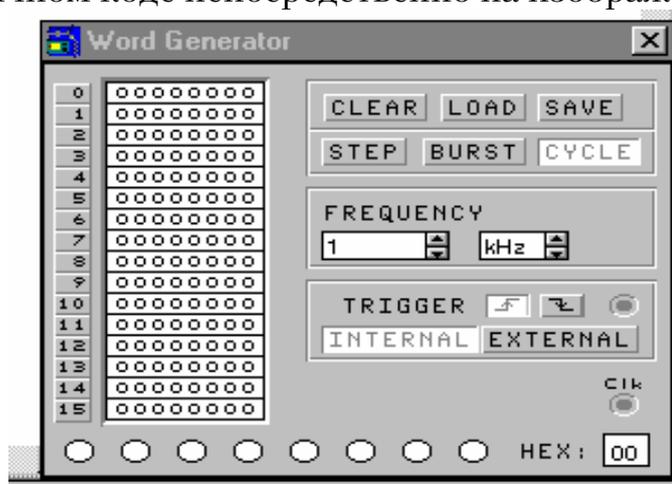


Рис. 39. Генератор двоичных слов

Ввод слов осуществляется после щелчка в нужной позиции информационного поля вводом символа 1 или 0 с клавиатуры. После первого такого щелчка в дальнейшем для выбора позиции символа можно использовать стрелочные клавиши клавиатуры ПЭВМ.

Режим работы генератора слов задается щелчками на нужных клавишах и кнопках:

- клавиши верхнего ряда **CLEAR**, **LOAD** и **SAVE** предназначены соответственно для очистки (обнуления) информационного поля, его загрузки извне и сохранения. После щелчков на **LOAD** и **SAVE** появляется диалоговое окно, запрашивающее имена файлов образцов информационного поля, которые образуются с расширением **.DP**;
- режим генерации слов устанавливается клавишами **STEP** (пошаговый режим с остановом после выдачи очередного слова), **BURST** (режим пачки с выдачей всех 16 слов, начиная с высвеченного слова, и последующим остановом) и **CYCLE** (циклический режим непрерывной генерации). В режиме **CYCLE** остановить генератор можно, либо выключив питание схемы, либо нажав комбинацию клавиш **Ctrl+T**;
- тактовая частота, с которой слова сменяют друг друга (или, что то же самое, частота синхронизации **CLK**), назначается стрелочными кнопками сбоку от табло **FREQUENCY** (частота);

- запуск генератора определяется группой клавиш **TRIGGER** и может осуществляться внутренним синхроимпульсом (**INTERNAL**) или внешним сигналом (**EXTERNAL**), подключаемым к зажиму в правом верхнем углу изображения генератора. В последнем случае можно назначить запуск как фронтом, так и спадом внешнего сигнала.

Логический анализатор (рис. 40) имеет 8 каналов и может развертывать во времени одновременно до восьми последовательностей прямоугольных логических сигналов. Он также формирует двоичное и шестнадцатеричное отображения текущего восьмиразрядного слова, удобные для проверки сложных устройств. При шестнадцатеричном представлении текущее – прорисовываемое в данный момент слово, например **10011101**, – разбивается на две четырехразрядные группы, причем значение числа в шестнадцатеричном коде равно $A = a_1 \times 16^1 + a_0 \times 16^0$, где a_i может принимать значения:

| Десятичное | Двоичное | Шестнадцатеричное |
|------------|----------|-------------------|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |

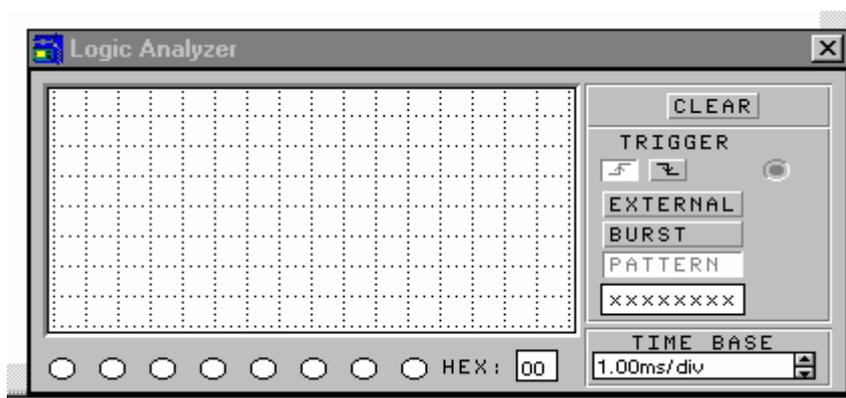


Рис. 40. Логический анализатор

Щелчок на клавише **CLEAR** приводит к очистке экрана логического анализатора. Условия запуска логического анализатора определяются клавишами блока **TRIGGER** (запуск):

- кнопки «┌» и «└» устанавливают, каким фронтом запускающего сигнала инициируется развертка анализируемых сигналов;
- клавиша **EXTERNAL** (внешний) задает запуск от внешнего сигнала, который должен быть подключен к зажиму правее клавиши;
- клавиша **BURST** (пачка) задает запуск от самих исследуемых сигналов. Это обычный режим работы анализатора;
- клавиша **PATTERN** (кодовая комбинация) определяет начало запуска развертки от момента поступления на входы анализатора кодовой комбинации, которую для данного режима необходимо напечатать под клавишей. Если в какой-либо позиции кодовой комбинации допустим как 0, так и 1, то в ней следует ввести X.

Логический преобразователь не имеет физического прототипа и является чисто компьютерным инструментом для выполнения преобразований цифровой схемы и ее логического описания. Изображение прибора (рис. 41) содержит шаблон таблицы истинности в левой части, клавиши управления в правой части и дисплей для отображения логического выражения в нижней части лицевой панели.

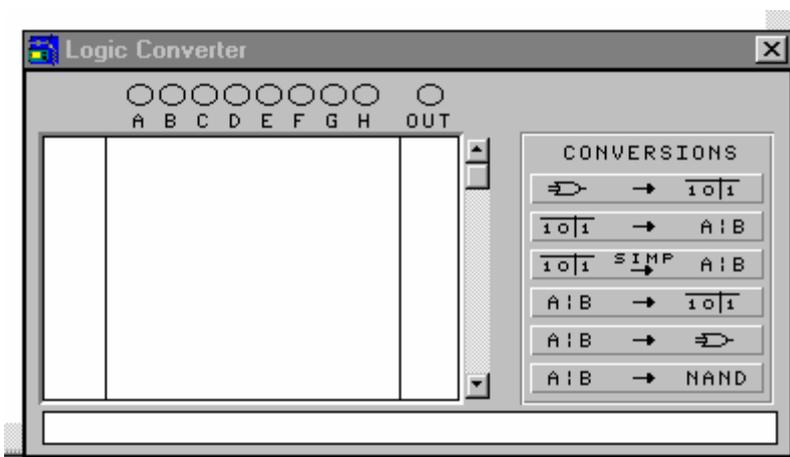
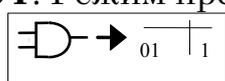


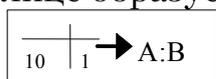
Рис. 41. Логический преобразователь

Анализируемая или синтезируемая цифровая схема может содержать до восьми входных переменных (аргументов) и один выход (функцию). Аргументы обозначаются буквами **A...H**, а функция – **OUT**. Режим преобразования назначается клавишами:



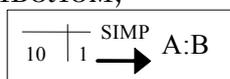
– преобразование схемы в таблицу истинности. В этом режиме входы схемы следует подключить к зажимам **A...H**, а выход – к зажиму **OUT**. После щелчка на клавише шаблон таблицы

истинности заполняется значениями переменных (0 и 1). Всего в таблице образуется 2^n строк, где n – число аргументов;



– преобразование таблицы истинности в логическое

выражение. Для создания таблицы истинности (если она еще не образована предыдущим преобразованием) необходимо щелкнуть на нескольких входных зажимах, соответствующих аргументам **A...H**. Появляется полный набор комбинаций аргументов, против которых в колонке OUT функции надо напечатать значения функции (0, 1 или X, причем X обозначает неопределенное значение функции). После щелчка на клавише в нижней части прибора появляется алгебраическая запись функции, в которой вместо обычного знака отрицания в виде черты над символом используется знак «'», проставляемый за символом;



– преобразование таблицы истинности в минимизированное логическое выражение.

Минимизация в ПЭВМ проводится методом Мак-Класки, допускающим по сравнению с методом карт Карно большее число аргументов. Следует заметить, что минимизация может потребовать значительного объема памяти. Результат минимизации выводится в нижней строке прибора;



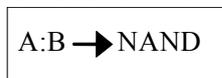
– преобразование логического выражения в таблицу истинности.

Перед щелчком на этой клавише следует ввести алгебраическое выражение в нижней строке прибора. Результат преобразования автоматически записывается в шаблон таблицы истинности;



– преобразование логического выражения в цифровую комбинационную схему.

Напечатанное предварительно выражение после щелчка на этой клавише преобразуется в схему на рабочем поле монтажного стола;



– создание цифровой схемы в функциональном

базисе И, НЕ, распространенном в инженерной практике.

Измерение постоянных токов и напряжений предусмотрено в программе возможностью подключения амперметров и вольтметров из магазина компонентов. Их количество не ограничено, поэтому при перемещении этих приборов на рабочее поле монтажного стола их изображения сохраняются в исходных позициях.

5.8. Выполнение анализа

Общие рекомендации. Вся информация программы **Electronics Workbench**, выводимая на экран **ПЭВМ**, располагается в окнах, которые можно перемещать, прокручивать (просматривать), изменять размеры и закрывать подобно другим окнам операционной среды **WINDOWS**. В окнах размещаются рабочее поле, магазин компонентов, панель инструментов, описание схемы и пр.

Если нужное в данный момент окно накрыто другими окнами, можно вывести его поверх остальных щелчком на титульной верхней полосе окна. Окно можно перетащить в другое место экрана, установив указатель на титульной полосе, нажав левую клавишу мыши, переместив окно и отпустив клавишу, когда окно окажется на новом месте. Двойной щелчок на кнопке в верхнем углу окна приводит к его закрытию. Для изменения размера окна следует перетащить его сторону или угол после появления на месте указателя двунаправленной стрелки. Для прокрутки окна можно перетащить кнопку, расположенную на правой стороне окна, или полосу в нижней части окна. Щелчок внутри области частично скрытого окна делает это окно активным и выводит его наверх. Следует иметь в виду, что общая площадь рабочего поля монтажного стола в четыре раза больше той, которая выводится на экран, и для того чтобы разместить дополнительные компоненты или просмотреть полностью рабочее поле, надо использовать кнопки прокрутки, расположенные с правого и нижнего краев окна.

Открытие ранее созданного и сохраненного схемного файла осуществляется командой **Open** (Открыть) из меню **File** (Файл). При этом появляется диалоговое окно, из которого выбирается имя схемного файла, после чего производится щелчок на кнопке **OK**. К такому же результату приводит нажатие на клавиатуре **Ctrl+O**.

В необходимых случаях на рабочем поле можно поместить текстовую информацию: позиционные обозначения, типы и значения параметров компонентов; краткое описание схемы; значения величин при настройке измерительных приборов. В большинстве случаев вставка текста назначается автоматически после вывода указателя на нужную позицию и щелчка мышью. Для перемещения указателя можно также использовать стрелочные клавиши и клавишу табуляции **Tab** на клавиатуре **ПЭВМ**.

Информация или комментарий к схеме печатается в окне описания (**Description Window**). Окно описания открывается командой **Description** из меню **Window** или сочетанием клавиш **Ctrl+D**. Если

нужно показать окно описания одновременно со схемой, то следует перетащить его к верхнему или нижнему краю рабочего поля и выбрать команду **Arrange** (Упорядочивание) из меню **Window**. Эта команда (или сочетание клавиш **Ctrl+W**) используется для упорядочивания составных частей монтажного стола. Указанная команда заставляет все открытые окна увеличиваться, насколько это возможно в конкретной ситуации. Для изменения порядка расположения частей монтажного стола следует перетащить эти части примерно туда, где их хотелось бы видеть, и выбрать вновь команду **Arrange** из меню **Window**. Способ упорядочивания окон зависит от их текущих позиций.

Перед проведением анализа собранную схему можно вывести на передний план командой **Circuit** из меню **Window**.

После создания схемы и подключения приборов можно назначить директивы анализа – опции (по умолчанию набор опций устанавливается автоматически и зависит от типов подключенных измерительных приборов). Директивы анализа задаются командой **Analysis Options (Ctrl+Y)** и предназначены для определения видов проводимых программой расчетов. Указанная команда выводит на экран диалоговое окно со следующими опциями:

| | |
|------------------------------------|---|
| Analysis Type | Вид анализа |
| Transient | Переходной процесс |
| Steady state | Установившийся режим |
| Active component | Активный компонент |
| Assume linear operation | Линеаризованный режим |
| Oscilloscope Display | Экран осциллографа |
| Pause after each screen | Пауза после каждого экрана |
| Store results for all nodes | Запоминание результатов для всех узлов |
| Points per cycle | Точек на период |
| Node plotter points per cycle | Точек частотной характеристики |
| Temporary file size for simulation | Размер временного файла для моделирования |

Перечисленные опции относятся к моделированию аналоговых электронных схем.

Исследование цифровых схем. Исследование цифровых схем, состоящих из логических комбинационных и последовательностных узлов, проводится на функциональном и принципиальном уровнях. При функциональном анализе все цифровые компоненты полагаются по умолчанию идеальными, не обладающими задержками переключения и не требующими источников питания. Часто такого анализа оказывается достаточно для оценки функционирования спроектиро-

ванного устройства. Однако, если необходим более детальный анализ поведения схемы или если схема включает наряду с цифровыми и аналоговые компоненты, то следует проводить моделирование на принципиальном уровне с указанием конкретных типов микросхем и цепей их питания.

Исследование обычно проводится во временной области с использованием осциллографа и/или логического анализатора. Проверка комбинационных схем без элементов памяти может осуществляться также с помощью логического преобразователя.

Синтез комбинационной схемы начинается, как правило, с таблицы истинности, в которой каждому набору аргументов ставится в соответствие то или иное значение (0 или 1) булевой функции. Таблица истинности заполняется на экране раскрытого изображения логического преобразователя. Количество строк в таблице определяется количеством аргументов, которое назначается щелчками на кнопках, расположенных над формируемой таблицей истинности, и равно 2^n , где n – число аргументов. Для каждого набора аргументов после позиционирования указателя с помощью клавиатуры вводится значение функции.

Таблица истинности может быть преобразована в алгебраическое выражение или в минимизированное алгебраическое выражение щелчком на соответствующей клавише раскрытого изображения преобразователя. Образовавшееся под таблицей истинности выражение, в котором знак отрицания изображается знаком «'» за соответствующим символом, можно далее реализовать на логических элементах **И**, **ИЛИ**, **НЕ** или же только на логических элементах **И**, **НЕ** щелчком на соответствующей клавише преобразователя. Автоматически синтезированная схема появляется на рабочем поле монтажного стола.

Возможно также обратное преобразование алгебраического выражения или схемы в таблицу истинности, например при проверке правильности проведенного формального синтеза схемы. Эти преобразования назначаются также щелчками на соответствующих клавишах после прорисовки комбинационной схемы на рабочем поле монтажного стола.

Цифровая схема создается на монтажном столе в последовательности, описанной ранее. Для извлечения компонента выбирается тот или иной раздел магазина компонентов и нужный объект перетаскивается на рабочее поле с помощью мыши. Если перетаскивается компонент в виде корпуса микросхемы, то перед выполнением межсоединений двойным щелчком на изображении корпуса посредством открывающегося диалогового окна назначается тип микросхемы по ее

функциональному признаку. Вторично произведенный двойной щелчок вызывает следующее диалоговое окно, предлагающее выбрать раздел библиотеки микросхем (**library**). Всего имеется четыре раздела: «**cmos**» (**КМОП**), «**default**» (по умолчанию), «**misc**» (смешанные), «**ttl**» (**ТТЛ**). По умолчанию компонент полагается идеальным. Выбор разделов «**cmos**» и «**ttl**» предлагает уточнить через диалоговое окно технологическую модификацию микросхемы (напряжение питания, наличие мощного выхода «**BUF**», открытый коллектор «**OC**» или сток «**OD**», скоростные свойства – «**HC**» (High-speed CMOS), маломощная **ТТЛШ** – «**LS**» (Low-power Schottky)). Вызов функции «**HELP**» (**ПОМОЩЬ**) клавишей **F1** или выбором из меню выдает дополнительную информацию по выделенному компоненту.

Electronics Workbench располагает обширной библиотекой моделей цифровых компонентов, **УГО** которых могут изображаться двояко: либо в соответствии со стандартом **ANSI** (American National Standard Institute), либо в соответствии с европейским стандартом **DIN** (второй весьма близок к **ГОСТам РФ**). Система обозначений назначается при инсталлировании программы. Цифровые компоненты содержатся в разделах **Gates** (Вентили), **Combinational** (Комбинационные узлы), **Sequential** (Последовательностные узлы), **IC** (Микросхемы), **Indicators** (Индикаторы). Кроме того, в разделе **Hybrid** (Смешанные узлы) содержатся модели мультивибратора и таймера, необходимые для построения времязадающих устройств.

Раздел **Gates** включает **УГО** вентиляей (логических элементов) (табл.1П2) и изображения корпусов микросхем с обозначением функции. Двойной щелчок на **УГО** вентиля приводит к появлению запроса о количестве его входов.

Раздел **Comb'1** предлагает комбинационные функциональные узлы, назначение которых обозначено на изображениях корпусов микросхем: **MUX** (Мультиплексоры), **DEC/DEMUX** (Дешифраторы/Демультимплексоры), **ENCODERS** (Шифраторы), **ARITHMETIC** (Арифметические узлы). Кроме того, имеются **УГО** полусумматора (**HS**) и полного одноразрядного сумматора (**SM**). В табл.2П2 приведены типовые представители данного раздела цифровых компонентов.

В разделе **Seq'1** магазина компонентов сосредоточены **УГО** различных триггеров, а также изображения корпусов микросхем с обозначением функции. Двойной щелчок на изображении корпуса предлагает перечни: **FLIP-FLOPS** – триггеров, **COUNTERS** – счетчиков либо **SHIFT REGS** – сдвигающих регистров, из которых выбираются конкретные типы и серии микросхем. **УГО** триггеров сведены в табл.3П2.

Раздел **IC** предлагает к выбору серии микросхем (**74xxxx – TTL**, **4xxx – КМОП**; знаком «**x**» обозначены переменные позиции обозначения микросхем).

Раздел **Indic** магазина компонентов содержит **УГО** приборов индикации, световой и звуковой сигнализации в соответствии с табл.4П2.

В разделе гибридных (смешанных) микросхем имеются цифроаналоговые и аналого-цифровые преобразователи, а также мультивибратор и таймер **555** (аналог отечественной микросхемы **КР1006ВИ1**), которые можно использовать для создания схем с встроенными тактовыми генераторами и схем для связи цифровых узлов с аналоговыми.

Анализ цифровых схем проводится преимущественно с помощью генератора двоичных слов (**Word generator**) и логического анализатора (**Logic analyzer**). Генератор слов позволяет задать по тактам повторяющиеся с периодом 16 тактов 8 различных входных логических сигналов для проверяемой схемы. Логический анализатор позволяет наблюдать до восьми логических сигналов одновременно в восьми узлах схемы, каковыми могут быть входы и выходы схемы, а также любые внутренние узлы в схеме.

Если назначить цвета проводников, которыми входы анализатора или осциллографа подключены к схеме, то именно этими цветами будут окрашены соответствующие сигналограммы на экранах этих приборов. Анализ схемы при низкой тактовой частоте может проводиться также с помощью световых пробников, подключаемых к интересующим пользователя узлам схемы. В отдельных случаях может оказаться полезным подключение головок громкоговорителей, подающих звуковые сигналы при превышении напряжением в контролируемом узле их порогов срабатывания.

5.9. Документирование лабораторной работы

Документирование лабораторной работы заключается в создании текстового файла отчета о работе, включающего сведения о цели работы, описание аналитической части работы (проектирование и расчет схемы), описание лабораторной установки (схема и набор измерительных приборов на монтажном столе), экспериментальные данные и выводы. В отдельных случаях требуется выполнить твердую копию отчета путем распечатки файла.

В программе **Electronics Workbench** можно задать распечатку любого или всех окон монтажного стола (окнами являются схема, каждый из раскрытых приборов, описание схемы и пр.). Задание на печать назначается с помощью диалогового окна, вызываемого командой Print

из меню **File**. Диалоговое окно предлагает к выбору всю доступную информацию о моделировании, составляющие которой отмечаются щелчком в соответствующей строке. Печать инициируется щелчком на клавише **OK**.

Можно также включить в файл отчета общий вид экрана **ПЭВМ** после выполнения анализа и отражения результатов на экранах измерительных приборов. Для этого нажатием клавиши **Print Screen** экран **ПЭВМ** копируется в буфер, из которого затем производится его вставка в текстовый файл отчета с помощью панели инструментов редактора **Microsoft Word**.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что понимается под вычислительной техникой?
2. Сколько существует поколений компьютеров?
3. Чем они отличаются друг от друга?
4. Назовите этапы развития вычислительной техники.
5. Что такое булева алгебра?
6. Кто основал алгебру логики?
7. Что такое дизъюнкция? Как обозначается графически?
8. Что такое конъюнкция? Как обозначается графически?
9. Что такое инверсия? Как обозначается графически?
10. Что такое импликация?
11. Что такое эквиваленция?
12. Что обозначает «1» в дизъюнкции?
13. Что такое минимизация переменных?
14. Чем отличается дизъюнктивная нормальная форма (ДНФ) от конъюнктивной нормальной формы (КНФ)?
15. Что такое триггер? Перечислите виды триггеров.
16. Что такое сумматор, шифратор, дешифратор?
17. Что такое карта Карно? Как графически исполняется карта Карно?
18. В чем суть принципов минимизации?
19. Что за программа Workbench?
20. Перечислите компоненты компьютерного монтажного стола.
21. Из чего состоит опорное меню?
22. Что такое совершенная дизъюнктивная (СДНФ) и совершенная конъюнктивная нормальная (СКНФ) формы?
23. Метод Квайна.
24. В чём состоит суть метода «склеивания»?
25. Понятие об архитектуре ЭВМ.
26. Анализ комбинационных схем.
27. Синтез комбинационных схем.

ЗАКЛЮЧЕНИЕ

Учебное пособие ориентировано в первую очередь на студентов и специалистов, перед которыми ставится задача теоретического ознакомления и практической реализации методов моделирования и конструирования электронных схем. В ряде случаев для решения прикладных задач анализа и конструирования таких схем необходимы навыки использования математико-компьютерного и технического моделирования.

Быстрое развитие транспортной отрасли и рост повышенных требований к инженерам и научным сотрудникам ставят перед ними задачи расширения возможных средств для управления различными звеньями отрасли.

Однако нельзя освоить арсенал всех предлагаемых средств, да это и не обязательно. Достаточно научиться использовать лишь те, которые просты в понимании и применении, но позволяют эффективным образом решать поставленные задачи.

Пособие поможет студентам, магистрантам и аспирантам сориентироваться в динамично меняющемся мире различных технологических ситуаций, что важно как в процессе обучения в вузе, так и при повышении своей квалификации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Архитектура многопроцессорных вычислительных систем [Текст] / О.С. Козлов [и др.]; под ред. В.И. Тимохина. – Л.: Изд-во ЛГУ, 1981. – 104 с.
2. Балашов, Е.П. Микропроцессоры и микропроцессорные системы [Текст] / Е.П. Балашов, Д.В. Пузанков; под ред. В.Б. Смопова. – М.: Радио и связь, 1981. – 328 с.
3. Завьялов, А.В. Цифровая обработка информации на основе быстродействующих БИС [Текст] / А.В. Завьялов, П.П. Мальцев, В.Г. Соколов; под ред. В.Г. Домрачева. – М.: Энергоатомиздат, 1998. – 136 с. (Микропроцессорные БИС и их применение).
4. Каган, Б.М. Электронные вычислительные машины и системы [Текст] / Б.М. Каган. – М.: Энергоатомиздат, 1985. – 552с .
5. Микропроцессорные комплекты интегральных схем. Состав и структура [Текст]: справочник / под ред. А.А. Васенкова и В.А. Шахнова. – М.: Радио и Связь, 1982. – 192 с.
6. Елисеев, В.А. Проектирование элементов и устройств информационно-вычислительных систем в среде Workbench [Текст] / В.А. Елисеев, А.П. Ремонтов. – Пенза: ПГАСА, 2002.
7. Елисеев, В.А. Проектирование цифровых элементов и устройств информационно-вычислительных систем в Workbench [Текст]: учеб. пособие / В.А. Елисеев, А.В. Баженов, А.П. Ремонтов. – Пенза: ПГУАС, 2009. – Ч. 2.
8. www.wikipedia.org/wiki/

ПРИЛОЖЕНИЯ

Приложение 1

ЗАДАНИЯ И ПРИМЕРЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ ПО ПРОЕКТИРОВАНИЮ ЭЛЕМЕНТОВ И УСТРОЙСТВ ЦИФРОВЫХ СХЕМ

1. Проектирование и синтез комбинационных схем

Задание. Найти минимальную двухступенчатую реализацию функции f четырех переменных X_1, X_2, X_3, X_4 заданной таблицей истинности.

Например, таблица истинности для функции f задана следующим образом:

| № вар. | m_0 | m_1 | m_2 | m_3 | m_4 | m_5 | m_6 | m_7 | m_8 | m_9 | m_{10} | m_{11} | m_{12} | m_{13} | m_{14} | m_{15} | Пр |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|----------|----|
| 19 | 1 | 1 | | | 1 | 1 | | | 1 | | | | 1 | | 1 | 1 | |

Здесь $m_0...m_{15}$ – минтермы функции.

Функция f может быть представлена в виде суммы тех минтермов для точек, в которых она обращается в «1»: $f = m_0 + m_1 + m_4 + m_5 + m_8 + m_{12} + m_{14} + m_{15}$. Для минимизации этой функции можно использовать либо аксиомы и правила алгебры логики, либо карты Карно, либо табличный метод (метод Квайна), который подробно изложен в разд. 2. На первом этапе этого метода необходимо получить перечень простых импликант требуемой функции.

| Группы | переменные | | | | исп. | Исходные минтермы | Импликанты |
|-------------------|------------|-------|-------|-------|------|---------------------|-----------------------------------|
| | X_1 | X_2 | X_3 | X_4 | | | |
| $S_0(m_0)$ | 0 | 0 | 0 | 0 | √ | | |
| $S_{1-1}(m_1)$ | 0 | 0 | 0 | 1 | √ | | |
| $S_{1-2}(m_4)$ | 0 | 1 | 0 | 0 | √ | | |
| $S_{1-3}(m_8)$ | 1 | 0 | 0 | 0 | √ | | |
| $S_{2-1}(m_5)$ | 0 | 1 | 0 | 1 | | | $P1(\bar{X}_1 X_2 \bar{X}_3 X_4)$ |
| $S_{2-2}(m_{12})$ | 1 | 1 | 0 | 0 | √ | | |
| $S_3(m_{14})$ | 1 | 1 | 1 | 0 | √ | | |
| $S_4(m_{15})$ | 1 | 1 | 1 | 1 | √ | | |
| S'_{1-1} | 0 | 0 | 0 | - | √ | $S_0 + S_{1-1}$ | |
| S'_{1-2} | 0 | - | 0 | 0 | √ | $S_0 + S_{1-2}$ | |
| S'_{1-3} | - | 0 | 0 | 0 | √ | $S_0 + S_{1-3}$ | |
| S'_{2-1} | 0 | - | 0 | 1 | | $S_{1-1} + S_{2-1}$ | $P2(\bar{X}_1 \bar{X}_3 X_4)$ |
| S'_{2-2} | 0 | 1 | 0 | - | | $S_{1-2} + S_{2-1}$ | $P3(\bar{X}_1 X_2 \bar{X}_3)$ |

Продолжение прил. 1

| | | | | | | | |
|------------|---|---|---|---|--|-------------------|--------------------------|
| S'_{2-3} | - | 1 | 0 | 0 | | $S_{1-2}+S_{2-2}$ | $P4(X2\bar{X}3\bar{X}4)$ |
| S'_{2-4} | 1 | - | 0 | 0 | | $S_{1-3}+S_{2-2}$ | $P5(X1\bar{X}3\bar{X}4)$ |
| S'_3 | 1 | 1 | - | 0 | | $S_{2-2}+S_3$ | $P6(X1X2\bar{X}4)$ |
| S'_4 | 1 | 1 | 1 | - | | S_3+S_4 | $P7(X1X2X3)$ |

| | | | | | | | |
|-------------|---|---|---|---|--|---------------------|------------------------|
| S''_{1-1} | 0 | - | 0 | - | | $S'_{1-1}+S'_{2-2}$ | $P8(\bar{X}1\bar{X}3)$ |
| S''_{1-2} | 0 | - | 0 | - | | $S'_{1-2}+S'_{2-1}$ | $P8(\bar{X}1\bar{X}3)$ |
| S''_{1-3} | - | - | 0 | 0 | | $S'_{1-2}+S'_{2-4}$ | $P9(\bar{X}3\bar{X}4)$ |
| S''_{1-4} | - | - | 0 | 0 | | $S'_{1-3}+S'_{2-3}$ | $P9(\bar{X}3\bar{X}4)$ |

Простыми импликантами в этой таблице будут импликанты (строки), не помеченные символом «v». Это будут импликанты $P1(\bar{X}1X2\bar{X}3X4)$, $P2(\bar{X}1\bar{X}3X4)$, $P3(\bar{X}1X2\bar{X}3)$, $P4(X2\bar{X}3\bar{X}4)$, $P5(X1\bar{X}3\bar{X}4)$, $P6(X1X2\bar{X}4)$, $P7(X1X2X3)$, $P8(\bar{X}1\bar{X}3)$ и $P9(\bar{X}3\bar{X}4)$. Для получения «минимального покрытия» (набор тех простых импликант, которые покрывают все «1-точки» функции f) составим таблицу простых импликант.

| Импликанты | m_0 | m_1 | m_4 | m_5 | m_8 | m_{12} | m_{14} | m_{15} |
|----------------------------|-------|-------|-------|-------|-------|----------|----------|----------|
| $P1(\bar{X}1X2\bar{X}3X4)$ | | | | 1 | | | | |
| $P2(\bar{X}1\bar{X}3X4)$ | | 1 | | 1 | | | | |
| $P3(\bar{X}1X2\bar{X}3)$ | | | 1 | 1 | | | | |
| $P4(X2\bar{X}3\bar{X}4)$ | | | 1 | | | 1 | | |
| $P5(X1\bar{X}3\bar{X}4)$ | | | | | 1 | 1 | | |
| $P6(X1X2\bar{X}4)$ | | | | | | 1 | 1 | |
| $P7(X1X2X3)$ | | | | | | | 1 | 1 |
| $P8(\bar{X}1\bar{X}3)$ | 1 | 1 | 1 | 1 | | | | |
| $P9(\bar{X}3\bar{X}4)$ | 1 | | 1 | | 1 | 1 | | |

Поскольку импликанта $P7$ является существенной для столбца m_{15} , $P8$ доминирует над $P1$, $P2$ и $P3$, а $P9$ доминирует над $P4$, $P5$ и $P6$, то в минимальное покрытие функции f достаточно включить импликанты $P7(X1X2X3)$, $P8(\bar{X}1\bar{X}3)$ и $P9(\bar{X}3\bar{X}4)$.

Продолжение прил. 1

| | | | | |
|-----------------|-----------------|-----------------|-----------------|----------|
| | $X2$ | $\overline{X2}$ | | |
| $X1$ | $1(m_{12})$ | | | $1(m_8)$ |
| | $1(m_{14})$ | $1(m_{15})$ | | |
| $\overline{X1}$ | | | | |
| | $1(m_4)$ | $1(m_5)$ | $1(m_1)$ | $1(m_0)$ |
| | $\overline{X4}$ | $X4$ | $\overline{X4}$ | |

$\overline{X3}$
 $X3$
 $\overline{X3}$

Аналогичное решение может быть получено с помощью карт Карно. Для 4 входных переменных карта Карно должна содержать 16 ячеек (2^4), полученных разбиением квадрата области на 16 частей. В ячейках, соответствующих «1-точкам», записаны «1». Для минимизации объединяем соседние ячейки с

«1»: импликанта $P1(\overline{X1}\overline{X3})$, объединившая 4 точки m_4, m_5, m_1 и m_0 , импликанта $P2(\overline{X3}\overline{X4})$, объединившая 4 точки m_4, m_0, m_8 и m_{12} , и импликанта $P3(X1X2X3)$, объединившая 2 точки m_{14} и m_{15} . Это решение полностью совпадает с решением, полученным табличным методом.

Задания для самостоятельного решения

Найти минимальную двухступенчатую реализацию функции f четырех переменных $X1, X2, X3, X4$ заданной таблицей истинности:

| $N_{\text{в.ад.}}$ | m_0 | m_1 | m_2 | m_3 | m_4 | m_5 | m_6 | m_7 | m_8 | m_9 | m_{10} | m_{11} | m_{12} | m_{13} | m_{14} | m_{15} |
|--------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|----------|
| 1 | 1 | 1 | | | 1 | 1 | | | 1 | | | 1 | 1 | | | 1 |
| 2 | 1 | 1 | | 1 | 1 | | | 1 | | 1 | | | 1 | 1 | | |
| 3 | | | 1 | 1 | | | 1 | 1 | | | 1 | | | 1 | 1 | 1 |
| 4 | 1 | | 1 | | | | 1 | 1 | | | 1 | 1 | 1 | | | 1 |
| 5 | 1 | 1 | 1 | 1 | | | | | 1 | 1 | | | | 1 | 1 | |
| 6 | | 1 | 1 | 1 | 1 | | | 1 | 1 | | | | 1 | 1 | | |
| 7 | 1 | | | | 1 | 1 | | | 1 | 1 | | | 1 | | 1 | 1 |
| 8 | | 1 | 1 | | | 1 | | | 1 | 1 | | | 1 | 1 | | 1 |
| 9 | 1 | 1 | 1 | | | | | 1 | 1 | | | | | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | | | | 1 | | | | 1 | 1 | | | 1 | 1 |
| 11 | | 1 | 1 | | 1 | | | 1 | | | 1 | 1 | | 1 | 1 | |
| 12 | 1 | | 1 | 1 | | | 1 | 1 | | | | | 1 | 1 | 1 | |
| 13 | | 1 | 1 | | | | | 1 | 1 | | 1 | | 1 | 1 | | 1 |
| 14 | | 1 | | 1 | | | 1 | 1 | | 1 | 1 | | 1 | 1 | | |
| 15 | | 1 | 1 | 1 | | | | 1 | 1 | 1 | | | | | 1 | 1 |

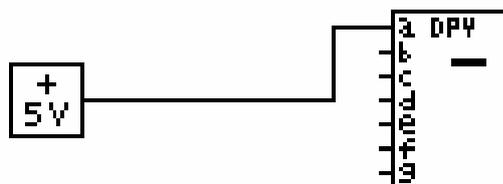
| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16 | 1 | 1 | | | 1 | 1 | | | | 1 | 1 | | 1 | 1 | | |
| 17 | | 1 | | | 1 | 1 | | | | 1 | 1 | | | 1 | 1 | 1 |
| 18 | 1 | | | 1 | 1 | | | 1 | | 1 | | | 1 | | 1 | 1 |
| 19 | 1 | 1 | | | 1 | 1 | | | 1 | | | | 1 | | 1 | 1 |
| 20 | 1 | | 1 | 1 | | | | 1 | 1 | | | 1 | | 1 | | |
| 21 | | 1 | 1 | 1 | | | | | 1 | 1 | | | 1 | 1 | | 1 |
| 22 | | | 1 | | | 1 | 1 | | | 1 | 1 | 1 | 1 | 1 | | |
| 23 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | 1 | | | | 1 |
| 24 | | | | | 1 | 1 | 1 | 1 | | | | 1 | | 1 | | 1 |
| 25 | | 1 | 1 | | | 1 | 1 | | | 1 | 1 | | | 1 | 1 | |
| 26 | 1 | 1 | | | 1 | | | | | 1 | 1 | 1 | | | 1 | 1 |
| 27 | | 1 | 1 | 1 | 1 | | | | | | 1 | 1 | 1 | | | 1 |
| 28 | 1 | 1 | | | | 1 | | 1 | 1 | | | 1 | 1 | | | 1 |
| 29 | | 1 | 1 | | 1 | | | 1 | 1 | 1 | | | 1 | 1 | | |
| 30 | 1 | 1 | | | | 1 | 1 | | 1 | 1 | | | 1 | | | 1 |
| 31 | | 1 | 1 | | 1 | | | 1 | | | | 1 | 1 | | | 1 |
| 32 | 1 | 1 | 1 | 1 | | | | | | | 1 | 1 | 1 | | | 1 |

Здесь $m_0...m_{15}$ – минтермы функции f .

2. Проектирование и синтез комбинационных многовыходовых схем

З а д а н и е . Найти минимальную двухступенчатую реализацию множества функций, представляющих собой отображение на семисегментном индикаторе последовательно цифр номера зачетной книжки.

Семисегментный индикатор имеет семь входов A, B, C, D, E, F, G , каждый из которых при наличии единичного сигнала на нем зажигает соответствующий сектор индикатора (схема индикатора приведена ниже). Сегменты индикатора именуется буквами “ a, b, c, d, e, f, g ” по кругу по часовой стрелке, начиная с верхнего сегмента. Сегмент g располагается в центре.



Например, необходимо последовательно отобразить номер зачетной книжки «00193». Для этого каждой цифре дадим в соответствие определенную комбинацию входного кода трех переменных X_1, X_2 и X_3 .

Продолжение прил. 1

| X1 | X2 | X3 | Число | A | B | C | D | E | F | G |
|----|----|----|-------|---|---|---|---|---|---|---|
| 0 | 0 | 0 | Е | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 9 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 3 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | Е | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | Е | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

В этой таблице столбец D идентичен столбцу A , а столбец C идентичен столбцу B ; поэтому исключим их.

Найдем импликанты множества функций A, B, E, F и G табличным способом (см. разд.2). Для этого обратимся к таблице, в которой входные переменные сгруппированы в множества S_0, S_1, S_2, S_3 по количеству единиц в коде, соответствующем данной строке. После преобразования таблицы и слияния строк получим восемнадцать импликант $P_1 \dots P_{18}$ покрывающего множества.

| | | X1 | X2 | X3 | N | A | B | E | F | G | Метка | Составляющие |
|--------|---|----|----|----|---|---|---|---|---|---|----------|-------------------|
| S_0 | 1 | 0 | 0 | 0 | Е | 1 | 0 | 1 | 1 | 1 | P_1 | |
| S_1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | P_2 | |
| | 2 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | P_3 | |
| | 3 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | ∨ | |
| S_2 | 1 | 0 | 1 | 1 | 9 | 1 | 1 | 0 | 1 | 1 | P_4 | |
| | 2 | 1 | 0 | 1 | 3 | 1 | 1 | 0 | 0 | 1 | P_5 | |
| | 3 | 1 | 1 | 0 | Е | 1 | 0 | 1 | 1 | 1 | ∨ | |
| S_3 | 1 | 1 | 1 | 1 | Е | 1 | 0 | 1 | 1 | 1 | ∨ | |
| S'_1 | 1 | 0 | 0 | - | | 1 | 0 | 1 | 1 | 0 | P_6 | S_0+S_{1-1} |
| | 2 | 0 | - | 0 | | 1 | 0 | 1 | 1 | 0 | P_7 | S_0+S_{1-2} |
| | 3 | - | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | ∨ | S_0+S_{1-3} |
| S'_2 | 1 | 0 | - | 1 | | 1 | 1 | 0 | 1 | 0 | P_8 | $S_{1-1}+S_{2-1}$ |
| | 2 | - | 0 | 1 | | 1 | 1 | 0 | 0 | 0 | P_9 | $S_{1-1}+S_{2-2}$ |
| | 3 | 0 | 1 | - | | 1 | 1 | 0 | 1 | 0 | P_{10} | $S_{1-2}+S_{2-1}$ |
| | 4 | - | 1 | 0 | | 1 | 0 | 1 | 1 | 0 | P_{11} | $S_{1-2}+S_{2-3}$ |
| | 5 | 1 | 0 | - | | 0 | 1 | 0 | 0 | 0 | P_{12} | $S_{1-3}+S_{2-2}$ |
| | 6 | 1 | - | 0 | | 0 | 0 | 0 | 0 | 0 | ∨ | $S_{1-3}+S_{2-3}$ |

Продолжение прил. 1

| | | | | | | | | | | | | |
|---------|---|---|---|---|--|---|---|---|---|---|-------|---------------------|
| S'_3 | 1 | - | 1 | 1 | | 1 | 0 | 0 | 1 | 1 | $P13$ | $S_{2-1}+S_3$ |
| | 2 | 1 | - | 1 | | 1 | 0 | 0 | 0 | 1 | $P14$ | $S_{2-2}+S_3$ |
| | 3 | 1 | 1 | - | | 1 | 0 | 1 | 1 | 1 | $P15$ | $S_{2-3}+S_3$ |
| S''_1 | 1 | 0 | - | - | | 1 | 0 | 0 | 1 | 0 | $P16$ | $S'_{1-1}+S'_{2-3}$ |
| | 2 | - | 0 | - | | 0 | 0 | 0 | 0 | 0 | - | $S'_{1-1}+S'_{2-5}$ |
| | 3 | 0 | - | - | | 1 | 0 | 0 | 1 | 0 | $P16$ | $S'_{1-2}+S'_{2-1}$ |
| | 4 | - | - | 0 | | 0 | 0 | 0 | 0 | 0 | - | $S'_{1-2}+S'_{2-6}$ |
| | 5 | - | 0 | - | | 0 | 0 | 0 | 0 | 0 | - | $S'_{1-3}+S'_{2-2}$ |
| | 6 | - | - | 0 | | 0 | 0 | 0 | 0 | 0 | - | $S'_{1-3}+S'_{2-4}$ |
| S''_2 | 1 | - | - | 1 | | 1 | 0 | 0 | 0 | 0 | $P17$ | $S'_{2-1}+S'_{3-2}$ |
| | 2 | - | - | 1 | | 1 | 0 | 0 | 0 | 0 | $P17$ | $S'_{2-2}+S'_{3-1}$ |
| | 3 | - | 1 | - | | 1 | 0 | 0 | 1 | 0 | $P18$ | $S'_{2-3}+S'_{3-3}$ |
| | 4 | - | 1 | - | | 1 | 0 | 0 | 1 | 0 | $P18$ | $S'_{2-4}+S'_{3-1}$ |
| | 5 | 1 | - | - | | 0 | 0 | 0 | 0 | 0 | - | $S'_{2-5}+S'_{3-3}$ |
| | 6 | 1 | - | - | | 0 | 0 | 0 | 0 | 0 | - | $S'_{2-6}+S'_{3-2}$ |

Составим карту Карно для каждого столбца:

A

| | $\bar{X}3$ | $X3$ | $\bar{X}3$ | |
|------------|------------|----------|------------|------------|
| $X1$ | $1(m_6)$ | $1(m_7)$ | $1(m_5)$ | |
| $\bar{X}1$ | $1(m_2)$ | $1(m_3)$ | $1(m_1)$ | $1(m_0)$ |
| | | $X2$ | | $\bar{X}2$ |

F

| | $\bar{X}3$ | $X3$ | $\bar{X}3$ | |
|------------|------------|----------|------------|------------|
| $X1$ | $1(m_6)$ | $1(m_7)$ | | |
| $\bar{X}1$ | $1(m_2)$ | $1(m_3)$ | $1(m_1)$ | $1(m_0)$ |
| | | $X2$ | | $\bar{X}2$ |

B

| | $\bar{X}3$ | $X3$ | $\bar{X}3$ | |
|------------|------------|----------|------------|------------|
| $X1$ | | | $1(m_5)$ | $1(m_4)$ |
| $\bar{X}1$ | $1(m_2)$ | $1(m_3)$ | $1(m_1)$ | |
| | | $X2$ | | $\bar{X}2$ |

G

| | $\bar{X}3$ | $X3$ | $\bar{X}3$ | |
|------------|------------|----------|------------|------------|
| $X1$ | $1(m_6)$ | $1(m_7)$ | $1(m_5)$ | |
| $\bar{X}1$ | | $1(m_3)$ | | $1(m_0)$ |
| | | $X2$ | | $\bar{X}2$ |

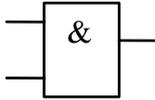
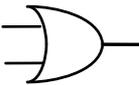
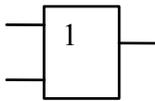
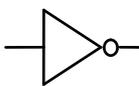
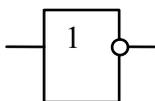
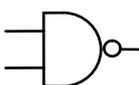
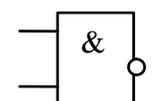
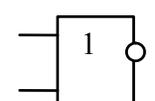
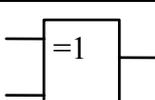
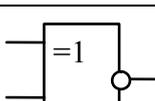
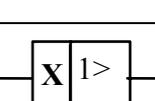
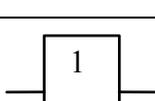
| | | | |
|-------------|-------------|-------------|-------------|
| E | \bar{X}_3 | X_3 | \bar{X}_3 |
| X_1 | $1(m_6)$ | $1(m_7)$ | |
| \bar{X}_1 | $1(m_2)$ | | $1(m_0)$ |
| | X_2 | \bar{X}_2 | |

Получим минимальное покрытие для каждого столбца.

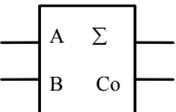
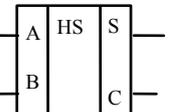
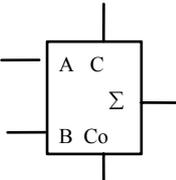
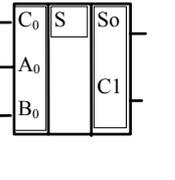
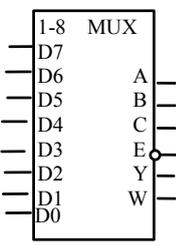
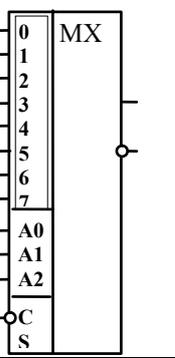
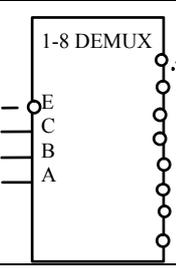
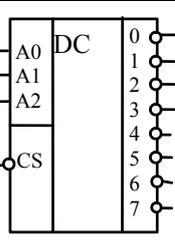
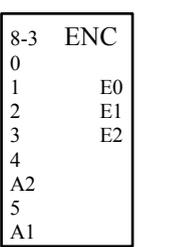
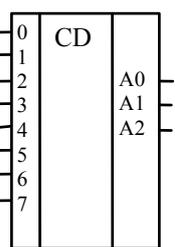
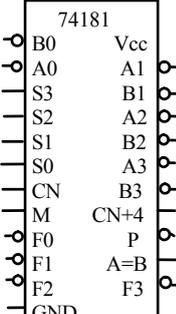
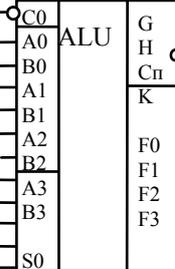
| | |
|-----------|---|
| A: | $P_5(X_1\bar{X}_2X_3) + P_{16}(\bar{X}_1) + P_{15}(X_1X_2)$ |
| B: | $P_{12}(X_1\bar{X}_2) + P_{10}(\bar{X}_1X_2) + P_9(\bar{X}_2X_3)$ |
| E: | $P_6(\bar{X}_1\bar{X}_2) + P_{11}(X_2\bar{X}_3) + P_{15}(X_1X_2)$ |
| F: | $P_{15}(X_1X_2) + P_{16}(\bar{X}_1)$ |
| G: | $P_1(\bar{X}_1\bar{X}_2\bar{X}_3) + P_{15}(X_1X_2) + P_4(\bar{X}_1X_2X_3) + P_5(X_1\bar{X}_2X_3)$ |

При получении минимального покрытия для каждой функции A, B, E, F, G учитывалась стоимость используемых импликант. Стоимость импликанты определяется количеством входов и выходов этой импликанты. Причем, если импликанта уже использовалась раньше для получения одной из функций, то для других функций она изменит свою стоимость, которая становится равной количеству только выходов. Так, если первоначально стоимость импликанты P_{15} была равна 5 (два входа и три выхода), то для функций “ E ” и “ F ” она уменьшается до 1.

Приложение 2
Таблица 1 П 2

| ANSI | | ГОСТ | | Примечание |
|---|--------------------|---|------------------------------------|-----------------------------|
| УГО | Функция | УГО | Функция | |
|  | AND |  | И | $Z = X \wedge Y$ |
|  | OR |  | ИЛИ | $Z = X \vee Y$ |
|  | NOT |  | НЕ | $Z = \bar{X}$ |
|  | NAND |  | И-НЕ | $Z = \overline{X \wedge Y}$ |
|  | NOR |  | ИЛИ-НЕ | $Z = \overline{X \vee Y}$ |
|  | XOR |  | ИСКЛ ИЛИ | $Z = X \oplus Y$ |
|  | XNOR |  | ИСКЛ ИЛИ-НЕ | $Z = \overline{X \oplus Y}$ |
|  | TRISTATE BUFFER |  | Буфер с 3 состояниями выхода | $Z = X \cdot E$ |
|  | BUFFER |  | Буферный повторитель | $Z = X$ |

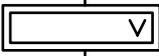
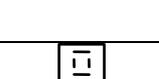
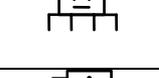
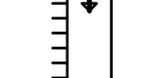
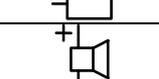
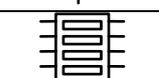
Продолжение прил. 2
Таблица 2 П 2

| ANSI | | ГОСТ | | Примечание |
|---|------------------------------------|---|-------------------------------------|---|
| УГО | Функция | УГО | Функция | |
|  | Half Summer (HS) |  | Полусумматор | A, B – операнды Σ – сумма Co – перенос |
|  | Summer (SM) |  | Сумматор | A, B – операнды C – вход переноса Σ – сумма Co – выход переноса |
|  | Multiplexer (MUX) |  | Мультиплексор | D0–D7 – информационные входы A, B, C – адресные входы E – вход разрешения Y – выход W – инверсный выход |
|  | Decoder/ Demultiplexer (DEC/DEMUX) |  | Дешифратор | A, B, C – адресные входы E – вход разрешения |
|  | Encoder (ENC) |  | Шифратор приоритета | 0-7 информационные входы E0 – E2 – входы разрешения A0 – A2 – выходы |
|  | Arithmetic logic unit (ALU) |  | Арифметическо-логическое устройство | Аналог К155ИПЗ |

Продолжение прил. 2
Таблица 3 П 2

| ANSI | | ГОСТ | | Примечание |
|------|--------------|------|------------|---|
| УГО | Функция | УГО | Функция | |
| | RS flip-flop | | RS-триггер | Рабочие уровни асинхронных входов S,R – высокие |
| | JK flip-flop | | JK-триггер | |
| | JK flip-flop | | JK-триггер | |
| | D flip-flop | | D-триггер | Имеются асинхронные входы S и R с низкими рабочими уровнями |
| | D flip-flop | | D-триггер | |

Окончание прил. 2
Таблица 4 П 2

| УГО | Функция | Примечание |
|---|---------------------------------------|---|
|  | Вольтметр | После двойного щелчка устанавливается сопротивление и шкала вольтметра и амперметра |
|  | Амперметр | |
|  | Лампа накаливания | Устанавливается цвет свечения: красный, зеленый, синий |
|  | Световой пробник | |
|  | Семисегментный индикатор | Преобразует 4-разрядный двоичный код в символ шестнадцатеричного кода (0...9, A...F) |
|  | Декодирующий семисегментный индикатор | |
|  | Запоминающее устройство | |
|  | Головка громкоговорителя | После двойного щелчка назначается частота звука, порог срабатывания и выходная мощность. Срабатывает при превышении порога срабатывания |
|  | Полосковый индикатор | |
|  | Декодирующий полосковый индикатор | Содержит матрицу из 10 светодиодов(аноды – слева, катоды – справа). Каждый светодиод включается при протекании тока Имеет встроенный АЦП. В зависимости от напряжения между выводами, светятся младшие n светодиодов. |

ОГЛАВЛЕНИЕ

| | |
|--|----|
| ВВЕДЕНИЕ..... | 3 |
| 1. ИСТОРИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ..... | 4 |
| 1.1. Этапы развития вычислительной техники | 4 |
| 1.2. Поколение ЭВМ | 10 |
| 1.3. Принципы организации вычислительных систем | 13 |
| 1.3.1. Классификация вычислительных систем | 13 |
| 1.4. Понятие об архитектуре ЭВМ..... | 17 |
| 2. ОСНОВЫ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ..... | 20 |
| 2.1. Понятие информационной системы | 20 |
| 2.2. Основные элементы цифровых устройств..... | 21 |
| 2.3. Элементы булевой алгебры (алгебра логики) | 24 |
| 2.4. Дизъюнктивная нормальная форма (ДНФ), совершенная дизъюнктивная нормальная форма (СДНФ), конъюнктивная нормальная форма (КНФ), совершенная конъюнктивная нормальная форма (СКНФ) | 26 |
| 3. АНАЛИЗ И СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ | 29 |
| 3.1. Анализ комбинационных схем | 29 |
| 3.2. Синтез комбинационных схем | 29 |
| 3.3. Табличный метод получения простых импликант | 37 |
| 3.4. Получение простых импликант с помощью карт Карно..... | 38 |
| 3.5. Анализ и синтез комбинационных схем с разветвленным выходом | 40 |
| 3.6. Получение простых многовыходовых импликант табличным методом | 42 |
| 3.7. Принципы минимизации | 44 |
| 4. СИНТЕЗ ТИПОВЫХ УЗЛОВ ЦИФРОВЫХ СХЕМ | 51 |
| 4.1. Синтез триггеров..... | 51 |
| 4.2. Синтез пересчётных схем (счётчиков) | 53 |
| 4.3. Синтез регистров..... | 54 |
| 4.4. Синтез сумматоров..... | 55 |
| 4.5. Синтез дешифраторов, шифраторов и преобразователей кодов..... | 56 |
| 4.6. Синтез распределителей и мультиплексоров | 57 |
| 5. ПРИНЦИПЫ МОДЕЛИРОВАНИЯ ЭЛЕКТРОННЫХ СХЕМ В СРЕДЕ WORKBENCH | 59 |
| 5.1. Компьютерный монтажный стол. Принципы компьютерного макетирования | 59 |

| | |
|--|----|
| 5.2. Магазин компонентов..... | 61 |
| 5.3. Панель приборов..... | 65 |
| 5.4. Опорное меню..... | 66 |
| 5.5. Размещение и обозначение компонентов | 71 |
| 5.6. Выполнение межсоединений | 72 |
| 5.7. Подключение и настройка измерительных приборов..... | 73 |
| 5.8. Выполнение анализа | 83 |
| 5.9. Документирование лабораторной работы | 87 |
| КОНТРОЛЬНЫЕ ВОПРОСЫ..... | 89 |
| ЗАКЛЮЧЕНИЕ..... | 90 |
| БИБЛИОГРАФИЧЕСКИЙ СПИСОК..... | 91 |
| ПРИЛОЖЕНИЯ | 92 |

Учебное издание

Камбург Владимир Григорьевич
Кувшинова Ольга Александровна
Елисеев Владимир Александрович

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И СЕТИ В ОТРАСЛИ
Учебное пособие

Редактор М.А. Сухова
Верстка Н.А. Сазонова

Подписано в печать 8.04.14. Формат 60×84/16.
Бумага офисная «Снегурочка». Печать на ризографе.
Усл. печ. л. 6,045. Уч.-изд. л. 6,5. Тираж 80 экз.
Заказ № 113.



Издательство ПГУАС.
440028, г. Пенза, ул. Германа Титова, 28.