В.Л.Артюхов,
Г.А.Копейкин,
А.Ашалыто

Настраиваемые модули для управляющих логических устройств

Ленинград
Энергоиздат
Ленинградское отделение
1981

ББК 32.97
А 86
УДК 681.325.6

Рецензент В.Л.Генкин

Артюхов В.Л. и др.
А86 Настраиваемые модули для управляющих логических устройств/Артюхов В.Л., Копейкин Г.Н., Шалыто А.А. – Л.: Энергоиздат. Ленингр.отд-ние, 1981. – 168 с., ил.
60 к.

Приведены новые методы построения и рационального использования настраиваемых логических модулей из функциональных элементов и элементов с двусторонней проводимостью. Найдены оценки сложности схемных реализаций. Предлагаемый подход распространен на цифровые интегральные схемы и релейно-контактные элементы, серийно выпускаемые промышленностью. Впервые рассматриваются вопросы унификации нерегулярных комбинационных схем.
Книга предназначена для специалистов в области в области проектирования логических элементов и устройств, а также для студентов и аспирантов соответствующих специальностей.

А $\frac{30502-115}{051(01)-81}$ 162-81(Э). 2405000000

ББК 32.97
6Ф7

Мы надеемся, что, прочитав эту книгу, читатель сможет:

а) имея список требуемых логических формул, создать схему наиболее простого модуля, который путем настройки можно заставить «выполнять» любую формулу из этого списка;

б) строить логические схемы из настраиваемых модулей, наиболее полно используя их функциональные возможности;

в) объективно оценивать логическую эффективность логических элементов и их серий.

Те, кто сталкивается с подобного рода задачами в своей повседневной работе, вряд ли нуждаются в дополнительных разъяснениях и поэтому могут сразу приступать к изучению гл. 2. Однако мы рассчитываем и на интерес тех читателей, которые только начинают заниматься проблемами проектирования дискретных систем или испытывают затруднения в связи с появлением в дискретной технике массы новых сведений. Именно их вниманию предлагается материал, изложенный во введении в гл. 1.

Настоящая книга посвящена методам построения настраиваемых модулей, применяемых в управляющих логических устройствах. В более узком плане, речь в ней пойдет о модулях комбинационного типа и схемах преимущественно с параллельной обработкой информации.

У читателя - проектанта систем управления может возникнуть серия вопросов: «Мы в настоящее время создаем свои устройства на серийно выпускаемых элементах. Предлагаете ли вы перейти на какие-то неизвестные нам «настраиваемые модули»? Где эти модули приобрести? Какие преимущества они обеспечат?»

На самом деле «настраиваемые модули» - это не какие-то небывалые, новые элементы. Все проектанты стремятся к тому, чтобы не строить систему из элементов «россыпью», а применять заранее сделанные заготовки или сборки, которые и названы здесь модулями.
В большинстве случаев модули реализуются из серийно выпускаемых промышленностью элементов и входят в состав микросборок, субблоков, кассет. Модули имеют некоторое число дополнительных наружных выводов. Назначение этих выводов состоит в том, чтобы сделать схему более универсальной. А это, в свою очередь, необходимо для того, чтобы сократить номенклатуру схемно-конструктивных единиц и добиться их унификации.

В последнее время модули все чаще приобретают законченное конструктивное исполнение. Примером тому являются модули, разработанные по программе стандартизации радиоэлектронной аппаратуры ВМС США [50].

Авторы предвидят также другой вопрос: «Сейчас все шире внедряются вычислительные машины, а вы пишите о логических устройствах с параллельной обработкой информации. Может быть, лучше и проще перейти к использованию управляющих логических машин (УЛМ) или цифровых вычислительных машин (ЦВМ), чем строить дискретные специализированные схемы?»

По нашему мнению, программный и схемный принципы обработки информации еще долго будут сосуществовать, не отрицая, а дополняя друг друга. Область работы авторов, а именно промышленная автоматика, такова, что нам ближе асинхронные параллельные автоматы. На их базе получены основные результаты, излагаемые в книге. Однако это не исключает применения этих результатов при построении УЛМ, микропроцессоров и других устройств с последовательной обработкой информации ввиду того, что при их построении и использовании должны применяться те же основные принципы, что и для предлагаемых модулей: универсальность, максимальное использование. функциональных возможностей, простота построения, многофункциональность и настраиваемость.

Цель приводимого ниже краткого обзора развития дискретной автоматики состоит в том, чтобы ввести читателя в круг проблем, связанных с созданием и применением настраиваемых логических модулей.

Дискретная, или релейная, автоматика существует очень давно. Не будем останавливаться на хитроумных механических приспособлениях, предложенных в древности. Отметим, что развитие дискретной автоматики, в современном смысле слова, началось с изобретения электромагнитного реле в первой половине XIX века. Релейно-контактные схемы нашли применение на автоматических телефонных станциях, на железных дорогах, во всевозможных устройствах защиты, пуска и блокировки. Их использование расширилось в 50-е годы нашего века, чему способствовали возросший спрос на автоматизацию различных промышленных процессов, успехи технологии, приведшие к созданию надежных малогабаритных реле, а также развитие теории синтеза контактных логических

сетей. Особую роль в СССР сыграли исследования члена-корреспондента АН СССР М. А. Гаврилова, чья книга [13] до сих пор является одной из наиболее полных и лучших в этой области.

В начале 60-х годов стала интенсивно развиваться бесконтактная релейная техника. К этому времени выяснилось, что ряд полупроводниковых и магнитных элементов ведут себя подобно контактам реле, т. е. могут находиться в двух стабильных состояниях: открытом и закрытом. При этом они значительно превосходили контактные элементы по быстродействию и допустимому числу переключений, а также по некоторым электрическим характеристикам.

В это время были созданы также известные промышленные серии бесконтактных логических элементов, такие, как «Логика-Т», магнитно-вентильные элементы и др. [45]. Дальнейшим шагом явилось создание микросхем и больших интегральных схем.

Однако ожидавшегося полного вытеснения контактных элементов не произошло вследствие преимуществ, которые они обеспечивали: простоты реализации схем, удобства стыковки с источниками информации и командоаппаратами, возможности коммутации мощных сигналов, простоты схем электропитания, наличия гальванической развязки цепей.

Сейчас все чаще говорят о смешанном элементном базисе, включающем в себя микросхемы для реализации логических операций, контактные элементы для ввода и вывода информации, выполнения ряда простых блокировок и гальванических развязок, а также ферритовые и полупроводниковые «объемные» элементы, осуществляющие длительное запоминание, промежуточное усиление и некоторые другие необходимые функции.

Микроминиатюризация элементов привела к дальнейшему росту объема автоматизации, так как появилась возможность создавать малогабаритные устройства программного управления, дискретные регуляторы, цифровые измерительные и преобразовательные схемы.

Если раньше ЦВМ строились на электровакуумных приборах, а системы дискретной автоматики - на реле, то с появлением бесконтактных логических элементов, и в особенности микросхем, создалась единая элементная база для построения ЦВМ и устройств промышленной автоматики. При этом одни и те же программы управления можно реализовать как с помощью специализированных асинхронных автоматов, так и посредством ЦВМ, причем их основные отличия состоят в методах программирования и выполнения логических операций. Если в асинхронных автоматах алгоритмы работы задаются схемами соединения логических элементов, а обработка информации ведется, как правило, параллельно, то в ЦВМ обычно используется некоторая универсальная схема соединения элементов. Алгоритм работы записывается в специальное программное устройство, а обработка информации ведется последовательно. Этот же принцип заложен в основу построения УЛМ и появившихся в последнее время микропроцессоров.

Со схемной точки зрения ЦВМ обладает более упорядоченной структурой, так как она содержит большое число устройств регулярного типа (регистры, счетчики, дешифраторы и т. д.) [6].

Проигрывая в гибкости и универсальности, специализированные автоматы обладают рядом ценных преимуществ по сравнению с ЦВМ: большим быстродействием при решении крупных массивов логических уравнений и меньшими габаритами, если выполняемые функции не слишком сложны.

Следует отметить также, что применение параллельных структур в промышленной автоматике позволяет в большинстве случаев обеспечить значительный выигрыш по надежности и долговечности. Действительно, объекты автоматизации часто содержат большое число исполнительных механизмов *(ИМ),* работающих взаимосвязано, но не одновременно. Если в ЦВМ или УЛМ один и тот же блок (например, процессор) обслуживает все *ИМ,* то в рассматриваемых автоматах эта нагрузка распределена между отдельными параллельными цепями. При этом у каждой из них ответственность меньше, а временной режим легче.

Представляется вероятным, что последовательный и параллельный принципы обработки информации будут и в дальнейшем применяться в сочетании друг с другом.

Область промышленной автоматики развивалась в соответствии с веяниями времени. В ней были и повальное увлечение бесконтактной техникой, и возврат к контактной, и споры вокруг микроэлектроники, и борьба между телемеханическими и радиальными принципами передачи информации, и, конечно, между параллельной («зачем делать сложно то, что можно сделать просто?») и последовательной («зачем делать по-разному то, что можно сделать одинаково?») структурами логических устройств.

Однако даже в самых противоположных технических решениях сохранялись с замечательным постоянством две тенденции развития: а) от схем из малых компонентов - к схемам повышенного уровня функциональной и конструктивной интеграции; б) от индивидуальных устройств - к

устройствам широкого применения. Указанные тенденции наблюдались на каждом этапе совершенствования элементной базы, структур и принципов построения систем дискретной автоматики.

Обратимся к примерам. В начале 60-х годов создавался набор магнитно-вентильных элементов [45]. Каждая элементарная ячейка представляла собой инвертор, выполняющий функцию «2 ИЛИ - НЕ». Известно, что, имея в номенклатуре единственный тип ячейки «2 ИЛИ - НЕ», можно, пользуясь теоремой о функциональной полноте [43], построить любую комбинационную схему. Поэтому казалось заманчивым обойтись ячейкой одного типа, и подобное предложение было подробно проработано.

Рассматривался также другой вариант - заранее сформировать из элементарных ячеек более крупные элементы, например «2 ИЛИ», «3 ИЛИ», «2 И», «3 И», «Запрет» и др. Это расширяло номенклатуру и приводило к сокращению межэлементного монтажа и уменьшению габаритов проектируемых устройств.

Для сокращения номенклатуры можно пойти на некоторую избыточность, вспомнив, что один крупный элемент может заменить несколько типов более мелких элементов. В частности, в магнитно-вентильной серии был предложен элемент, реализующий функцию

$$y = (x_1 \vee x_2 \vee x_3)(x_4 \vee x_5)x_6\overline{x_7}\,\overline{x_8}.$$

Выбор функции, описывающей элемент, оказался удачным, так как такой элемент реализует большое число подфункций от числа переменных, меньших восьми. Чтобы получить эти подфункции, необходимо на некоторые входы подать константы О или 1 либо приравнять входные сигналы друг другу. Элемент с приведенной выше функцией являлся «многофункциональным» или «настраиваемым», хотя в то время эти термины почти не употреблялись.

При использовании микроэлектронной технологии также существуют три варианта выбора номенклатуры серии: а) один элемент «2 И - НЕ»; б) несколько специализированных элементов повышенной интеграции; в) один или несколько многофункциональных или настраиваемых элементов. В этом случае преимущества от повышения уровня интеграции еще выше, так как технология межвентильного монтажа в пределах корпуса значительно более совершенна по сравнению с технологией межкорпусного монтажа.

Однако с повышением уровня интеграции число типов элементов стремительно растет. Воспользуемся такой аналогией: предположим, что завод, выпускающий типографские шрифты, состоящие из отдельных букв, захочет делать целые слова. Тогда ему вместо 33 наименований продукции придется выпускать десятки и сотни тысяч наименований (в таком положении находится, кстати, производство шрифтов в странах, пользующихся иероглифами). Повысим уровень интеграции еще больше - до фраз, состоящих из нескольких слов. С такой номенклатурой уже невозможно справиться никакому производству.

С логическими элементами получается подобная картина. Известно, что число функций от n переменных составляет

$$\phi(n) = 2^{2^n}$$

Это означает, что, например, при пяти входах может существовать

$$\phi(5) = 2^{32}$$

различных элементов. Вот и попробуйте угадать, какие из них стоит запускать в серийное производство. Проблема схемной интеграции логических устройств является очень важной, так как ее неудовлетворительное решение препятствует полному использованию достижений технологии. Мы можем в одном корпусе разместить сложнейшую схему, но не знаем какую, и в итоге в большинстве случаев размещаем в нем несколько простых не связанных между собой вентилей!

Поиски решения указанной проблемы ведутся в разных направлениях, и в том числе на путях, предусматривающих пересмотр традиционных методов построения систем управления. Но все предложения в своей основе используют многофункциональные или настраиваемые модули либо в роли ячеек однородной среды, либо в качестве микропроцессоров, либо, наконец, в виде самостоятельных схем для построения более сложных устройств.

Иногда многофункциональные модули выпускаются промышленностью в виде микросхем, как это имеет место, например, в серии 108 [44, 51]. Чаще же они формируются из серийно выпускаемых элементов непосредственно предприятиями - разработчиками систем управления и конструктивно оформляются в вида микросборок или на платах печатного монтажа.

По сравнению со специализированными устройствами настраиваемые модули более универсальны, но обладают определенной избыточностью - при любой конкретной настройке часть их внешних выводов или внутренних элементов могла бы быть исключена.

Из сказанного ясно, в каких случаях принимается решение о применении (формировании) настраиваемых модулей. Делается это на основе рассмотрения структур создаваемых систем путем выделения в них схем массового и редкого применения. Первые из них целесообразно оставить специализированными, а вторые заменить небольшой номенклатурой модулей, настраиваемых на реализацию различных схем. Задача здесь состоит в том, как привести различные схемы «к общему знаменателю», т. е. найти такую «покрывающую» схему, из которой они получались! бы путем настройки. Эта схема должна иметь минимальное число дополнительных выводов и простую реализацию, так как никому не нужно универсальное устройство, являющееся механической суммой объединяемых им схем.

Предположим теперь, что настраиваемый модуль создан. Возникает вопрос, как им пользоваться при синтезе различных сложных устройств, так как при неправильном его применении очень легко растерять его богатые функциональные возможности. Опыт создания и применения настраиваемых модулей убедительно свидетельствует: вместе с модулями должна создаваться и четкая простая инструкция по их использованию. Занимаясь разработкой такого рода инструкции, авторы обнаружили, что она при годна не только для специально созданных настраиваемых модулей, но и для обычных, широко применяемых серий микросхем [51].

Попытаемся обобщить сказанное. Итак, схемы дискретной автоматики рациональнее строить из крупных модулей, чем из отдельных элементов. Для этого надо уметь разрабатывать такие модули и знать, как ими потом пользоваться. В одном модуле пи. 1с ню объединить целый ряд функций, если это не связано со слишком большой избыточностью. Такой настраиваемый модуль пригоден для построения самых различных логических схем. Наконец, умея пользоваться настраиваемыми модулями, можно обобщить. эти методы и на элементы, не объединенные в модули. Насколько авторам удалось справиться с изложенными задачами, можно будет судить из последующих глав книги. Первая из них содержит более подробные сведения о структуре, составе, особенностях технической реализации и алгоритмах работы управляющих логических устройств. Вторая объясняет, какие именно настраиваемые модули надо формировать. Третья - как их формировать из функциональных элементов, с тем, чтобы модули получились хорошими по выбранным критериям. В четвертой те описано, как пользоваться этими модулями. В пятой все и (ложечное обобщено и распространено на цифровые интегральные микросхемы широкого применения. Шестая глава отдает дань одному из «экзотических» направлений в теории логиче-14 синтеза - однородным структурам. Седьмая посвящена модулям из элементов с двусторонней проводимостью - их построению и применению. К таким элементам относятся и электромагнитные реле, к которым, вопреки моде, авторы питают глубокую привязанность.

Материал книги формировался в результате работ авторов области проектирования управляющих логических устройств и настраиваемых модулей для их построения, а также на основе читаемых авторами курсов лекций на курсах повышения квалификации.

Авторы выражают искреннюю благодарность за внимание и помощь, оказанные им, докт. техн. наук, проф. И. В. Прангишвили, докт. техн. наук, проф. В. Г. Лазареву, докт. техн. наук, проф. В. А. Кондрашову, докт. техн. наук, проф. И. Р. Фрейдзону, канд. техн. наук Л. М. Фишману и канд. техн. наук Л. Я. Розенблюму. Авторы пользуются также случаем поблагодарить многочисленных слушателей курсов повышения квалификации за то, что они заставляли нас отвечать на такие вопросы, которые мы предпочли бы обойти, но без ответов на которые вряд ли родилась бы эта книга.

Замечания и пожелания по книге просьба присылать по адресу: 191041, Ленинград, Марсово поле, 1, Ленинградское отделение Энергоиздата.

Авторы

Логическое управление в системах автоматики

1-1. ЗАДАЧИ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ. СТРУКТУРА И ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ ЛОГИЧЕСКИХ УСТРОЙСТВ

Логическим называется управление с помощью сигналов, принимающих конечное число фиксированных значений, осуществляемое в соответствии с заданной программой (алгоритмом). Обычно используют двухуровневые, или двоичные, сигналы. Один уровень обозначают условно 0, а другой - 1.

Примерами управляющих логических устройств могут служить распространенные в промышленной автоматике системы пуска и останова различных агрегатов, автоматического аварийного управления и защиты, дистанционного управления, сигнализации и контроля [1, 54].

Наряду с использованием в составе больших систем, логические устройства применяются также и при решении разного рода частных задач, например, в специализированных преобразователях, обычно выполняющих алгебраические, геометрические или табличные преобразования.

Логические устройства могут существовать как в виде законченных приборов, так и входить в состав систем регулирования в сочетании с аналоговыми схемами, осуществляя те или иные переключательные функции.

Наиболее общая модель логического устройства, или автомата, имеет *n* двоичных входов $(x_1, x_2, ..., x_n)$ и *m* двоичных выходов $(y_1, y_2, ..., y_m)$, т. е. представляет собой (n,m) - полюсник. Сигналы, поступающие на входы автомата, образуют множество входных воздействий *X,* а сигналы, вырабатываемые на его выходах, - множество выходных реакций Y. Входные воздействия поступают от источников информации (*ИИ*), к которым относятся кнопки, ключи, тумблеры, а также сигнализаторы состояния внешней среды и управляемого объекта. Источниками информации могут являться также выходы других дискретных устройств.

Определение состава и характеристик *ИИ* должно основываться на глубоком изучении автоматизируемого процесса. При их выборе обычно, сознательно или интуитивно, руководствуются принципом максимального воздействия *ИИ* на автоматизируемый процесс. Иными словами, в первую очередь выбирают такие ИИ, изменение состояния которых наиболее явно и непосредственно сказывается на изменении состояний исполнительных механизмов (*ИМ*). Особенно легко указанный принцип прослеживается на примере выбора органов управления на пульте оператора. Каждый ключ, кнопка, тумблер имеют надпись, характеризующую то действие, которое должно произойти при воздействии на них, если этому не помешают блокировки, осуществляемые системой автоматики. Аналогично и другие *ИИ* выбираются так, чтобы последствия их срабатывания можно было предвидеть.

Принцип максимального воздействия не всегда осуществим, так как ряд «ключевых», с точки зрения автоматизируемого процесса, параметров не может быть замерен непосредственно, например, вследствие отсутствия соответствующих приборов. В этих случаях приходится прибегать к косвенным измерениям или вычислениям, вводя вместо одного *ИИ* их совокупность, в целом дающую необходимую информацию. Иногда от принципа максимального воздействия сознательно отходят, если опасаются нарушений в управлении при ложном срабатывании одного или нескольких *ИИ.* В этих случаях прибегают к прямому либо косвенному дублированию входных данных. Наконец, в особенности в специализированных преобразователях, мы не всегда обладаем возможностью выбирать различные варианты задания входной информации. Однако в целом принцип максимального воздействия является довольно универсальным. Непосредственное следствие его применения состоит в простоте получающихся логических устройств, так как число элементов в схеме, *ИИ* которой выбраны по принципу максимального воздействия, зависит от числа входов по закону, близкому к линейному, в то время как сложность произвольной схемы определяется показательным законом [29].

Выходные реакции Y выдаются схемой на объекты управления - исполнительные механизмы, к которым относятся всевозможные пусковые устройства, средства сигнализации, запоминающие устройства и т. п. Основными характеристиками *ИМ* являются: позиционность и наличие либо отсутствие памяти.

Позиционность характеризует число устойчивых состояний, в которых может находиться *ИМ.* В дальнейшем мы будем рассматривать *ИМ,* имеющие два устойчивых состояния («открыто - закрыто», «включено - отключено»).

ИМ, не обладающий памятью, находится в активном (включенном) состоянии до тех пор, пока на его вход действует командный сигнал, и переходит в пассивное (отключенное) состояние при снятии этого сигнала. *ИМ,* обладающий памятью, сохраняет свое состояние после снятия управляющего воздействия. Для таких *ИМ* требуются два управляющих сигнала - «Пуск» и «Стоп» либо «Запись» и «Стирание», каждый из которых подается на свой вход. Разрешающий сигнал («Пуск», «Запись» и т. п.) будем обозначать индексом 1; например, сигнал y_{i1} означает команду на пуск i-го *ИМ.* Запрещающий сигнал («Стоп», «Стирание» и т. п.) обозначим индексом 0; например, y_{i0} - команда на останов. При этом необходимо различать командные (разрешающие и запрещающие) и информационные сигналы (y'_{i1} и y'_{i0}), которые появляются на выходе сигнализаторов положения *ИМ* после отработки командных сигналов.

Автомат преобразует в соответствии с заданным алгоритмом набор входных воздействий X в набор выходных реакций Y. В зависимости от вида реализуемого алгоритма различают однотактные (комбинационные) и многотактные (с памятью) автоматы. У первой разновидности автоматов выходная реакция в данный момент времени определяется только текущим значением набора входных воздействий и не зависит от того, какие сигналы и в каком порядке подавались на входы или были на выходах автомата в предшествующие моменты времени, а у второй - выходная реакция зависит также от предшествующих значений входных воздействий или выходных реакций.

Запоминание предшествующих значений аппаратурно реализуется совокупностью обратных связей, содержащих элементы «память» (*Я*) и «выдержка времени» *(ВВ).* Эти обратные связи могут также замыкаться через объект управления или внешнюю среду.

Рассмотрим более подробно структуру систем логического управления. Важнейшей особенностью этой структуры является то, что схемы включения и отключения *ИМ* обычно совмещают с элементами памяти, которые выполняются при этом механическим путем. Это делается с той целью, чтобы при исчезновении напряжения питания не возникали аварийные ситуации. Таким образом, в системах рассматриваемого класса имеется память, распределенная по объектам управления, которая поэтому не может быть минимизирована [20]. В большинстве случаев для таких систем характерно также и то, что соответствующие им алгоритмы функционирования можно реализовать как при помощи автоматов с памятью, так и комбинационными схемами за счет введения входных сигналов от дополнительно устанавливаемых сигнализаторов положения и параметров [21]. В системах промышленной автоматики алгоритмы работы обычно реализуются вторым способом - путем почти повсеместного использования комбинационных схем. Для таких систем характерно, что комплекс «система управления - объект управления» может вести себя как многотактное устройство, несмотря на то, что собственно система управления практически однотактна. Это объясняется наличием обратных связей, охватывающих систему управления совместно с объектом управления. Эти обратные связи содержат механические элементы (в частности, управляемую арматуру), которые исключают замыкание контуров обратной связи для электрических сигналов. Поэтому сигнализаторы положения рассматриваются в этих системах как собственно *ИИ* наряду с органами управления и сигнализаторами параметров.

Таким образом, можно сделать вывод, что для систем промышленной автоматики важнейшие задачи в области логического синтеза связаны с построением комбинационных схем. Однако построение этих схем в данном случае обладает рядом особенностей. Во-первых, основная проблема синтеза асинхронных автоматов, к классу которых обычно относятся эти системы, - проблема устранения состязаний - в данном случае не стоит достаточно остро, так как *ИМ* обладают большой инерционностью; а во-вторых, это свойство *ИМ* резко снижает требования к быстродействию схем. Указанные ограничения упрощают построение комбинационных схем в системах рассматриваемого класса, например, по сравнению с построением соответствующих схем, используемых в вычислительной технике.

1-2. СПОСОБЫ ЗАДАНИЯ АЛГОРИТМОВ РАБОТЫ УПРАВЛЯЮЩИХ ЛОГИЧЕСКИХ УСТРОЙСТВ

Не составляет секрета, что большинство реально созданных устройств логического управления построено непосредственно по словесному описанию алгоритмов их работы без использования формализованных методов. Последовательность проектирования их обычно такова: техническое задание в виде текста - формулы или функциональные схемы - принципиальные схемы. Формализованные методы, основанные на использовании автоматных таблиц, графов переходов и т. п., применяются лишь в некоторых наиболее сложных случаях, как правило, для построения и минимизации отдельных небольших подсхем. Объясняется это в значительной степени тем, что сложность описания при использовании упомянутых формализованных подходов определяется числом входов и состояний синтезируемого автомата и почти не зависит от сложности алгоритма и соответственно от сложности предстоящей схемной реализации. Так, размер таблицы состояний автомата, имеющего n входов и содержащего s триггеров, будет $2^n \cdot 2^s$, причем одинаковым как для самого простого, так и для самого сложного устройства.

Идеальным представляется случай, когда сложность описания алгоритма пропорциональна сложности его предстоящей схемной реализации. Забегая несколько вперед, отметим, что логические формулы обладают именно таким свойством: число букв в них линейно связано с числом элементов, требующихся для реализации этих формул. Но как написать формулу, минуя стадию составления таблиц? В ряде случаев это возможно непосредственно по словесному описанию. Однако иногда по словесному описанию записать формулу чрезвычайно трудно.

Из сказанного вытекают требования, которым должен отвечать формализованный язык для описания алгоритмов работы дискретных управляющих устройств: иметь структуру, близкую к структуре естественного языка; сложность описания на этом языке должна быть связана линейной зависимостью со сложностью предстоящей схемной реализации; допускать простой, однозначный и строго формализованный переход к логическим формулам.

Указанным требованиям в большей или меньшей степени отвечают так называемые и м п л и к а ц и о н н ы е я з ы к и, в основе которых лежит логическая конструкция «если x, то y». Они продуктивно применяются в программировании, но при проектировании логических схем делаются лишь первые попытки их использования.

Ниже излагаются основные идеи, заложенные при разработке двух языков, ориентированных на решение задач логического синтеза: языка «с е к в е н ц и й» и языка «у с л о в н ы х с е к в е н ц и й».

Прежде чем перейти к их изложению, остановимся на анализе первого из изложенных выше требований. Что имеется в виду под структурой, близкой к структуре естественного языка (словесного задания)? Чем естественный язык отличается от логической формулы? Основных отличий два: словесное описание позволяет задавать работу устройства (группы устройств) по частям, в то время как логическая формула требует указания всех условий срабатывания ИМ сразу; словесное описание ориентировано обычно на задание условий работы от входа к выходу, т. е. от причины к следствию, в то время как в логических формулах следствие предшествует причине.

Следовательно, формализованный язык, имеющий структуру, близкую к словесной, должен быть ориентирован на запись условий работы от причины к следствию и позволять записывать эти условия по частям. Последующий переход к логической формуле должен состоять в объединении этих частей по определенному правилу.

Секвенции. При использовании языка секвенций каждая составная часть сложного логического высказывания имеет вид «если $\varphi(x)=1$, то $\psi(y)=1$», что записывается с помощью знака «секвенция» следующим образом:

$$\varphi(x) \vdash \psi(y) \qquad (1\text{-}1)$$

Написанная выше секвенция означает, что при $\varphi(x)=1$ функция $\psi(y)$ также равна 1, но ничего не говорит о том, чему (0 или 1) равна $\psi(y)$ при $\varphi(x)=0$.

Составим функцию $z(\varphi;\psi)$, обращающуюся в единицу на наборах, не противоречащих секвенции $\varphi \vdash \psi$, и в нуль - на наборах, ей противоречащих (табл. 1-1).

Отметим, что ситуация, задаваемая третьей строкой таблицы, является единственной запрещенной выражением $\varphi \vdash \psi$ ситуацией.

По таблице нетрудно записать, что

$$z = \overline{\varphi} \vee \psi , \qquad (1\text{-}2)$$

т. е. функция $z(\varphi;\psi)$ представляет собой импликацию [43].

Возможность сопоставления секвенций и импликаций облегчает выполнение математических преобразований над секвенциями, так как они сводятся к операциям, характерным для хорошо известного базиса $\{\&, \vee, \overline{\ }\}$.

Хотя в правой части секвенции возможна любая функция φ, в практике их использования ограничиваются случаем, когда ψ представляет собой конъюнкцию выходных переменных. Такие секвенции называют правильными.

Таблица 1-1

Таблица истинности

φ	ψ	z
0	0	1
0	1	1
1	0	0
1	1	1

Если одной секвенции соответствует импликация, то системе секвенций - произведение импликаций. Это утверждение следует из того факта, что две системы секвенций эквивалентны, т. е. описывают один и тот же алгоритм, если произведения соответствующих им импликаций равны. Частный, но весьма важный в практическом отношении случай эквивалентности устанавливается теоремой разделимости: *системе правильных секвенций можно поставить в соответствие эквивалентную ей систему, у которой в правой части каждой секвенции содержится только одна переменная* y_i *из множества Y, а левая часть секвенции, соответствующей* y_i*, представляет собой дизъюнкцию левых частей тех исходных секвенций, в правые части которых входила* y_i. В разделенной системе знак секвенции можно заменить знаком равенства. Отсюда получается простое правило перехода от системы секвенций к системе логических формул: *для каждой выходной переменной надо сложить левые части тех секвенций, в которые она входит.*

Пример: Для системы секвенций вида

$$x_1 \vdash y_1 y_2 y_3;$$
$$\overline{x_2}\,\overline{x_3} \vdash y_1 y_2;$$
$$\overline{x_1} x_2 \vdash y_1 y_3;$$
$$x_4 \vdash y_2$$

определить соответствующую систему формул.

Пользуясь изложенным выше правилом, получим систему формул вида

$$y_1 = x_1 \vee \overline{x_2}\,\overline{x_3} \vee \overline{x_1} x_2 = x_1 \vee x_2 \vee \overline{x_3};$$
$$y_2 = x_1 \vee \overline{x_2}\,\overline{x_3} \vee x_4;$$
$$y_3 = x_1 \vee \overline{x_1} x_2 = x_1 \vee x_2.$$

Из теоремы разделимости следует, что от системы секвенций возможен однозначный переход к логическим формулам в дизъюнктивной нормальной форме (ДНФ). При этом каждая секвенция является, как минимум, одним из членов ДНФ. Отсюда ясен выигрыш, который дает применение языка секвенций: вместо логических формул в целом мы можем задавать отдельные слагаемые ДНФ. Теорема разделимости устанавливает, как эти частичные условия объединять в полные.

Задание программы работы в виде совокупности отдельных частей сопряжено с риском: не противоречат ли они друг другу? Противоречия возникают, если в правых частях присутствуют прямые и инверсные значения y_i. Признаком противоречия между условиями для y_i и $\overline{y_i}$ является то, что произведения левых частей соответствующих секвенций не обращаются в нуль.

Язык секвенций предусматривает наличие у *ИМ* памяти. Для того чтобы показать это, рассмотрим систему

$$\left.\begin{array}{l} f(x)\vdash y; \\ \varphi(x)\vdash \overline{y}. \end{array}\right\} \qquad (1\text{-}3)$$

Предположим, что она непротиворечива. Это означает, что $f(x)\varphi(x)=0$, т. е. если присутствует сигнал $\varphi(x)$, приводящий к $y=0$, то сигнал $f(x)$, приведший к $y=1$, в это время должен отсутствовать. Если бы при исчезновении сигнала $f(x)$ сигнал y обращался бы также в нуль, то необходимости в подаче сигнала $\varphi(x)$ не было бы. Отсюда следует, что при прекращении подачи $f(x)$ сигнал y сохраняет значение единицы, т. е. имеет место память. То, что язык секвенций предполагает наличие памяти у *ИМ,* ограничивает его применимость так называемыми автоматами с распределенной памятью. Необходимо отметить, что в рассматриваемом языке инверсия входной переменной означает сигнал на останов *ИМ, а* не сигнал запрета пуска *ИМ.* Если мы управляем совокупностью *ИМ,* часть из которых памятью не обладает, то существующая форма секвенциальной записи не содержит информации об этом, что, несомненно, является недостатком. Вместе с тем язык секвенций по своей конструкции во многом отвечает изложенным выше требованиям. Он, по мнению авторов, составляет хорошую основу для формализации логического синтеза, однако нуждается в усовершенствовании. Во-первых, желательно, чтобы язык записи алгоритмов был единым как для автоматов с распределенной памятью, так и для автоматов, *ИМ* которых памятью не обладают. Во-вторых, представляется полезным еще больше приблизить структуру языка к словесному описанию, т. е. задавать программу работы не в виде слагаемых ДНФ, а более мелкими частями. Наконец, следует предусмотреть правила, сводящие к минимуму возможную противоречивость *в* описании. Конечной целью в этом направлении является создание формализованного языка, позволяющего по словесному алгоритму, «под диктовку», записывать частичные логические условия и корректно объединять их в формулы. В определенной степени изложенным требованиям отвечает язык условных секвенций.

Условные секвенции. Условной секвенцией называется выражение вида

$$\varphi(x)\succ K(y) \qquad (1\text{-}4)$$

где $\varphi(x)$ - некоторая функция от группы входных переменных; $K(y)$ - конъюнкция некоторой группы выходных переменных.

Это выражение означает, что если $\varphi(x)=1$, то y_i, входящее в конъюнкцию $K(y)$ без отрицания, равно единице, если отсутствуют запрещающие условия.

Запрещающим условием для y_i является совокупность тех условных секвенций из числа образующих систему, в правые части которых y_i входит с отрицанием. При этом отметим, что в отличие от языка секвенций в данном случае инверсия выходной переменной означает не сигнал на останов, а запрещение пуска *ИМ.*

Очевидно, что система условных секвенций может содержать большое число условий. При этом никакая формализованная процедура не дает гарантии, что все условия приведены, и поэтому в качестве «расписки» заказчика в содержательной полноте системы следует записать оператор «Конец». Он свидетельствует о завершенности задания условий работы системы. Далее, необходимо указать начальное состояние *ИМ,* выделив его оператором «Начало».

Приведем правило перехода от системы условных секвенций к системе логических формул:

1) для каждой выходной переменной, имеющей исходное состояние $y_i=0$, сложить левые части тех условных секвенций, в которые она входит без отрицания;
2) полученную сумму умножить на произведение отрицаний левых частей тех секвенций, в которые y_i входит с отрицанием;
3) для выходных переменных с исходным состоянием $y_i=1$ следует ограничиться вторым пунктом настоящего правила, т. е. записать *y,* как произведение отрицаний левых частей тех секвенций, в которые y_i входит с отрицанием.

П р и м е р . Пусть задана следующая система условных секвенций. «Начало»:
$y_1=0, y_2=0, y_3=1.$

Программа:

$$x_1 \succ y_1 y_2;$$
$$x_2 \bar{x}_3 \succ \bar{y}_1 \bar{y}_2;$$
$$\bar{x}_1 x_2 \succ y_2 \bar{y}_3;$$
$$x_4 \succ \bar{y}_1$$

«Конец».

Тогда, руководствуясь изложенным правилом перехода, получим систему формул

$$y_1 = x_1 \overline{\bar{x}_2 \bar{x}_3} \bar{x}_4 = x_1 (\bar{x}_2 \vee x_3) \bar{x}_4;$$
$$y_2 = (x_1 \vee \bar{x}_1 x_2) \overline{\bar{x}_2 \bar{x}_3} = (x_1 \vee x_2)(\bar{x}_2 \vee x_3);$$
$$y_3 = \overline{\bar{x}_1 x_2} = x_1 \vee \bar{x}_2.$$

Язык условных секвенций позволяет описывать работу устройств, охваченных обратными связями.

П р и м е р . Построить схему, имеющую два входа x_1 и x_2 и один выход y, работающую по следующей программе. При подаче сигнала x_1 на выходе появляется сигнал y, который запоминается, а при подаче сигнала x_2 этот сигнал исчезает.

Записывая словесный алгоритм на языке условных секвенций, получим:

«Начало»: $y = 0$.

Программа:

$$x_1 \succ y;$$
$$y \succ y;$$
$$x_2 \succ \bar{y}$$

«Конец». Отсюда

$$y = (x_1 \vee y)\bar{x}_2;$$

Нетрудно видеть, что при составлении формулы приоритет был отдан сигналу x_2, т. *е.* считалось, что

$$x_1 x_2 \succ \bar{y}.$$

Возникает вопрос, что делать, если мы желаем отдать предпочтение разрешающему сигналу, т. е. хотим, чтобы при одновременной подаче сигналов x_1 и x_2 выходной сигнал y был равен единице?

Для этой цели в языке условных секвенций предусмотрен оператор «НО». После оператора «НО» следует приводить те сочетания разрешающих и запрещающих переменных, для которых мы хотим сделать исключение с точки зрения приоритета.

П р и м е р . Пусть задана следующая система условных секвенций.

«Начал о»: $y = 0$.

Программа:

$$x_1 \succ y;$$
$$y \succ y;$$
$$x_2 \succ \bar{y};$$
"НО"
$$x_1 x_2 \succ y$$

«Конец».

Как в этом случае перейти к булевой формуле? Для этого необходимо дополнить введенное ранее правило перехода от системы условных секвенций к системе логических формул следующим пунктом: 4) если в системе присутствует оператор «НО» то для каждой выходной переменной к формуле, полученной по условным секвенциям, записанным до оператора, прибавляются формулы, полученные по условным секвенциям, записанным после оператора.

Воспользовавшись этим правилом для рассмотренного выше примера, получим:

до оператора «НО» $y' = (x_1 \vee y)\bar{x}_2$;

после оператора «НО» $y'' = x_1 x_2$.

Сумма

$y = y' \vee y'' = x_1\bar{x}_2 \vee y\bar{x}_2 \vee x_1 x_2 = x_1 \vee \bar{x}_2 y.$

Использование условных секвенций позволяет вести описание алгоритмов не только на микро-, но и на макроуровне. Было бы странным каждый раз заново строить триггеры, счетчики, дешифраторы и другие стандартные ячейки и блоки. Значительно удобнее рассматривать их как своеобразные исполнительные механизмы, одновременно являющиеся *ИИ* для последующих ступеней преобразований. Примеры использования условных секвенций на макроуровне приведены в работе [54].

В заключение отметим, что изложенные языки формализованного представления алгоритмов управляющих логических устройств нуждаются в дальнейшем совершенствовании и развитии, однако уже сегодня они позволяют достаточно просто переходить от словесного описания автомата к логическим формулам. Для перехода от логических формул к схемам также необходимы простые правила перехода, разработке которых, по существу, и посвящены следующие главы.

ГЛАВА ВТОРАЯ

ФУНКЦИОНАЛЬНЫЙ БАЗИС НАСТРАИВАЕМЫХ МОДУЛЕЙ

2-1. НАСТРАИВАЕМЫЕ МОДУЛИ. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Во введении мы отметили важность задачи создания ограниченной номенклатуры схемно-конструктивных единиц повышенного уровня интеграции, а также то, что наиболее сложной является разработка унифицированных модулей для построения комбинационных логических схем с нерегулярной структурой.

Действительно, умея строить, например, трехразрядный счетчик, мы без особого труда построим и двадцатиразрядный счетчик, так как регулярные схемы допускают простое наращивание. В то же время нерегулярные структуры с увеличением числа входов существенно изменяются и не могут рассматриваться как механическая сумма структур с меньшим числом входов.

Задача сокращения номенклатуры логических элементов обычно решается путем отказа от их схемной интеграции. При этом используется подход, основанный на известной теореме Поста - Яблонского о функциональной полноте [43]. Из нее следует, что любое логическое устройство можно построить с применением всего одного типа элементов «И - НЕ» либо «ИЛИ - НЕ».

Несмотря на то что такие элементы входят в состав большинства серий интегральных микросхем, выпускаемых промышленностью, их использование связано с рядом трудностей. Во-первых, проектант при описании алгоритмов работы использует обычно логические связки «И», «ИЛИ», «НЕ», вследствие чего построение функциональных схем обычно выполняется в базисе $\{\&, \vee, \bar{}\}$, называемом булевым. При этом возникает необходимость их последующего перевода в принципиальные схемы из элементов «И - НЕ», «ИЛИ - НЕ». Процедура построения схем в этих базисах значительно более сложна, чем при использовании булева базиса. Это объясняется тем, что для операций «И - НЕ», «ИЛИ - НЕ» не выполняется сочетательный закон

$$x_1 | x_2 | x_3 \neq (x_1 | x_2) | x_3 \neq x_1 | (x_2 | x_3); \qquad (2\text{-}1)$$

$$x_1 \downarrow x_2 \downarrow x_3 \neq (x_1 \downarrow x_2) \downarrow x_3 \neq x_1 \downarrow (x_2 \downarrow x_3), \qquad (2\text{-}2)$$

в то время как этот закон имеет место для операции $\&$ и $\vee$:

$$x_1 \& x_2 \& x_3 = (x_1 \& x_2) \& x_3 = x_1 \& (x_2 \& x_3); \qquad (2\text{-}3)$$

$$x_1 \vee x_2 \vee x_3 = (x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3); \qquad (2\text{-}4)$$

Отметим, что этот закон выполняется и для операции «Сложение по модулю 2»:

$$x_1 \oplus x_2 \oplus x_3 = (x_1 \oplus x_2) \oplus x_3 = x_1 \oplus (x_2 \oplus x_3), \qquad (2\text{-}5)$$

что обеспечивает удобство построения схем также и в базисе $\{\&, \vee, \bar{}, \oplus\}$.

Логические элементы «И - НЕ» и «ИЛИ - НЕ» обладают еще и тем недостатком, что не удовлетворяют требованиям интегральной технологии - имеют большое число выводов, приходящихся на один вентиль. В результате в корпусе микросхемы может быть размещено лишь небольшое число вентилей, несмотря на то что интегральная технология позволяет реализовать сотни и тысячи вентилей на одном кристалле. Поэтому логические возможности интегральных микросхем (ИМС), содержащих элементы «И - НЕ», «ИЛИ - НЕ», относительно невысоки и требуется большое их число для реализации даже достаточно простых комбинационных схем.

Малый уровень интеграции таких элементов, кроме того, приводит к большому объему неунифицированного внешнего монтажа, что резко снижает надежность и увеличивает сложность печатных плат, используемых для размещения ИМС.

Удовлетворить требованиям интегральной технологии, упростить синтез, уменьшить объем внешнего монтажа, а также расширить функциональные возможности при небольшой номенклатуре элементов можно при использовании настраиваемых модулей.

Настраиваемым модулем называют устройство, которое при подаче на его входы внешних воздействий изменяет требуемым образом реализуемые им выходные функции.

При этом входы, связанные с источниками информации, от которых зависит рассматриваемая выходная функция, называют информационными входами модуля, а остальные входы - настроечными.

В настоящее время имеется большое число работ, в которых рассмотрены настраиваемые модули различных типов [9, 18, 19, 44, 58]. Для этих модулей может быть предложена классификация, проводимая по следующим основным признакам:

а) по уровню функциональной и конструктивной интеграции: малого, среднего и большого уровня интеграции;

б) по типу элементной базы: из функциональных элементов, из элементов с двусторонней проводимостью;

в) по типу структуры: с однородной или с неоднородной структурой;

г) по числу выходов: одновыходные, с прямым и инверсным выходами, многовыходные;

д) по степени разделения информационных и настроечных входов: с полным разделением, с частичным разделением, с совмещением множества информационных и настроечных входов;

е) по наличию памяти: автоматы с памятью или комбинационного типа;

ж) по использованию памяти: модули, память в которых используется только для запоминания настройки; модули, в которых память используется как для запоминания настройки, так и в процессе работы;

з) по способу введения информации: с параллельным вводом, с последовательным вводом, с параллельно-последовательным вводом;

и) по типу настройки: с постоянной (необратимой) или с переменной настройкой;

к) по способу переменной настройки, осуществляемой путем: подачи констант 0 и 1; отождествления входов; антиотождествления входов; выбора выхода многовыходного модуля; наложения перемычек на выходные выводы модуля; выбора модуля из набора;

л) по виду объекта настройки: реализующие путем настройки автоматы; реализующие путем настройки булевы функции; настраиваемые на заданные конфигурации схем;

м) по степени универсальности: универсальные или многофункциональные;

н) по разновидностям реализуемых функций и схем: универсальные в классе всех булевых функций; универсальные в некотором классе булевых функций (в некотором классе схем); многофункциональные модули, реализующие путем настройки некоторый набор функций (схем).

В предыдущей главе уже отмечалось, что для построения управляющих логических устройств в основном должны использоваться модули комбинационного типа. Поэтому далее будут рассматриваться модули только этого типа, которые будем называть настраиваемыми логическими модулями (НЛМ).

С другой стороны, ввиду того что при построении логических устройств могут использоваться как функциональные элементы, так и элементы с двусторонней проводимостью, в книге будут рассматриваться методы построения и использования НЛМ на элементах каждой из указанных разновидностей.

2-2. ВЫБОР ФУНКЦИОНАЛЬНОГО БАЗИСА МОДУЛЕЙ

Функция, задающая однозначное соответствие между двоичными наборами входных переменных и двоичными значениями выходной переменной, называется булевой функцией.

Булевы функции задаются обычно двоичными таблицами, называемыми таблицами истинности. При этом каждая из них характеризуется числом входных переменных n, от которого она зависит.

Аналитическая запись функций алгебры логики (ФАЛ), в которой используются символы булевых операций и переменных, называется булевой формулой. Одноместные и двухместные булевы операции, используемые при записи булевых формул, называются базисом этих формул.

Основными характеристиками булевой формулы являются ее базис и число символов входных переменных в ней. В дальнейшем символы входных переменных в булевой формуле будем называть буквами.

Если основной характеристикой ФАЛ является число переменных *n,* то для булевой формулы такой характеристикой является число букв h. *Булева формула называется нормальной, если символы отрицаний расположены в ней лишь над одиночными символами переменных.*

Отметим, что если задана некоторая булева формула из *n* букв, которая не принадлежит классу нормальных формул, то она может быть представлена эквивалентной (соответствующей той же функции) нормальной булевой формулой, содержащей то же число букв. В дальнейшем основное внимание будет уделяться нормальным булевым формулам, так как они используются обычно при описании алгоритмов работы управляющих логических устройств.

Класс функций (формул), реализуемых модулем путем настройки, называют функциональным базисом модуля.

Выбор функционального базиса определяет возможность выполнения ряда требований, которым должны удовлетворять разрабатываемые методы построения и использования НЛМ. Перечислим основные из них: а) возможность построения НЛМ, наиболее отвечающих потребностям класса управляющих логических устройств, для применения в которых они предназначены; б) приспособленность к решению задач большой размерности, т. е. к синтезу систем с многими входами и выходами; в) простота и удобство при «ручном» проектировании и возможность использования при автоматизации синтеза; г) предсказуемость результатов, т. е. гарантированное соответствие их некоторым оценкам сложности. Удовлетворить этим требованиям можно лишь при выборе функционального базиса НЛМ, учитывающем специфику булевых функций, описывающих алгоритмы функционирования управляющих логических устройств.

С целью рассмотрения особенностей этих функций были проанализированы более 2000 ФАЛ, описывающих нерегулярные логические схемы систем управления судовыми техническими средствами [54]. При этом было установлено, что булевы формулы, соответствующие этим функциям, бесповторны либо обладают малой повторностью переменных в булевом базисе.

Напомним, что *булева формула называется бесповторной, если число букв в ней равно числу переменных.* Для определения степени повторности переменных введем коэффициент повторности $\wedge$, вычисляемый как отношение числа букв в формуле h к числу независимых переменных в ней:

$$\wedge = h / n \qquad (2\text{-}6)$$

Бесповторность практически каждой булевой формулы $(\wedge = 1)$ в алгоритмах работы систем управления существует наряду с повторностью переменных в различных формулах одной системы, т. е. для устройств рассматриваемого класса характерна связность формул, количественной характеристикой которой является коэффициент связности, вычисляемый как отношение суммарного числа букв $H = \sum_{i=1}^{N} h_1$ в системе из N формул к суммарному числу независимых переменных $\prod = \sum_{i=1}^{N} n_1$ [54]. Анализ систем формул, соответствующих алгоритмам работы большого числа устройств промышленной автоматики, показал, что средний коэффициент связности для каждой системы имеет небольшое значение и приблизительно равен 2 - 3.

Выводы относительно структуры булевых формул рассмотренного класса систем справедливы также и для других управляющих логических устройств. В работе [1] приведены формулы, описывающие системы управления газотурбинными установками компрессорных станций газопроводов. Рассмотрение формул, описывающих алгоритмы работы турбодетандера, стопорных и регулирующих клапанов,

маслонасосов и кранов, показывает, что каждая из них бесповторна. Этот же вывод можно сделать из анализа формул, описывающих устройства управления оросительными системами [35] и различным технологическим оборудованием [36].

В работе [39] была проведена перепись наиболее распространенных схем автоматов, управляющих соединением в устройствах автоматической телефонии. Выбор этого класса устройств был обусловлен тем, что релейные устройства, используемые в автоматической телефонии, относятся к наиболее массовым и сложным релейно-контактным схемам. При этом было исследовано 1786 цепей, которые описывались 986 различными булевыми формулами. Выяснилось, что каждая из этих формул принадлежит

Таблица 2-1

Процентное соотношение разновидностей булевых функций в устройствах управления ЦВМ

Тип ЦВМ	Класс функций			
	А	Б/А	О/Б	Н/О
"Наири"	96,5	3,2	0,3	-
"Минск-2"	73,4	21,2	0,2	5,2
СЦВМ	71,5	22,0	2,8	3,7
БЭСМ-6	54	8	19	19

хотя бы одному из четырех классов упорядоченного типа: бесповторным формулам; функциям, обладающим групповой инвариантностью; функциям, допускающим разделительную декомпозицию; однородным функциям.

Таким образом, и для этих устройств бесповторность формул является характерным свойством, тем более что и однородные, и разделимые функции обычно имеют малую повторность переменных в соответствующих формулах.

В настоящее время наиболее сложными устройствами логического управления являются устройства управления ЦВМ. Авторы работы [42] исследовали функции, описывающие эти устройства для различных ЦВМ. Результаты этой работы, выраженные в процентах, приведены в табл. 2-1, в которой А обозначает класс бесповторных ДНФ; Б/А - класс бесповторных скобочных формул; О/Б - класс однородных повторных функций и Н/О - класс неоднородных функций.

Таким образом, доля бесповторных формул в устройствах управления указанных ЦВМ является основной и соответственно равна 99,7, 94,6, 93,5 и 62%.

Для выяснения влияния неупорядоченных функций на средний коэффициент повторности авторами были исследованы 155 представителей типов функций, на которые разбиваются 1704 функции, описывающие устройство управления ЦВМ «Минск-2». При этом оказалось, что лишь девять представителей типов не содержат бесповторных формул, причем к ним относятся всего лишь 92 функции. В табл. 2-2 приведены указанные девять представителей и определены соответствующие коэффициенты повторности для каждого из них.

Средний элемент повторности для массива повторных формул

$$\wedge_{ср} = \frac{\sum_{i=1}^{9} \wedge_i N_i}{\sum_{i=1}^{9} N_i} = 1{,}10, \qquad (2\text{-}7)$$

а средний коэффициент повторности для своего массива

$$\wedge_{ср} = \frac{\sum_{i=1}^{155} \wedge_i N_i}{\sum_{i=1}^{155} N_i} = 1{,}005. \qquad (2\text{-}7)$$

Следовательно, и для рассмотренного класса устройств управления характерно, что описывающие их булевы формулы бесповторны или близки к ним. При этом необходимо отметить, что многие формулы бесповторны непосредственно в ДНФ. Так например из табл. 2-1 следует, что в устройствах управления ЦВМ «Наири» и «Минск-2» соответственно 96,5 и 73,4% всех формул бесповторны в ДНФ.

Таблица 2-1

Представители типов повторных формул, описывающих устройство управления ЦВМ «Минск-2»

Представитель типа функций	Коэффициент повторности	Число функций, принадлежащих типу
$(x_1 \vee x_3)\bar{x}_2 \vee x_3$	1,67	1
$x_1(x_3 \vee x_4) \vee \bar{x}_2 x_3$	1,25	1
$x_1\bar{x}_2 \vee x_4(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \vee \bar{x}_5 x_6$	1,34	1
$\bar{x}_1\left[\bar{x}_4\left(\bar{x}_2 \vee \bar{x}_3\right) \vee \bar{x}_2\bar{x}_3\right] \vee x_4 \vee x_5 x_6$	1,12	1
$\bar{x}_6[x_1(\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4) \vee x_2\bar{x}_5] \vee x_7\bar{x}_8$	1,12	1
$x_1\bar{x}_2(x_5 \vee \bar{x}_6) \vee x_3x_4(x_5 \vee \bar{x}_7)$	1,14	1
$\bar{x}_3(x_1 \vee x_2) \vee \bar{x}_4 \vee \bar{x}_5 \vee \bar{x}_6x_7 \vee x_2x_8$	1,12	13
$\bar{x}_8[x_1(\bar{x}_2 \vee \bar{x}_3) \vee \bar{x}_3x_4 \vee x_7(\bar{x}_5 \vee \bar{x}_6)]$	1,12	1
$x_1\bar{x}_2 \vee x_4(x_3 \vee x_5) \vee x_6\bar{x}_7 \vee x_8x_9 \vee x_{10}\bar{x}_{11} \vee x_{11}x_{13}$	1,08	7

посредственно в ДНФ. Так, например, из табл. 2-1 следует, что в устройствах управления ЦВМ «Наири» и «Минск-2» соответственно 96,5 и 73,4% всех формул бесповторны в ДНФ.

Таким образом, из результатов проведенного исследования вытекает, что НЛМ для построения управляющих логических устройств, исходя из специфики булевых формул, описывающих алгоритмы их функционирования, должны быть способны реализовать путем настройки не произвольные функции n переменных, а лишь только те из них, для которых булевы формулы бесповторны или обладают малой повторностью переменных, что обеспечит построение модулей с малыми элементной сложностью и числом внешних выводов.

Наиболее вероятной причиной низкой повторности переменных в формулах, описывающих алгоритмы работы управляющих логических устройств, является использование принципа максимального воздействия при выборе источников информации (см. § 1-2). При таком выборе каждый *ИИ, как* правило, либо только способствует, либо только препятствует срабатыванию того или иного *ИМ.* Кроме того, оказывается возможным проследить влияние

Таблица 2-3

Характеристики схем, реализующих формулу $y = (x_1x_2x_3 \vee x_4x_5)(x_6 \vee x_7)$ в различных базисах

Базис	Характеристики схем		
	Номенклатура элементов	Число элементов	Число внешних выводов
Двухвходные элементы «И» и «ИЛИ»	2	6	18
Трехвходовые элементы «И» и «ИЛИ»	2	5	20
Двухвходовые элементы «И-НЕ»	1	8	24
Трехвходовые элементы «И-НЕ»	1	6	24
Настраиваемые модули, универсальные в классе функций от трех переменных	1	3	18
Универсальный набор трехвходовых модулей	4(3)	3	12
Настраиваемые модули, универсальные в классе формул из трех букв	1	3	15

ИИ или их группы на состояние *ИМ,* т. е. произвести простую разделительную декомпозицию. При этом известно, что функциям, допускающим простую разделительную декомпозицию по всем переменным, соответствуют бесповторные формулы.

Рассмотрев статистические характеристики булевых формул, определяющие выбор функционального базиса НЛМ, вернемся к вопросу о влиянии выбранного функционального базиса на такие характеристики схем из функциональных элементов (ФЭ), как номенклатура, количество элементов и суммарное число внешних выводов.

Указанный вопрос рассмотрим на примере формулы, бесповторной в базисе $\{\&, \vee, \bar{}\}$, что, как было показано выше, характерно для рассматриваемого класса устройств.

Пусть требуется реализовать некоторую бесповторную формулу в указанном базисе, например $y = (x_1x_2x_3 \vee x_4x_5)(x_6 \vee x_7)$, на двухвходовых элементах «И» и «ИЛИ». На рис. 2-1, *а* приведена схема, построенная из этих элементов, которая характеризуется параметрами, приведенными в табл. 2-3.

Попытаемся уменьшить значения этих характеристик путем некоторого повышения уровня интеграции используемых элементов - применения трехвходовых элементов «И» и «ИЛИ» (рис. 2-1,б). Из табл. 2-3 следует, что при переходе к таким элементам номенклатура не изменилась, число элементов уменьшилось незначительно, а число внешних выводов возросло. Следовательно, повышение уровня интеграции элементов путем простого увеличения числа входов элементов «И» и «ИЛИ» не позволяет улучшить все указанные характеристики схемы.

Переход к элементам типа «И - НЕ» позволяет уменьшить номенклатуру составляющих схемы, однако их число и суммарное число внешних выводов при этом не уменьшаются (рис. 2-1, в, г).

Реализуем заданную формулу на существенно других элементах - настраиваемых логических модулях среднего уровня интеграции, универсальных в классе всех булевых функций от трех переменных (рис. 2-1,д). Использование таких модулей позволяет резко сократить число дискретных компонентов в схеме без увеличения их номенклатуры. Однако ввиду того, что каждый такой модуль имеет (при использовании в числе операций настройки операции «Антиотождествление») шесть внешних выводов (пять входов и один выход), то суммарное число внешних выводов компонентов схемы не уменьшается (табл. 2-3).

Таким образом, использование настраиваемых модулей, универсальных в классе всех булевых функций от трех переменных, также не позволяет улучшить все рассматриваемые характеристики схемы.

Суммарное число внешних выводов компонентов схемы может быть резко снижено при одновременном уменьшении числа этих компонентов, если в качестве базиса схемы использовать универсальный набор трехвходовых элементов, каждый из которых реализует одного из представителей типов формул в булевом базисе:

$$\varphi_1 = z_1z_2z_3; \quad \varphi_2 = z_1 \vee z_2 \vee z_3; \quad \varphi_3 = (z_1 \vee z_2)z_3; \quad \varphi_4 = z_1z_2 \vee z_3.$$

В этом случае (рис. 2-1, е) номенклатура универсального набора компонентов увеличивается до четырех (в данной схеме до трех), однако значения двух остальных характеристик резко уменьшаются (табл. 2-3). Поэтому, если удастся построить логический модуль, реализующий путем настройки каждого из указанных представителей типов формул из трех букв с пятью внешними выводами (четырьмя входами и одним выходом), то заданная формула будет реализована схемой (рис. 2-1,ж), обладающей наилучшими характеристиками из всех рассмотренных схем. Модуль, обладающий указанным числом внешних выводов, разработан на основе метода, изложенного в гл. 3, и выпускается в настоящее время в составе серии поликорпусных микросборок.

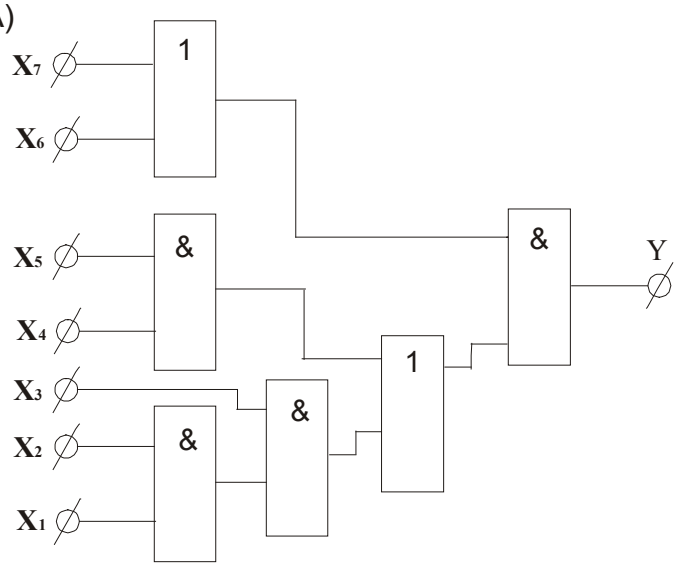
Приведенный пример характерен для бесповторных формул и формул, обладающих малой повторностью переменных в булевом базисе. Для функций, реализуемых формулами с большой повторностью переменных, более эффективными могут оказаться модули, универсальные в классе всех логических функций [9]. Так, например, формула $y = (\bar{x}_1x_2 \vee x_1\bar{x}_2)x_3 \vee (\bar{x}_1\bar{x}_2 \vee x_1x_2)\bar{x}_3 = x_1 \oplus x_2 \oplus x_3$ требует для своей реализации всего лишь одного модуля, универсального в классе функций от трех переменных, с семью либо шестью внешними выводами, в то время как модуль, универсальный в классе булевых формул из 10 букв, или схема из модулей, универсальных в классе формул из меньшего числа букв, имеют значительно большее число внешних выводов.

Для реализации формул с большой повторностью переменных в булевом базисе, но обладающих малой повторностью при их представлении в расширенном базисе $\{\&, \vee, \bar{}, \oplus\}$, можно использовать

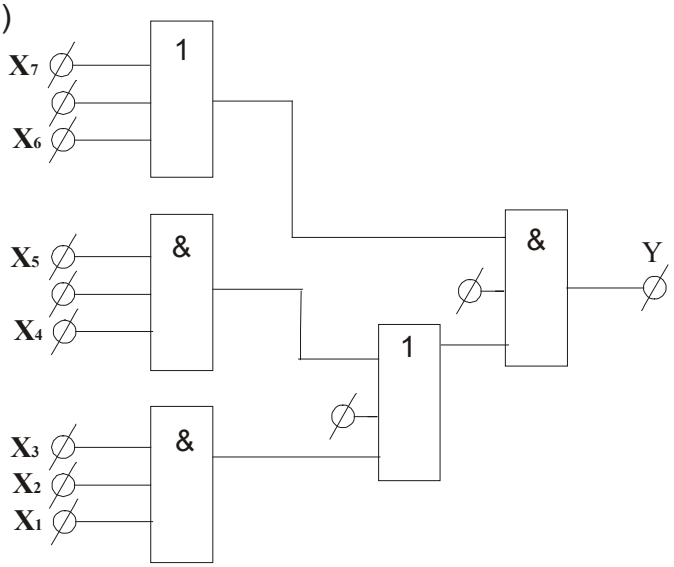
настраиваемые модули, универсальные в классе формул в базисе $\{\&, \vee, \bar{}, \oplus\}$, так как схемы, построенные из таких модулей, будут иметь меньшее число внешних выводов, чем схемы, выполненные на модулях, универсальных в классе всех булевых функций.

Таким образом, выбор в качестве функционального базиса НЛМ класса булевых формул либо класса формул $\{\&, \vee, \bar{}, \oplus\}$

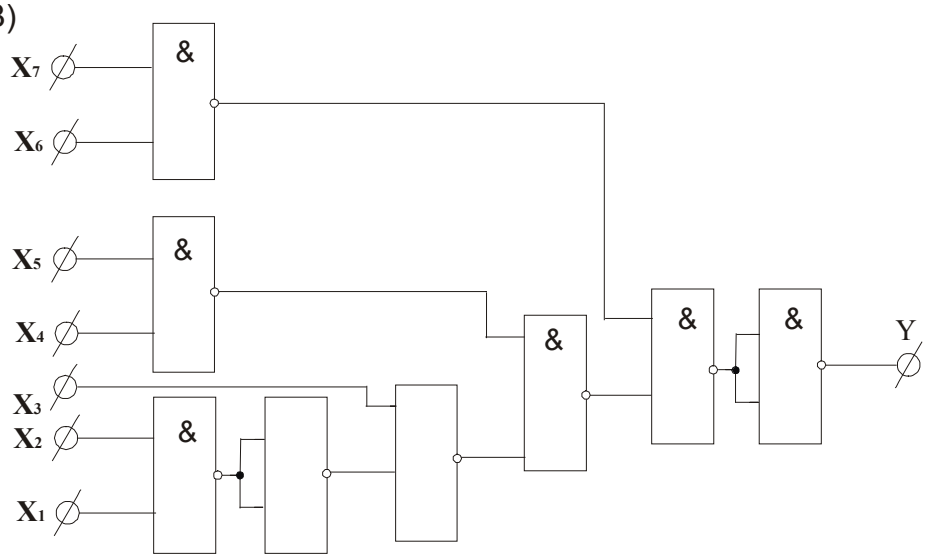
А)



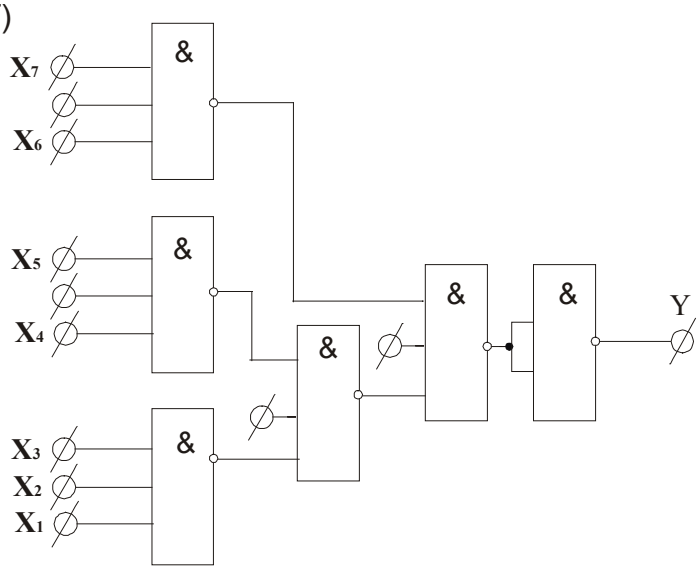
Б)



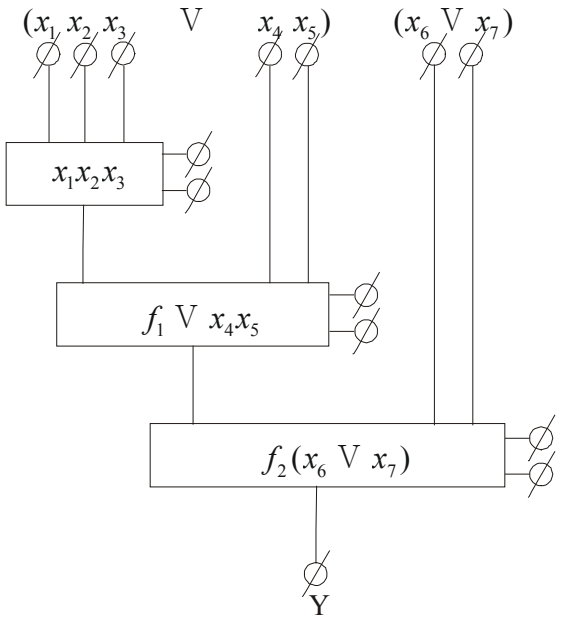
В)



Г)



Д)



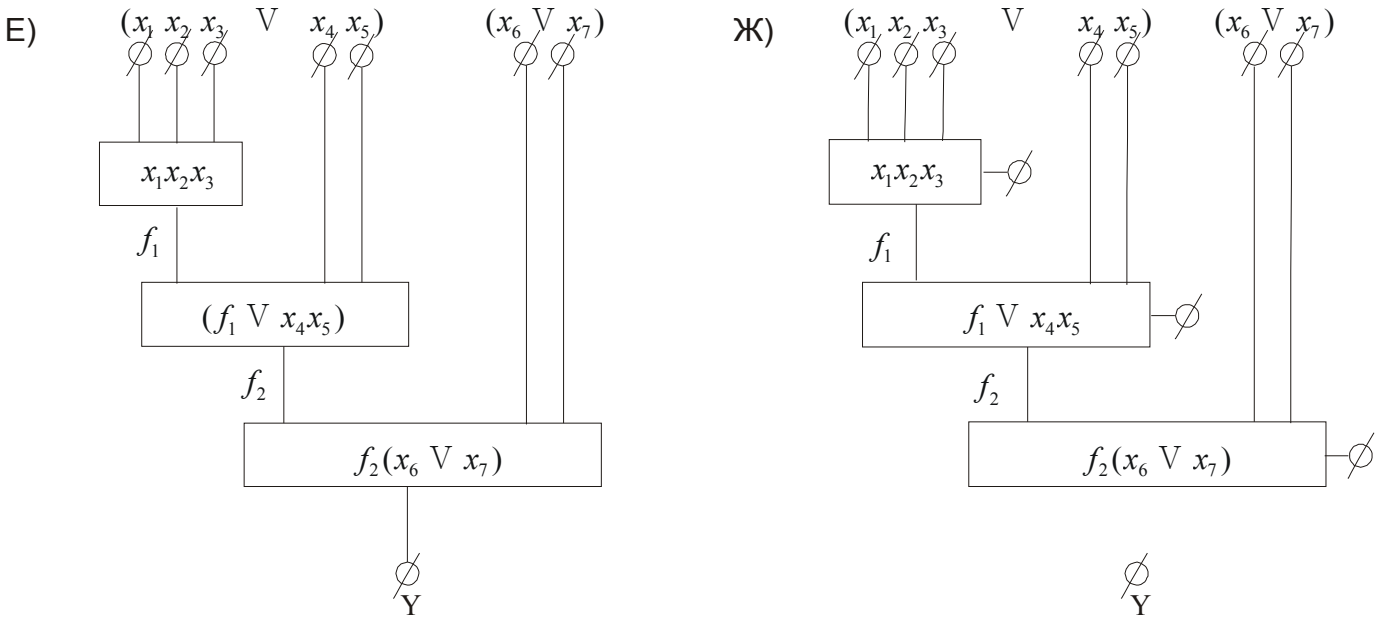


Рис. 2-1. Пример реализации бесповторной булевой формулы на раз личных элементах: а - на двухвходовых элементах «И» и «ИЛИ»;б - на трехвходовых элементах «И» и ИЛИ»; в - на двухвходовых элементах «И - НЕ»; *г* - на трехвходовых элементах «И - НЕ»; *д* - на настраиваемых модулях, универсальных в классе функций от трех переменных; *е* - на универсальном наборе трехвходовых модулей; *ж* - на настраиваемых модулях, универсальных в классе формул из трех букв

обеспечивает сокращение аппаратурных затрат при реализации формул, имеющих малую повторность переменных, по сравнению с затратами при использовании классических базисов, известных из литературы.

Необходимо отметить также, что для всех двухместных операций выбранного функционального базиса выполняется сочетательный закон. Это обеспечивает, как будет показано в гл. 4, возможность разработки простого метода рационального использования таких модулей и получения оценок сложности предстоящих схемных реализаций.

Модуль, реализующий путем настройки нормальные формулы длиной K_1 букв в базисе, для двухместных операций которого выполняется сочетательный закон, будем называть K_1 - многофункциональным. Если многофункциональный модуль реализует все нормальные формулы указанного базиса из K букв, то он является K - универсальным.

Универсальность модуля в классе бесповторных формул обеспечивает также его универсальность и в классе произвольных нормальных формул из того же числа букв. Базис, разумеется, предполагается одинаковым.

Например, модуль, настроенный на бесповторную формулу $x_1x_2 \vee x_3x_4$, может реализовать также и повторную формулу $z_1\bar{z}_2 \vee \bar{z}_1z_2$. Для этого надо на входы x_1 и x_4 подать соответственно переменные z_1 и z_2, а на входы x_2 и x_3 - переменные $\bar{z}_2$ и $\bar{z}_1$.

Реализация повторных формул из K букв в базисе K -универсальных модулей осуществляется в два этапа: а) на первом этапе модуль настраивается на реализацию бесповторной формулы, структура которой совпадает со структурой заданной формулы; б) на втором этапе у модуля, настроенного указанным способом, осуществляется отождествление входов.

Таким образом, НЛМ, первоначально выбранные на основе исследования специфики схем, описываемых бесповторными формулами, пригодны также и для реализации любых других формул, хотя при этом эффективность их использования снижается, так как для подачи одной переменной приходится использовать несколько выводов, как в приведенном выше примере.

Возникает вопрос: если НЛМ должны выполнять бесповторные формулы от K, букв, то сколько этих формул существует и каковы они? Иными словами, в первую очередь необходимо рассмотреть вопросы подсчета, классификации и табулирования бесповторных формул в выбранных базисах $\{\&, \vee, \bar{}\}$ и $\{\&, \vee, \bar{}, \oplus\}$, чему и посвящен следующий параграф.

2-3. СВОЙСТВА БЕСПОВТОРНЫХ БУЛЕВЫХ ФОРМУЛ

Число PN -типов бесповторных формул в базисе $\{\&, \vee, \bar{}\}$.

Важнейшей работой, посвященной этому вопросу, является статья Дж. Риордана и К. Шеннона «Число двухполюсных параллельно-последовательных схем» [57]. В ней рассматривались

Таблица 2-4

Число представителей бесповторных двухполюсных П – схем

Число схем	Число контактов d									
	1	2	3	4	5	6	7	8	9	10
r(d)	1	2	4	9	22	57	154	429	1225	3565
$S_{PN}(d)$	1	2	4	10	24	66	180	522	1532	4624
t(d)	1	2	4	10	28	84	264	858	2860	9724

Неизоморфные двухполюсные бесповторные параллельно-последовательные схемы и определялось их число.

В табл. 2-4 приведены значения числа представителей рассматриваемых схем $S_{PN}(d)$ при $d = 1 \div 10$, где d – число контактов в схеме.

В указанной работе найдены также рекуррентные соотношения для определения нижних и верхних оценок числа этих схем:

$$r(d) = \frac{3}{2} r(d-1) + \frac{1}{4} \sum_{i=1}^{d-1} r(d-i) r(i), \qquad (2\text{-}9)$$

$$t(d) = t(d-1) + \frac{1}{2} \sum_{i=1}^{d-1} t(d-i) t(i), \qquad (2\text{-}10)$$

где

$$r(1) = t(1) = 1, \;\; r(2) = t(2) = 2 .$$

При этом

$$t(d) =\leq S_{PN}(d) \leq t(d). \qquad (2\text{-}11)$$

Асимптотическая оценка числа схем $S_{PN}(d)$ имеет вид

$$S_{PN}(d) \approx \frac{3}{7} (3,56)^d * d^{-3/2}. \qquad (2\text{-}12)$$

Наличие изоморфизма между П-схемами и булевыми формулами приводит к тому, что все результаты, полученные в работе [57], могут быть перенесены на класс бесповторных в булевом базисе формул. Однако в силу того, что в рассмотренной работе авторы не различают замыкающие и размыкающие контакты, а также схемы, эквивалентные относительно перестановок их последовательных и параллельных частей, можно утверждать, что эти результаты применимы лишь для определения числа *PN* - типов бесповторных формул.

Булева формула из h букв $f(x_1,...,x_h)$ принадлежит одному типу с формулой из h букв $g(x_1,...,x_h)$, если $f(x_1,...,x_h)$ переходит в $g(x_1,...,x_h)$ при некоторой перестановке переменных и замене некоторых переменных их отрицаниями.

Классификация булевых формул по типам относительно указанных операций называется PN - классификацией (от английских слов permutation - перестановка и negation - отрицание). Практически однотипность по *PN* - классификации соответствует возможности подачи на входы схемы произвольного сочетания входных переменных и их отрицаний.

Примеры однотипных формул:

$$x_1(x_2 \vee \bar{x}_3) \;\text{ и }\; x_4(\bar{x}_2 \vee \bar{x}_1);$$

$$x_1x_2 \vee x_3x_4 \;\text{ и }\; \bar{x}_1x_3 \vee \bar{x}_1x_3 \vee \bar{x}_2\bar{x}_4.$$

В результате классификации множество булевых формул распадается на попарно-непересекающиеся классы - множества однотипных формул. Каждую формулу данного класса можно выбрать в качестве представителя этого класса. Булевы формулы, принадлежащие одному классу, реализуются физически одинаковыми схемами. Поэтому для каждого класса достаточно реализовать

лишь одну схему, структура которой описывается формулой представителя класса. Получение любой формулы, принадлежащей классу, при этом осуществляется путем перестановки переменных на входе схемы-представителя за счет изменения подключения входов и подачи на входы прямых и инверсных значений входных переменных. При неравной доступности выходов *ИИ* инверсные значения входных переменных получают от инверторов, располагаемых обычно независимо от модулей, например, на отдельной плате. В случае, если инверсные значения входных переменных недоступны, а использование отдельных инверторов нежелательно, то переходят к *P* - классификации, которая будет рассмотрена ниже.

Использование введенной классификации позволяет объединять между собой лишь представителей типов заданных функций, а не сами функции, что резко упрощает процесс построения модулей. Знание числа представителей типов функций недостаточно для построения НЛМ. Для этой цели необходимо перечисление всех представителей, которые должны быть объединены в порождающую функцию модуля. Поэтому рассмотрим вопрос о табулировании представителей указанного класса формул.

Табулирование представителей *PN*-типов бесповторных формул в базисе $\{\&, \vee, \bar{}\}$**.** Рассмотрим процедуру образования бесповторных формул, несколько отличную от изложенной в работе [57]. Этот подход более приспособлен для целей табулирования. В его основе лежит определение количества разбиений

Таблица 2-5

Число разбиений числа *h* на слагаемые

h	1	2	3	4	5	6	7	8
R(h)	1	2	3	5	7	11	15	22

R(h) числа *h* на сумму целых ненулевых слагаемых. В работе [49] приведено рекуррентное соотношение для определения числа разбиений, которое имеет следующий вид:

$$R(h) = \sum_{k=1} (-1)^{k-1} \left[R(h - \frac{3k^2 - k}{2}) + R(h - \frac{3k^2 + k}{2}) \right]. \qquad (2\text{-}13)$$

Все аргументы в правой части этого соотношения должны быть неотрицательными, а граничное условие должно иметь значение, равное единице: R(0) = 1. В табл. 2-5 приведены значения *R(h)* при $h = 1 \div 8$.

Соотношение (2-13) может быть использовано также для подсчета числа представителей *PN*-типов бесповторных ДНФ из *h* букв, так как существует взаимно-однозначное соответствие между представителями *PN*-типов бесповторных ДНФ и разбиениями.

Например, при *h* = 3 существуют три *PN*-типа бесповторных ДНФ, представители которых имеют следующий вид: $f_1 = x_1 x_2 x_3$, $f_2 = x_1 x_2 \vee x_3$, $f_3 = x_1 \vee x_2 \vee x_3$. Им соответствуют следующие разбиения: 3; $2+1$; 1+1+1.

Воспользуемся *0(h)* - *R(h)* - 1 разбиениями (всеми разбиениями за исключением собственно числа *h)* для образования представителей *PN* - типов бесповторных в базисе $\{\&, \vee, \bar{}\}$ формул, в которых последней операцией является дизъюнкция (формулы первого вида). Так как среди бесповторных формул отсутствуют самодвойственные [43], то каждой бесповторной формуле, в которой последней операцией является дизъюнкция, соответствует двойственная ей формула, в которой последней операцией является конъюнкция (формулы второго вида).

Ввиду того что в бесповторных формулах все буквы различны или имеют различные индексы, с целью более компактной их записи введем представление *PN* - типов этих формул в виде арифметических полиномов.

Таблица 2-6

Представители *PN*-типов бесповторных формул в базисе $\{\&, \vee, \bar{}\}$

h	$O(h)$	№ п/п	Разбиения	Формулы первого вида	$S_{\vee}(h)$	Формулы второго вида	$S_{\&}(h)$	$S_{PN}(h)$
2	1	1	1+1	1+1	1	2	1	2
3	2	1	2+1	2+1	1	(1+1)1	1	4
		2	1+1+1	1+1+1	1	3		
4	4	1	3+1	(1+1)1+1 3+1	2	(2+1)1 (1+1+1)1	2	
		2	2+2	2+2	1	(1+1)(1+1)	1	10
		3	2+1+1	2+1+1	1	(1+1)2	1	
		4	1+1+1+1	1+1+1+1	1	4	1	
5	6	1	4+1	(2+1)1+1 (1+1+1)1+1 (1+1)(1+1)+1 (1+1)2+1 4+1	5	[(1+1)1+1]1 (3+1)1 (2+2)1 (2+1+1)1 (1+1+1+1)1	5	
		2	3+1	(1+1)1+2 3+2	2	(2+1)(1+1) (1+1+1)(1+1)	2	24
		3	3+1+1	(1+1)1+1+1 3+1+1	2	(2+1)2 (1+1+1)2	2	
		4	2+2+1	2+2+1	1	(1+1)(1+1)1	1	
		5	2+1+1+1	2+1+1+1	1	(1+1)3	1	
		6	1+1+1+1+1	1+1+1+1+1	1	5	1	

Таблица 2-7

Число *PN*-типов бесповторных формул в базисе $\{\&, \vee, \bar{}\}$

h	$O(h)$	№ п/п	Разбиения	$S_{\vee}(h)$	$S_{\&}(h)$	$S_{PN}(h)$
6	10	1	5+1	12	12	
		2	4+2	5	5	
		3	4+1+1	5	5	
		4	3+3	3	3	
		5	3+2+1	2	2	66
		6	3+1+1+1	2	2	
		7	2+2+2	1	1	
		8	2+2+1+1	1	1	
		9	2+1+1+1+1	1	1	
		10	1+1+1+1+1+1	1	1	

7	14	1	6+1	33	33	
		2	5+2	12	12	
		3	5+1+1	12	12	
		4	4+3	10	10	
		5	4+2+1	5	5	
		6	4+1+1+1	5	5	
		7	3+3+1	3	3	180
		8	3+2+2	2	2	
		9	3+2+1+1	2	2	
		10	3+1+1+1+1	2	2	
		11	2+2+2+1	1	1	
		12	2+2+1+1+1	1	1	
		13	2+1+1+1+1+1	1	1	
		14	1+1+1+1+1+1+1	1	1	
8	21	1	7+1	90	90	
		2	6+2	33	33	
		3	6+1+1	33	33	
		4	5+3	24	24	
		5	5+2+1	12	12	
		6	5+1+1+1	12	12	
		7	4+4	15	15	
		8	4+3+1	10	10	
		9	4+2+2	5	5	
		10	4+2+1+1	5	5	
		11	4+1+1+1+1	5	5	522
		12	3+3+2	3	3	
		13	3+3+1+1	3	3	
		14	3+2+2+1	2	2	
		15	3+2+1+1+1	2	2	
		16	3+1+1+1+1+1	2	2	
		17	2+2+2+2	1	1	
		18	2+2+2+1+1	1	1	
		19	2+2+1+1+1+1	1	1	
		20	2+1+1+1+1+1+1	1	1	
		21	1+1+1+1+1+1+1+1	1	1	

Арифметическим полиномом будем называть символическую запись булевой формулы посредством арифметического выражения, в котором числа изображают ранг каждой конъюнкции, а знаки суммы и произведения соответствуют операциям дизъюнкции и конъюнкции в формуле.

Обезличенное представление структуры формулы в виде арифметического полинома во многом аналогично представлению структуры контактной схемы ее «скелетом» [13]. Представление формул в виде арифметических полиномов имеет смысл не только при классификации бесповторных формул, но и при допусти мости переобозначения входных переменных, и для классификации произвольных формул. При этом арифметические полиномы могут использоваться как представители типов таких формул. Поэтому, подсчитывая число представителей *PN*-типов бесповторных нормальных формул, мы тем самым определяем и число представителей типов произвольных нормальных формул, заданных в том же базисе.

Введение представления формул в виде арифметических полиномов позволяет использовать разбиения для табулирования представителей рассматриваемого класса формул. При этом построение формул первого вида осуществляется путем подстановки в разбиения вместо каждой из цифр всех формул второго вида из меньшего числа букв. В табл. 2-6 приведен процесс образования представителей *PN*-типов бесповторных формул в базисе $\{\&, \vee, \bar{}\}$ при $h = 2 \div 5$.

Формулы, содержащие большее число букв, строятся по тому же правилу, однако при этих значениях Л необходимо учитывать то, что некоторые разбиения порождают изоморфные формулы, из которых в качестве представителя типа должна выбираться одна из них.

Таблица 2-8

Число двухполюсных П-схем с различными ярлыками

Число схем	Число букв h									
	1	2	3	4	5	6	7	8	9	10
$S_N(h)$	1	2	8	52	472	5504	78416	1320064	25637824	564275648

Появление изоморфных формул связано с наличием в некоторых разбиениях одинаковых цифр (при $h \geq 6$). При $h = 6$ существует лишь одно разбиение, порождающее изоморфные формулы. Действительно, подставляя в разбиение 3 + 3 формулы второго вида из трех букв, получим четыре формулы: (1+1)1+3; 3+(1+1)1; (1+1)1+(1+1)1; 3+3, из которых только три являются различными, а первые две совпадают. При $h = 7$ также существует лишь одно такое разбиение 3+3+1, а при $h = 8$ - уже три разбиения, порождающих изоморфные формулы: 4+4; 3 +3+2; 3 + 3+1.

В табл. 2-7 приведены разбиения для $h = 6 \div 8$ и указано число типов формул, получаемых из каждого разбиения.

На основе подхода, изложенного выше, авторами были протабулированы все представители *PN*-типов бесповторных формул в базисе $\{\&, \vee, \bar{}\}$ из $h = 1 \div 8$ букв.

При этом все представители типов формул из *h* букв были разбиты на группы, каждая из которых может быть реализована одной конфигурацией древовидной схемы, а ее элементы могут выполнять одну из операций базиса $\{\&, \vee, \bar{}\}$.

Число N-типов бесповторных формул в базисе $\{\&, \vee, \bar{}\}$**.** В книге Дж. Риордана [49] приводятся производящие функции для определения числа параллельно-последовательных схем, снабженных различными ярлыками (различными индексами в обозначениях контактов). При этом схемы, отличающиеся лишь наличием замыкающих контактов вместо размыкающих и наоборот, считаются различными. Численные значения различных схем для этого случая приведены в табл. 2-8.

Этим схемам соответствует подкласс формул, бесповторных в базисе $\{\&, \vee, \bar{}\}$, в котором представители типов могут отличаться не только структурой, но и перестановкой переменных. Этот подкласс формул формируется при использовании N-классификации. При этом формулы, отличающиеся только расстановкой инверсий, считаются неразличимыми и принадлежат одному N-классу.

Число формул этого класса, порождаемых одним PN-типом бесповторных формул из h букв,

$$S_N(h) = \frac{h!}{\prod(m_i!)}, \qquad (2\text{-}14)$$

где m_i - число букв в конъюнкции или число одинаковых членов в дизъюнкции.

Пример. Представитель *PN* – типа $f \equiv (1+1)1+1$ порождает

$$S_N(h) = \frac{4!}{2!} = 12$$

различных *N* – типов бесповторных формул:

$f_1 = (x_1 \vee x_2)x_3 \vee x_4;$ $f_5 = (x_1 \vee x_4)x_2 \vee x_3;$ $f_9 = (x_2 \vee x_4)x_1 \vee x_3;$

$f_2 = (x_1 \vee x_2)x_4 \vee x_3;$ $f_6 = (x_1 \vee x_4)x_3 \vee x_2;$ $f_{10} = (x_2 \vee x_4)x_3 \vee x_1;$

$f_3 = (x_1 \vee x_3)x_2 \vee x_4;$ $f_7 = (x_2 \vee x_3)x_1 \vee x_4;$ $f_{11} = (x_3 \vee x_4)x_1 \vee x_2;$

$f_4 = (x_1 \vee x_3)x_4 \vee x_2;$ $f_8 = (x_2 \vee x_3)x_4 \vee x_1;$ $f_{12} = (x_3 \vee x_4)x_2 \vee x_1;$

Число бесповторных формул в базисе $\{\&,\vee,\bar{}\}$**.** Значение числа N – типов бесповторных формул $S_N(h)$ позволяет определить

Таблица 2-9

Число бесповторных формул в базисе $\{\&,\vee,\bar{}\}$

Число формул	Число букв h									
	1	2	3	4	5	6	7	8	9	10
$S(h)$	2	8	64	832	15104	352256	10037248	$338\cdot10^6$	$13\cdot10^9$	$578\cdot10^9$

Число бесповторных формул $S(h)$ в базисе $\{\&,\vee,\bar{}\}$. Несмотря на то, что $S(h)$ вычисляется по $S_N(h)$ достаточно просто:

$$S(h) = S_N(h)2^h, \qquad (2\text{-}15)$$

в литературе до настоящего времени число бесповторных формул в базисе $\{\&,\vee,\bar{}\}$ не приводилось.

В табл. 2-9 указаны значения $S(h)$ при $h = 1 \div 10$.

Пример. Перечислим все бесповторные в рассматриваемом базисе формулы из двух букв:

$$f_1 = x_1x_2; \quad f_3 = x_1\overline{x_2}; \quad f_5 = x_1 \vee x_2; \quad f_7 = x_1 \vee \overline{x_2};$$
$$f_2 = \overline{x_1}x_2; \quad f_4 = \overline{x_1x_2}; \quad f_6 = \overline{x_1} \vee x_2; \quad f_8 = \overline{x_1} \vee \overline{x_2}.$$

Для характеристики класса бесповторных формул авторами введен коэффициент

$$B(h) = \frac{\log_2 S(h)}{\log_2 S_{PN}(h)}. \qquad (2\text{-}16)$$

Значения этого коэффициента при $h = 2 \div 9$ приведены в табл. 2-10.

Таблица 2-10

h	2	3	4	5	6	7	8	9
$B(h)$	3	3	2,92	2,98	3,05	3	3,02	3,09

Таким образом, можно утверждать, что при рассмотренных малых значениях числа букв $B(h) \approx 3$. Эта величина может использоваться в качестве инварианта класса бесповторных формул в базисе $\{\&,\vee,\bar{}\}$.

Число *P* - типов бесповторных формул в базисе $\{\&,\vee,\bar{}\}$**.**Представители *P* - типов бесповторных формул в базисе $\{\&,\vee,\bar{}\}$ образуют класс формул, в котором формулы, отличающиеся лишь перестановкой переменных, неразличимы. При *P* - классификации различают лишь формулы, отличающиеся структурой и расстановкой инверсий над одиночными символами переменных.

Путем непосредственного подсчета были найдены для каждого *PN* -типа бесповторных формул порождаемые им представители *P* - типов формул, что позволило определить их суммарное число при $h = 1 \div 6$ (табл. 2-11).

Таблица 2-11

Число *P* - типов бесповторных формул в базисе $\{\&,\vee,\bar{}\}$

h	1	2	3	4	5	6
$S_P(h)$	2	6	20	80	340	1582

Пример. Существует шесть P - типов бесповторных формул в базисе $\{\&,\vee,\bar{}\}$ из двух букв, представителями которых являются: $f_1 = x_1x_2;\ f_2 = \overline{x_1}x_2;\ f_3 = \overline{x_1x_2};\ f_4 = x_1 \vee x_2;\ f_6 = \overline{x_1} \vee x_2;\ f_6 = \overline{x_1} \vee \overline{x_2}.$

Перечислим также представителей *P* – типов формул из трех букв:

$f_5 = (\overline{x_1} \vee \overline{x_2})\overline{x_3};$ $f_{11} = \overline{x_1 x_2} \vee \overline{x_3};$

$f_1 = \overline{x_1 x_2 x_3};$ $f_6 = (\overline{x_1} \vee \overline{x_2})x_3;$ $f_{12} = \overline{x_1 x_2} \vee x_3;$ $f_{17} = \overline{x_1} \vee \overline{x_2} \vee \overline{x_3};$

$f_2 = \overline{x_1 x_2} x_3;$ $f_7 = (\overline{x_1} \vee \overline{x_2})\overline{x_3};$ $f_{13} = \overline{x_1} x_2 \vee \overline{x_3};$ $f_{18} = \overline{x_1} \vee \overline{x_2} \vee x_3;$

$f_3 = \overline{x_1} x_2 x_3;$ $f_8 = (\overline{x_1} \vee x_2)x_3;$ $f_{14} = \overline{x_1} x_2 \vee x_3;$ $f_{19} = \overline{x_1} \vee x_2 \vee x_3;$

$f_4 = x_1 x_2 x_3;$ $f_9 = (x_1 \vee x_2)\overline{x_3};$ $f_{15} = x_1 x_2 \vee \overline{x_3};$ $f_{20} = x_1 \vee x_2 \vee x_3;$

$f_{10} = (x_1 \vee x_2)x_3;$ $f_{16} = x_1 x_2 \vee x_3;$

Число *NPN* – типов бесповторных формул в базисе $\{\&, \vee, \bar{}\}$**.**

Рассмотрев *PN* – классификацию, а также классификации с большим числом представителей, остановимся на классификации, позволяющей сократить число этих представителей.

При применении *NPN* – классификации используются три операции: замена формулы её отрицанием *N*, перестановка переменных *P* и замена переменных их отрицаниями *N* [7]. Введение дополнительной операции позволяет сократить число представителей типов формул по сравнению с их числом в *PN* – классификации в два раза (табл. 2-12).

Некоторые свойства бесповторных формул. Задача построения настраиваемых логических модулей, выполняющих любую

Таблица 2-12

Число ***NPN* – типов бесповторных формул в базисе** $\{\&, \vee, \bar{}\}$

h	1	2	3	4	5	6	7	8
$S_{NPN}(h)$	1	1	2	5	12	33	90	261

формулу длиной ***K*** букв, требует для своего решения знания некоторых свойств бесповторных формул и соответствующих им функций. Для этого изложим несколько вопросов, которые могут показаться абстрактными, но на самом деле пригодятся в дальнейшем. Они относятся к числу единиц и структуре их расположения в таблицах истинности, соответствующих бесповторным формулам.

Число единиц в столбце значений булевой функции условимся называть рангом. Если имеются формулы, зависящие каждая от своего множества переменных:
$f_1(x_1,...,x_{h_1}), f_2(x_{h_1+1},...,x_{h_1+h_2}),..., f_k(x_{h_1+...+h_{t-1}},...,x_{h_1+...+h_{k-1}+h_k})$ - и имеющие ранги соответственно $r_1, r_2 ... r_t,$ то ранг их произведения $F = f_1 f_2 ... f_t$ равен произведению рангов $r = r_1 r_2 ... r_t$.

В качестве примера рассмотрим произведение двух функций f_1 и f_2.

Ясно, что на тех наборах переменных, где $f_1 = 0$, и произведение $f_1 f_2 = 0$, а также на тех наборах переменных, где $f_1 = 1$, функция $F = f_2$.

Таким образом, если в таблице истинности сначала разместить переменные от f_1, а потом – от f_2, то столбец значений $F = f_1 f_2$ будет состоять из 2^{h_1} фрагментов длиной 2^{h_2} двух разновидностей: состоящих сплошь из 0 на наборах $x_1,...,x_{h_1}$, где $f_1 = 0$, и повторяющих f_2 на наборах $x_1,...,x_{h_1}$, где $f_2 = 1.$

Пример. Определить ранг функции $F_1 = f_1 f_2$, если $f_1 = \overline{x_1 \oplus x_2}$, а $f_2 = \ddot{x} \vee x_4.$

Составим таблицы истинности для функций f_1 и f_2 (табл, 2-13, 2-14).

Ввиду того, что $r_1 = 2$, а $r_2 = 3$, на основании высказанного утверждения $r = r_1 r_2 = 6.$ Проверим полученный результат (табл. 2-15).

Таблица 2-13
Таблица истинности для f_1

x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

Таблица 2-14
Таблица истинности для f_2

x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

Таблица 2-15

Таблица истинности для F_1 и F_2

x_1	x_2	x_3	x_4	f_1	f_2	$F_1 = f_1 f_2$	$F_2 = f_1 \vee f_2$
0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	1
0	0	1	0	1	0	0	1
0	0	1	1	1	1	1	1
0	1	0	0	0	0	0	1
0	1	0	1	0	1	0	1
0	1	1	0	0	0	0	0
0	1	1	1	0	1	0	1
1	0	0	0	0	1	0	1
1	0	0	1	0	1	0	1
1	0	1	0	0	0	0	0
1	0	1	1	0	1	0	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	1	1	1

Ранг суммы двух формул вычисляется несколько сложнее, в соответствии с выражением

$$r = r_2 \cdot 2^{h_1} + r_1(2^{h_2} - r_2). \qquad (2\text{-}17)$$

Для формул, рассмотренных в предыдущем примере, ранг суммы $F_2 = f_1 \mathrm{V}\, f_2$ составит

$$r = 3 \cdot 2^2 + 2(2^2 - 3) = 14.$$

Это легко проверить, выполнив сложение функций f_1 и f_2 (табл. 2-15).

Отметим интересное следствие. Пусть ранг формулы $f_1(x_{n-1},...,x_1)$, бесповторной в базисе $\{\&, \vee, \bar{}\}$, равен r_1; тогда формула $f_2(x_h, x_{h-1},...,x_1) = x_h \vee f_1(x_{h-1},...,x_1)$ имеет ранг $r_2 = 2^{h-1} + r_1$.

Полученные соотношения позволяют вычислять ранг любой бесповторной формулы в указанном базисе аналитически, без построения истинности.

Пример. Определить ранг формулы $F = \{[(x_1 \vee x_2)x_3 \vee x_4]x_5 \vee x_6\}x_7$.

$f_1 = x_1;$ $\quad r_1 = 1;$

$f_2 = x_1 \vee x_2;$ $\quad r_2 = 2^1 + 1 = 3;$

$f_3 = (x_1 \vee x_2)x_3;$ $\quad r_3 = 3;$

$f_4 = (x_1 \vee x_2)x_3 \vee x_4;$ $\quad r_4 = 2^3 + 3 = 11;$

$f_5 = [(x_1 \vee x_2)x_3 \vee x_4]x_5;$ $\quad r_5 = 11;$

$f_6 = [(x_1 \vee x_2)x_3 \vee x_4]x_5 \vee x_6;$ $\quad r_6 = 2^5 + 11 = 43;$

$f_7 = \{[(x_1 \vee x_2)x_3 \vee x_4]x_5 \vee x_6\}x_7;$ $\quad r_7 = 43.$

Ранг функции, несущественно зависящей от некоторых свои; переменных, есть четное число, так как если в функции $f(x_n,.....,x_1)$ имеется хотя бы одна несущественная переменная, например с порядковым номером *n*, то столбец значений этой функции может быть построен удвоением столбца значений функции $f(x_{n-1},...,x_1)$. Следовательно, вне зависимости от ранга функции $f(x_{n-1},...,x_1)$ ранг функции $f(x_n,...,x_1)$ четен.

Приведем еще ряд утверждений. Если булева функция имеет, нечетный ранг, то она существенно зависит от всех своих переменных [56]. Ранг формулы, бесповторной в базисе $\{\&,\vee,\bar{}\}$, существенно зависящей от всех своих переменных, нечетен. Инвариантами *PN*-типов бесповторных пороговых функций, существенно зависящих от *n* переменных, являются их ранги (нечетные числа): $1,3,5,...,2^n-1$.

Классификация и табулирование бесповторных формул в базисе $\{\&,\vee,\bar{},\oplus\}$. В работе [66] приведено соотношение для определения числа *PN*-типов бесповторных формул в базисе $\{\&,\vee,\bar{},\oplus\}$, реализуемых линейной цепочкой из элементов, каждый из которых после настройки выполняет одну из трех двухместных операций $\{\&,\vee,\oplus\}$ над информационными переменными.

Указанное соотношение имеет вид

$$B_1(h)=\frac{1}{\sqrt{5}}\left[\left(\frac{3+\sqrt{5}}{2}\right)^h-\left(\frac{3-\sqrt{5}}{2}\right)^h\right]\approx\frac{(2,62)^h}{2,24}. \qquad (2\text{-}18)$$

Интересным является тот факт, что значения $B_1(h)$, вычисленные по этой формуле, равны целым числам, являющимся нечетными членами ряда Фибоначчи:

$$B_1(h)=\Phi(2h+1). \qquad (2\text{-}19)$$

Числа Фибоначчи определяются из соотношения

$$\Phi(h)=\Phi(h-1)+\Phi(h-2), \qquad (2\text{-}20)$$

где Ф(1)=0, Ф(2)=1.

Ряд Фибоначчи имеет вид: 0,1,1,2,3,5,8,13,21,34,55,89,144,233,377 и т.д.

В табл. 2-16 приведены значения $B_1(h)$ при $h=1\div10$.

Отметим, что при $h\geq3$

$$B_1(h)<3^{h-1}. \qquad (2\text{-}21)$$

Это связано с явлением изоморфизма, вызываемым операцией $\oplus$. Поэтому в базисе $\{\&,\vee,\bar{},\oplus\}$ формулы, обладающие разной структурой, могут принадлежать одному *PN* – типу.

Таблица 2-16

Число *PN* – типов бесповторных формул в базисе $\{\&,\vee,\bar{},\oplus\}$

h	1	2	3	4	5	6	7	8	9	10
$B_1(h)$	1	3	8	21	55	144	377	987	2584	6765
$B_2(h)$	1	3	8	29	101	405	?	?	?	?
$B_3(h)$	1	3	8	42	165	864	4147	22701	118864	662970

Таблица 2-17

Представители *PN* – типов бесповторных формул в базисе $\{\&,\vee,\bar{},\oplus\}$

h	№ п/п	Представитель типа формул	h	№ п/п	Представитель типа формул
1	1	1	4	10	(1+1)1+1
2	1	2		11	(1⊕1)1+1
	2	1+1		12	2+1+1
	3	1⊕1		13	1+1+1+1

3	1	3		14	1⊕1+1+1
	2	(1⊕1)1		15	2⊕1+1
	3	(1⊕1)1			(1+1)⊕1+1
	4	2+1		16	1⊕1⊕1+1
	5	1+1+1		17	3⊕1
	6	1⊕1+1			(1+1+1)⊕1
	7	2⊕1		18	(1+1)1⊕1
		(1+1)⊕1			(2+1)⊕1
	8	1⊕1⊕1		19	(1⊕1)1⊕1
4	1	4			(1⊕1+1)+1
	2	(1+1)2		20	(2⊕1)⊕1
	3	(1⊕1)2			(1+1)⊕ 1⊕1
	4	(2+1)1		21	1⊕1⊕1⊕1
	5	(1+1+1)1		22	3⊕2
	6	(1⊕1+1)1		23	(1+1)(1+1)
	7	(2+1)1			2⊕2
		[(1+1)⊕1]1		24	(1+1)⊕(1+1)
	8	(1⊕1⊕1)1			2⊕(1+1)
	9	3+1		25	(1⊕1)(1⊕1)
				26	1⊕1+1⊕1
				27	2+1⊕1
				28	(1+1)(1⊕1)
				29	

Например, формулы $f_1 = x_1x_2 \oplus x_3$ и $f_2 = (x_1 \vee x_2) \oplus x_3$ принадлежат одному *PN*-типу, так как $x_1x_2 \oplus x_3 = (\overline{x_1} \vee \overline{x_2}) \oplus \overline{x_3}$. К одному типу принадлежат также формулы $f_3 = (x_1 \oplus x_2)x_3 \oplus x_4$ и $f_4 = (x_1 \oplus x_2 \vee x_3) \oplus x_4$, так как $(x_1 \oplus x_2)x_3 \oplus x_4 = (\overline{x_1} \oplus x_2 \vee \overline{x_3}) \oplus \overline{x}^4$.

При этом справедливо утверждение: если формулы F_i и $F_i^{'}$ принадлежат одному *PN*-типу, то и формулы $F_1 \oplus F_2$ и $\overline{F}_1^{'} \oplus \overline{F}_2^{'}$ также принадлежат: одному *PN*-типов.

В работе [72] для $h = 1 \div 6$ найдено число представителей *PN*-типов бесповторных формул в базисе $\{\&, \vee, \bar{}, \oplus\}$, а авторами получено соотношение для определения значений $B_3(h)$ - верхней оценки числа PW-типов бесповторных формул в этом базисе для любых *h*. Значения $B_2(h)$ и $B_3(h)$ для $h \le 10$ приведены в табл. 2-16.

В табл. 2-17 приведены результаты табулирования представителей *PN*-типов бесповторных формул в рассматриваемом базисе при $h = 1 \div 4$, что необходимо для построения модулей, универсальных в этом классе формул.

Таким образом, ознакомившись с этой главой, можно сделать вывод, что неплохо иметь набор модулей, реализующих все формулы определенной длины. Еще лучше иметь один настраиваемый модуль, реализующий все формулы определенной длины. Однако изложенный материал дает возможность оценить и объем «бедствия», которое ждет, если погнаться за полной универсальностью и очень большими значениями *K*.

Проведенное табулирование показывает, что число бесповторных формул растет, хотя и неизмеримо медленнее, чем число формул вообще, но все-таки достаточно быстро. Поэтому, вероятно, разумно ограничиться значениями $K = 6 \div 8$, а также, по возможности, остаться в рамках *PN*-классификации.

Ограничив *K,* мы приходим к задаче: как строить схемы, соответствующие формулам, число букв в которых превышает *K?* Прежде чем привести ее решение (это будет сделано в гл. 4), займемся разработкой метода построения настраиваемых логических модулей.