

Предисловие

В эру информатизации и компьютеризации, когда большое количество операций по управлению ответственными технологическими процессами производится в автоматическом режиме, на первый план выходят мероприятия по обеспечению надежности и безопасности их реализации. Усложнение структур технических средств автоматизации, уменьшение их габаритов на фоне постоянно уменьшающихся размеров структурных единиц и непрерывно растущего функционала устройств и систем управления, эволюция элементной базы и физических материалов ее изготовления, влекут за собой повышение внимания к методам и способам обеспечения высокого качества их работы. Ключевыми направлениями развития являются адаптация известных методов технической диагностики и их совершенствование с учетом специфики современных разрабатываемых и конструируемых устройств автоматики и вычислительной техники.

За годы развития технической диагностики сформировалось отдельное направление, связанное с применением методов теории информации и кодирования при организации надежных систем управления. Использование основных принципов теории информации и кодирования позволяет наделять составляющие разрабатываемой системы свойствами обнаружения и парирования неисправностей.

Важное место среди всех избыточных кодов занимают коды, ориентированные на обнаружение ошибок (а не их исправление). Во-первых, такие коды обладают меньшей избыточностью, чем коды, исправляющие ошибки. И во-вторых, во многих приложениях не требуется корректировка возникающих вследствие неисправностей ошибок. К примеру, в системах диагностирования аппаратных средств важно своевременно обнаруживать возникающие дефекты, а не исправлять их последствия, что непременно привело бы к накоплению неисправностей. Избыточность кода напрямую влияет не только на свойства обнаружения ошибок, но и на избыточность технических средств диагностирования, необходимых для решения задач идентификации неисправных компонентов систем автоматики и вычислительной техники. Поэтому важным направлением развития технической диагностики является исследование способов построения и свойств кодов, ориентированных на обнаружение ошибок.

Данная книга посвящена изложению основ теории синтеза самопроверяемых цифровых вычислительных устройств и систем на основе классических кодов с суммированием и их основных модификаций. При этом освещены свойства обнаружения ошибок только в информационных векторах кодов при безошибочности разрядов контрольных векторов. Такая постановка задачи актуальна во многих приложениях кодов с суммированием. Одним из них является синтез

систем рабочего (функционального) диагностирования, когда информационный вектор сопоставляется с выходами объекта диагностирования, а контрольный – с выходами блока контрольной логики. Сам объект диагностирования и технические средства диагностирования в данном случае реализуются отдельно. На страницах данной книги читатель найдет описания классических и модульных кодов с суммированием единичных информационных разрядов, взвешенных кодов с суммированием, а также способов их применения при построении самопроверяемых структур.

Данная книга не повторяет других учебных изданий и будет полезна при изучении теоретических дисциплин специальностей группы 05.13.00 «Информатика, вычислительная техника и управление».

Введение

На возможности использования избыточного кодирования при построении надежных систем автоматики обратили внимание в середине XX столетия, когда начала свое активное развитие теория информации и кодирования [45, 60]. Была проведена аналогия между каналами передачи данных и вычислительными каналами систем автоматики, что позволило применять методы теории информации и кодирования для решения задачи «наделения» разрабатываемых устройств свойствами обнаружения возникающих неисправностей и при необходимости их парирования. Одними из пионеров применения методов кодирования при построении систем автоматики стали наши соотечественники М. А. Гаврилов и А. Д. Закревский в конце 1950-х гг. [3]. Внесение избыточности как на аппаратном, так и на программном уровне к настоящему времени является основным подходом при разработке и конструировании надежных систем автоматики [17, 24, 43, 48].

Избыточное кодирование используется на всех уровнях организации систем автоматики: от передачи данных между центральными и периферийными объектами до выбора архитектур управляющих блоков [25, 39, 49, 62]. Для решения различных задач используют соответственно и различные способы защиты и кодирования передаваемых данных. Нашли свое применение как простейшие избыточные коды (такие, например, как коды с проверкой четности комбинации (коды паритета), коды с суммированием, полиномиальные коды и др.), так и более сложные коды, обнаруживающие и исправляющие ошибки различных видов и с различными кратностями (к примеру, коды Хэмминга, Рида – Соломона, Боуза – Чоудхури – Хоквинема (БЧХ), сверточные, каскадные и т. д.).

Широко применяются методы теории информации и кодирования в технической диагностике [3, 8–14, 39, 65]. Использованию избыточного кодирования при разработке контролепригодных схем устройств автоматики, а также при выборе способа организации диагностического обеспечения посвящена не одна научная работа. Основные вопросы применения кодирования при построении алгоритмов диагностирования, тестирования логических схем, кодирования состояний автоматов и прочие затронуты в ставшей классической на территории постсоветского пространства монографии по технической диагностике под редакцией П. П. Пархоменко [5, 7]. В частности, отмечаются возможности применения кодов паритета [1] и равновесных кодов [38] при организации систем функционального диагностирования логических схем автоматики и вычислительной техники. Принципы применения данных кодов для решения поставленных задач определяются особенностями обнаружения ими ошибок: коды

паритета применяют с учетом обнаружения ими только одиночных искажений, а равновесные коды – любых однонаправленных (монотонных) искажений.

Важный класс избыточных кодов, эффективно применяемых при решении задач технической диагностики, образуют коды с суммированием. Такие коды являются разделимыми и, как правило, ориентированы на обнаружение искажений в информационных разрядах (а не на их исправление). Они строятся путем применения различных операций суммирования значений информационных разрядов или весовых коэффициентов, приписанных разрядам или переходам между разрядами, занимающими соседние позиции в информационных векторах [36].

Классический код с суммированием был предложен в 1961 г. в работе [28] и наиболее известен широкому кругу исследователей как код Бергера (принципы его построения описаны в третьей главе настоящей книги). Данный код позволяет обнаруживать монотонные искажения в разрядах информационного вектора и задумывался автором как код, применяемый в асимметричных каналах передачи данных. Свойство обнаружения кодами Бергера любых монотонных ошибок в настоящее время повсеместно используется для решения задач построения контролепригодных логических схем, выбора способа функционального диагностирования устройства и т. п. Следует отметить, что подобным свойством обладают также и равновесные коды, предложенные в [38], а также ряд модификаций как равновесных кодов, так и кодов с суммированием, ориентированных на обнаружение монотонных ошибок с определенными кратностями [61].

Для расширения приложений кодов Бергера оказывается проблематичным то, что ими не обнаруживается любая ошибка в информационном векторе, при которой происходит одинаковое число искажений нулевых и единичных разрядов. Более того, разряды контрольных векторов используются не столь эффективно (распределение информационных векторов между всеми контрольными векторами крайне неравномерно), а контрольные схемы для кодов (кодеры и генераторы) зачастую оказываются сложными. По этим причинам во всем мире разработаны и разрабатываются способы модификации кодов Бергера, дающие разнообразные по своим свойствам коды с суммированием, применение которых может оказаться более эффективным для каждого конкретного приложения (вплоть до использования только для устройства с конкретной структурной схемой и архитектурой на конкретной элементной базе).

Данная книга охватывает вопросы использования разнообразных кодов с суммированием для обнаружения ошибок в информационных векторах. Условно такие коды можно классифицировать на две большие группы: коды с суммированием единичных информационных разрядов и взвешенные коды с суммированием. К первой группе относятся классические и модульные коды с суммированием и некоторые их модификации, предполагающие выполнение операций с равнозначными разрядами. Ко второй группе относятся коды, у которых на начальном этапе построения между информационными разрядами было установлено неравноправие. Неравноправие устанавливается путем приписывания весовых коэффициентов разрядам или переходам от разряда к раз-

ряду в информационном векторе. Использование кодов с суммированием на практике позволяет строить самопроверяемые цифровые устройства и системы автоматики с возможностью минимизации аппаратных затрат.

Глава 1

Основные методы построения систем с обнаружением неисправностей

1.1. Введение

Одной из главных задач, возникающих при разработке устройств и систем автоматики и вычислительной техники, является обеспечение надежности и безопасности их функционирования. Решение данной задачи заключается, так или иначе, в комплексе мероприятий, включающих разнообразные методы повышения отказоустойчивости как компонентов, так и систем в целом.

Элементная база систем управления разнообразна и включает в себя составляющие с различными показателями надежности: от высоконадежных компонентов с низкими значениями интенсивностей отказов до уязвимых с точки зрения надежности. При этом часто в системах критического применения используют так называемые безопасные элементы – такие элементы, которые имеют асимметричную характеристику отказов: в таких элементах вероятность отказа, связанная с появлением ложного управляющего воздействия, на несколько порядков ниже вероятности защитного отказа, выключающего объект из работы. Как правило, безопасные элементы реализуются за счет применения схмотехнических методов исключения ложных управляющих воздействий, зачастую строятся с применением некоторой избыточности в компонентах и соответственно, имеют гораздо более сложную структуру, чем повсеместно применяемые, имеющие симметричную характеристику отказов [15].

Для обеспечения отказоустойчивости систем автоматики и вычислительной техники используются два основных направления. Первое основано на разработке устройств и систем с контролепригодными структурами, обеспечивающими простые процедуры проверки правильности функционирования путем тестового или рабочего диагностирования [4, 6, 18, 26, 62, 71]. Второе направление связано с широким применением методов резервирования – внесения избыточности в аппаратные и программные средства [15, 16, 18].

Резервирование возможно на различных уровнях реализации системы или устройства. Резервировать можно отдельные компоненты или целые блоки и узлы. При этом основное и резервное устройства могут быть абсолютно идентичными, а могут быть реализованы различными способами (это касается и аппаратных и программных средств). В последнем случае говорят о диверситете технических средств.

На практике при построении управляющих систем используют готовые компоненты, влиять на которые в сторону улучшения показателей надежности не представляется возможным. Поэтому применяют специальные методы для построения высоконадежных и безопасных систем из той элементной базы, которая выбрана для их реализации. В этом случае используют резервирование аппаратных и программных средств на уровне отдельных функциональных блоков. Широко распространены типовые структурные схемы высоконадежных систем, основанные на дублировании и троировании функциональных блоков [16, 18, 24].

Реализация высоконадежных систем не обходится без использования диагностического обеспечения. Важно своевременно идентифицировать объект, в котором присутствует дефект, и восстановить его работоспособность. В управляющих системах резервирование реализуют таким образом, чтобы дефектный объект по возможности не мог влиять на процесс выполнения основных функций всей системой. Либо вычисляемые данные маскируются и «отбрасываются» при использовании схем голосования (мажоритарных схем), либо некорректно функционирующий объект отключается от работы до тех пор, пока он не восстановит работоспособность. Процедура может быть основана на самовосстановлении, например, после перезагрузки объекта, или на техническом обслуживании и ремонте с привлечением специально обученного персонала.

Важным подходом к организации надежных и отказоустойчивых систем автоматики и вычислительной техники является использование принципов обнаружения неисправностей. С этой целью на аппаратном и программном уровнях применяют методы, основанные на использовании избыточного кодирования [10–14, 20, 21, 39, 50, 53, 70]. Избыточное кодирование используется на различных уровнях организации систем: от уровня выбора архитектуры технического объекта до уровня передачи данных между узлами систем. Ряд приложений требует только обнаружения неисправностей и фиксации данных событий с локализацией дефекта, а ряд – коррекции возникающих ошибок.

Сфокусируем внимание на проблеме синтеза систем с обнаружением неисправностей на аппаратном уровне.

1.2. Системы с обнаружением неисправностей

При организации систем с обнаружением неисправностей прежде всего анализируется множество допустимых неисправностей и выбирается математическая модель неисправности, покрывающая их с некоторой полнотой [7]. Множество допустимых неисправностей определяется различными признаками устройства: физической реализацией, компонентами структуры, связями между компонентами, технологиями изготовления, качеством сборки и т. д. Например, в релейных реализациях крайне распространены неисправности типа «обрыв» и «короткое замыкание», которые описываются с помощью модели константных неисправностей (*stuck-at faults*). В микроэлектронных и микропроцессорных реализациях устройств и систем автоматики классы неисправностей гораздо «бо-

гаче» и разнообразнее. Появляются неисправности, обусловленные спецификой реализации технического объекта.

Рассмотрим методы построения систем с обнаружением неисправностей с учетом модели одиночных константных неисправностей, возникающих во внутренних структурах технических объектов. Все эти методы основаны на использовании различных способов резервирования и технического диагностирования.

В общем случае речь идет об устройствах автоматики и вычислительной техники, обладающих возможностью запоминания своих состояний и описываемых моделями конечных автоматов. Такие устройства могут иметь синхронную и асинхронную реализацию, а свойство обнаружения неисправностей может быть заложено как на этапе синтеза самого автомата, так и при разработке диагностического обеспечения для уже полученного автомата.

При наделении конечного автомата свойством обнаружения неисправностей осуществляется кодирование его состояний некоторым избыточным кодом и дальнейший синтез устройства с избыточной структурой. Избыточность является ценой за возможность обнаружения неисправностей.

Если разрабатывается диагностическое обеспечение для уже синтезированного конечного автомата, то реализуется стратегия «наблюдения за результатами вычислений». Диагностическое обеспечение синтезируется с учетом вида полученного устройства, возможности выделения в нем регистров памяти и комбинационных составляющих и т. д. Если такая декомпозиция возможна, то отдельно реализуются технические средства диагностирования элементов памяти и комбинационных составляющих. При этом применяют как методы тестового, так и рабочего диагностирования.

Важным направлением в теории синтеза контролепригодных цифровых систем является использование принципов обеспечения самопроверяемости структур относительно заданной модели неисправности [11, 17, 18].

Система является *полностью самопроверяемой*, если она является самотестируемой и защищенной от неисправностей. Свойство *самотестируемости* подразумевает существование хотя бы одной входной рабочей комбинации, на которой проявляется неисправность из заданного класса, и фиксируется в виде защитной выходной комбинации. Свойство *защищенности от неисправностей* подразумевает вычисление правильных или защитных выходных значений на любых входных рабочих воздействиях при возникновении любой неисправности из заданного класса.

Полностью самопроверяемые системы не требуют дополнительных схем контроля, поскольку при возникновении отказа их выходы отключаются от объектов следующих каскадов и управляемых объектов, а неверно вычисленные данные блокируются и не используются.

При построении самопроверяемых систем избыточность вносится либо в структуру самого исходного объекта (исходное устройство модифицируется), либо проявляется в виде использования дополнительных схем контроля – самопроверяемых схем встроенного контроля (СВК) [54, 58]. В обоих случаях принципы внесения избыточности одинаковы и основаны на использовании избы-

точного кодирования. Для современных устройств автоматики и вычислительной техники, снабжаемых СВК, в качестве избыточных кодов используются двоичные коды, ориентированные на обнаружение (и реже – на исправление) ошибок [12–14, 18, 19, 39, 42, 50, 65]. Применяются коды с различным числом контрольных разрядов, что непосредственно влияет на их характеристики обнаружения ошибок, а также на сложность технической реализации кодирующего и декодирующего оборудования. Как правило, если речь идет о некоторой самопроверяемой системе, то ее принято сравнивать с системой дублирования, в основе которой лежит принцип «повторения» данных (или один из известных кодов с повторением [2]). В такой системе при сравнении вычисленных данных возможна идентификация любых неисправностей в объекте диагностирования, однако существенны и аппаратные затраты на обеспечение указанного свойства. Использование менее избыточных двоичных кодов, чем код с повторением, позволяет снизить аппаратные затраты на реализацию самопроверяемых систем с учетом их специфики и допустимых и вероятных неисправностей в их структурах.

1.3. Рабочее диагностирование устройств

Идентификация неисправностей косвенным образом по результатам вычислений рабочих функций устройством автоматики и вычислительной техники возможна за счет использования дополнительного оборудования, осуществляющего выявление какого-либо диагностического признака. К таким признакам относятся, например, принадлежность формируемого на выходах устройства двоичного вектора заранее выбранному избыточному коду или же принадлежность каждой формируемой функции некоторому особому виду, например особому классу функций алгебры логики (к примеру, так осуществляется контроль устройств по признаку самодвойственности функций [16, 20]). Устройство автоматики и вычислительной техники снабжается техническими средствами диагностирования, осуществляющими процедуры сравнения.

Классической схемой контроля является схема дублирования [41]. Согласно такой схеме объект диагностирования дополняется техническими средствами диагностирования в составе копии объекта диагностирования и самопроверяемого компаратора (рис. 1.1). Объект диагностирования $F(x)$ и его копия $F'(x)$ в процессе эксплуатации формируют идентичные информационные векторы, которые поразрядно сравниваются компаратором TRC [59]. Так как объект диагностирования и технические средства диагностирования реализуются отдельно, в любой момент времени неисправность может возникнуть только в каком-либо из отдельных функциональных блоков. Одновременное же возникновение искажений в обоих блоках при отсутствии конструктивных и производственных дефектов маловероятно. Неисправность в объекте диагностирования приводит к возникновению ошибки на выходе конкретного элемента в его структуре. Далее эта ошибка может быть скомпенсирована или транслирована на рабочие выходы устройства. Трансляция должна осуществляться хотя бы на одном

рабочем входном воздействии, в противном случае устройство не будет контролепригодным. Таким образом, хотя бы на одном входном наборе происходит искажение информационного вектора, формируемого на выходах устройства автоматики и вычислительной техники. Внутренняя ошибка может быть транслирована на выходы устройства в виде ошибки произвольной кратностью и вида, что связано со структурой самого объекта диагностирования, ее свойствами, наличием инверсий по различным путям трансляции ошибок на выходы устройства и т. д.

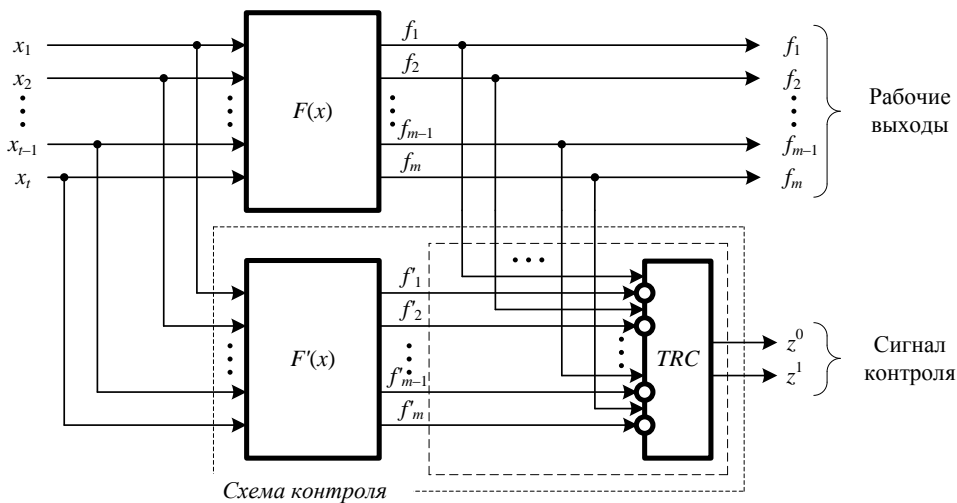


Рис. 1.1. Схема дублирования

При дублировании обнаруживается любая неисправность в объекте диагностирования, так как компаратором обнаруживаются искажения любого вида и любой кратностью. По этой причине схема дублирования нашла широкое применение на практике в решении различных задач обеспечения обнаружения неисправностей и отказоустойчивости систем управления.

Дублирование подразумевает внесение существенной избыточности по отношению к исходному устройству. Зачастую дублирование многократно усложняет систему ввиду сложности реализации компаратора с самопроверяемой структурной схемой [18]. Это обстоятельство приводит разработчиков систем диагностирования к техническому решению, подразумевающему упрощение структуры дополнительных средств контроля. Такое уменьшение возможно при использовании избыточных кодов, в том числе большого класса кодов с суммированием [12, 13].

При дублировании реализуется идея применения кодов с повторением – таких кодов, у которых контрольный вектор повторяет информационный вектор [2]. Избыточность кода, при которой число разрядов контрольного вектора равно числу разрядов информационного вектора, определяет и высокую избыточность системы диагностирования. Использование кодов с суммированием (обозначим их как (m,k) -коды, где m и k – число информационных и контроль-

ных разрядов) приводит к уменьшению числа контрольных разрядов: $k \in \{1, 2, \dots, m-1\}$. Соответственно возможно и уменьшение избыточности технических средств диагностирования.

При использовании кодов с суммированием схема реализации системы диагностирования несколько изменяется по отношению к схеме дублирования (рис. 1.2). Технические средства диагностирования включают в себя блок контрольной логики $G(x)$, который формирует значения разрядов контрольного вектора $\langle g_k g_{k-1} \dots g_2 g_1 \rangle$ и дополняет информационный вектор $\langle f_m f_{m-1} \dots f_2 f_1 \rangle$ до кодового слова заранее выбранного кода с суммированием, а также тестер выбранного кода (*TSC – totally self-checking checker*). Тестер, зачастую реализуемый в виде каскадного соединения кодера выбранного кода $G(f)$ и компаратора *TRC*, позволяет осуществлять контроль принадлежности поступающего на его входы кодового слова выбранному коду с суммированием [11, 17, 59].

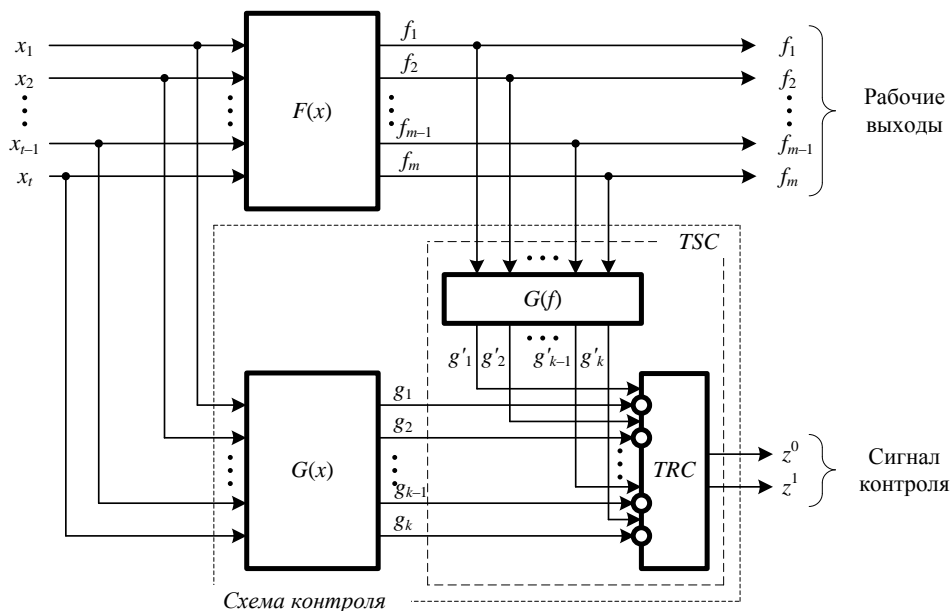


Рис. 1.2. Схема реализации системы диагностирования на основе кодов с суммированием

Схема дублирования является стандартной структурной схемой реализации систем рабочего диагностирования. Эффект от уменьшения аппаратных затрат при использовании вместо схемы дублирования схемы контроля по заранее выбранному коду с суммированием принято оценивать по отношению к показателю сложности технической реализации схемы дублирования [12]:

$$\mu = \frac{L_{(m,k)}}{L_D} \cdot 100\%, \quad (1.1)$$

где $L_{(m,k)}$ – показатель сложности реализации системы диагностирования на основе (m, k) -кода, L_D – показатель сложности реализации системы диагностирования на основе схемы дублирования.

В качестве показателя сложности технической реализации зачастую выступает показатель условной площади, занимаемой устройством на кристалле, рассчитываемой либо в условных единицах площади, либо в конкретных единицах площади (нанометрах).

Чем меньше значение показателя μ , тем, естественно, эффективнее использование кода с суммированием при организации системы диагностирования по сравнению со схемой дублирования.

Следует отметить, что в качестве показателя эффективности использования (m,k) -кодов при организации систем диагностирования может быть использована доля площади, занимаемой системой диагностирования на кристалле, по отношению к площади объекта диагностирования:

$$\varphi = \frac{L_{(m,k)}}{L_{F(x)}} \cdot 100\%. \quad (1.2)$$

Показатель φ , однако, используется гораздо реже, так как он менее нагляден.

Обратным эффектом снижения структурной избыточности конечной системы является невозможность обнаружения полного множества ошибок на выходах объекта диагностирования. В классе необнаруживаемых появляются ошибки с различными кратностями и видами. Характеристики сложности технической реализации и обнаружения ошибок на выходах объектов диагностирования находятся в некотором «противоречии» (являются конфликтующими). Поэтому стоит вопрос выбора наилучшего (m,k) -кода для каждого конкретного объекта диагностирования [36]. На рисунке 1.3 приводится интерпретация зоны поиска наилучшего по критериям сложности реализации (L) и числа обнаруживаемых ошибок (D) способа построения системы рабочего диагностирования. Сложность реализации и число обнаруживаемых ошибок в коде изменяются от минимального значения пары (L_{\min}, D_{\min}) для кода с $k=1$ к максимальному значению пары (L_{\max}, D_{\max}) для кода с $k=m$. Поиск эффективного решения для какого-либо устройства автоматики и вычислительной техники осуществляется из заштрихованной области на рисунке.

Учет характеристик обнаружения ошибок (m,k) -кодами при выборе способа построения системы рабочего диагностирования позволяет уменьшать сложность ее технической реализации по сравнению с дублированием. При определенном установленном ограничении на модель неисправностей (например, учет одиночных константных неисправностей выходов внутренних элементов объекта диагностирования или учет неисправностей задержек пути) возможна идентификация любых их проявлений на выходах объекта диагностирования [12, 13, 18, 42]. В конечном итоге возможно достижение уменьшения не только стоимости разработки технического устройства, снабженного средствами идентификации неисправностей, но и снижение стоимости его эксплуатации (меньшее количество аппаратных средств требует меньших по мощности устройств отведения тепла, что вкупе с уменьшением общего числа элементов приводит и к снижению энергозатрат).

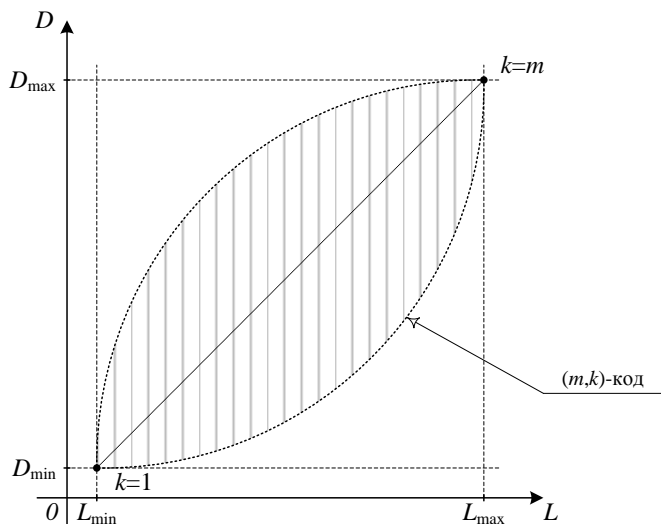


Рис. 1.3. Сравнение способов реализации систем диагностирования по сложности и свойствам обнаружения ошибок

1.4. Основные коды с обнаружением ошибок

При решении задач технической диагностики и синтеза контролепригодных систем автоматики и вычислительной техники широко применяются как избыточные двоичные коды, так и неизбыточные.

Неизбыточные коды не содержат избыточных символов для придания им свойств помехозащищенности. Неизбыточные коды могут быть равномерными или неравномерными.

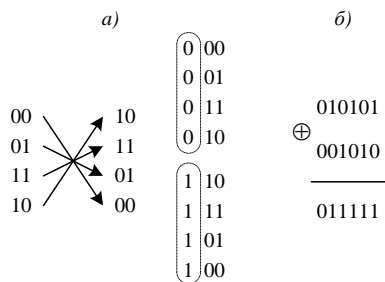


Рис. 1.4. Построение кода Грея: рекурсивное (а) и путем побитового сложения (б)

Среди равномерных неизбыточных кодов распространены обыкновенные двоичные коды и коды Грея. Первые, например, часто используют при кодировании состояний конечных автоматов. Каждому состоянию автомата S_j , $j \in \{1, 2, \dots, n\}$ соответствует кодовая комбинация, включающая в себя $q = \lceil \log_2 n \rceil$ бит (разрядов), где запись $\lceil \dots \rceil$ обозначает целое сверху от вычисляемого значения. Для защиты проектируемого устройства от критических состояний могут быть использованы коды Грея, в которых упорядоченные кодовые слова, т.е. следующие в лексикографическом порядке, отличаются друг от друга только в одном разряде [2].

Например, код Грея для двухразрядных двоичных чисел включает в себя следующую последовательность кодовых комбинаций: {00, 01, 11, 10}. Код Грея для q -разрядных чисел может быть построен рекурсивно на основе кода для $(q - 1)$ -разрядных чисел. Для этого требуется запись кодов в обратном порядке, приписывание к исходным комбинациям нулевого старшего разряда и «отраженного» единичного старшего разряда и последующая конкатенация списков кодовых комбинаций. На рисунке 1.4а) приведен пример построения кода Грея для трехразрядных двоичных чисел. Известен и другой способ построения кода Грея, основанный на побитном сложении по модулю два исходного кодового слова со сдвинутым вправо на один разряд кодовым словом (рис. 1.4б).

Неравномерные двоичные избыточные коды часто используются при синтезе алгоритмов диагностирования на основе первичной информации о вероятностях возникновения дефектов. Например, известные алгоритмы построения бинарных алгоритмов диагностирования, оперирующие именно этими исходными данными, фактически приводят к получению комбинаций кодов Шеннона – Фано и Хаффмана [7]. Рассмотрим два этих алгоритма на примере кодирования шести событий $X_1 - X_6$ с вероятностями $p_1=0,1$, $p_2=0,2$, $p_3=0,15$, $p_4=0,15$, $p_5=0,3$, $p_6=0,1$ (рис. 1.5).

Алгоритм Шеннона – Фано подразумевает ранжирование вероятностей в порядке их невозрастания, а затем постепенное деление на группы таким образом, чтобы в каждой новой группе сумма вероятностей была по возможности одинаковой. Сначала производится разбиение полного множества событий на две группы, затем каждой полученной группы до тех пор, пока все события не будут разделены. Графически этот процесс удобно показывать в виде построения древовидного графа (см. рис. 1.5а). На дугах в графе, расположенных слева, отмечают значение 1, а на дугах, расположенных справа, – значение 0. Каждый маршрут к висячей вершине, соответствующей конкретному событию, включает в себя последовательность нулей и единиц, образующих соответствующую кодовую комбинацию.

Несколько иначе строится код при использовании алгоритма Хаффмана. Он также подразумевает ранжирование вероятностей в порядке их невозрастания. Затем из списка берутся два события с наименьшими вероятностями (вероятности из списка удаляют), им в соответствие ставится вершина графа, вероятности суммируются, а полученное число помещается в список, который вновь ранжируется по убыванию. Процедура повторяется до тех пор, пока не образуется единственная корневая вершина графа (рис. 1.5б).

Оба алгоритма, и Шеннона – Фано и Хаффмана, дают префиксный код, т. е. код, каждая комбинация которого полностью не является началом другой. Это позволяет однозначно декодировать сообщения из потока данных. Алгоритмы построения кодов Шеннона – Фано и Хаффмана позволяют построить алгоритмы диагностирования для технических объектов. При этом в задачах синтеза диагностического обеспечения или устройств с контролепригодными схемами они, как правило, не используются.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru