

В.А. АФОНИН

ОСНОВЫ ТЕОРИИ НАДЕЖНОСТИ

Учебное пособие
по курсу «Основы теории надежности»
для студентов, обучающихся по направлению
«Информатика и вычислительная техника»

Под редакцией И.И. Ладыгина

УДК 621.398
А 946

Утверждено учебным управлением НИУ «МЭИ»
в качестве учебного пособия для студентов

Подготовлено на кафедре вычислительных машин, систем и сетей

Рецензенты: докт. техн. наук проф. А.Б. Фролов,
канд. техн. наук проф. Б.Г. Теряев

Афонин В.А.

А 946 Основы теории надежности [Электронный ресурс] : учебное пособие / В.А. Афонин; под ред. И.И. Ладыгина. — М.: Издательский дом МЭИ, 2016. — Загл. с тит. экрана.

ISBN 978-5-383-01030-3

В настоящем пособии содержится материал по расчету и статистической оценке различных показателей надежности электронных средств вычислительной техники. Особое внимание уделено методам и средствам оперативного контроля, диагностики, обеспечения контролепригодности и тестопригодности как основным формам обеспечения надежности систем.

Настоящее электронное издание подготовлено на основе одноименного печатного издания (2-е изд., стереотип.), вышедшего в Издательском доме МЭИ в 2016 году.

Пособие предназначено для студентов, обучающихся по специальности «Вычислительные машины, комплексы, системы и сети», но различные разделы пособия могут использоваться студентами других специальностей, изучающими надежность цифровой техники.

ОГЛАВЛЕНИЕ

Принятые сокращения и обозначения	6
Введение	10
1. Показатели и пути обеспечения надежности средств вычислительной техники	11
1.1. Основные понятия и определения	11
1.2. Показатели надежности	13
1.3. Рекомендации по выбору показателей надежности вычислительных систем	14
1.4. Основные средства обеспечения надежности вычислительных систем	16
2. Математический аппарат и модели надежности	19
2.1. Аппарат теории вероятностей. Вероятностные модели надежности	19
2.2. Аппарат цепей Маркова. Марковские модели надежности	24
2.3. Аппарат булевой алгебры. Логические модели надежности	29
2.4. Аппарат теории графов. Структурно-графовые модели надежности	31
2.5. Аппарат статистического моделирования. Имитационные модели надежности	34
3. Расчет и оценка показателей надежности нерезервированных невосстанавливаемых объектов	38
3.1. Объекты с произвольной функцией распределения времени безотказной работы	38
3.2. Объекты с показательной функцией распределения времени безотказной работы («экспоненциальные» объекты)	46
4. Показатели надежности восстанавливаемых объектов	48
4.1. Показатели процесса отказов и восстановлений	48
4.2. Показатели процесса отказов/восстановлений с показательной функцией распределения ВБР и ВВ	51
4.3. Комплексные показатели надежности	52
4.4. Марковская модель восстанавливаемого объекта	55
4.5. Время установления стационарного режима	56
5. Анализ надежности невосстанавливаемых многоэлементных систем	58
5.1. Схема надежности. Нерезервированная система без восстановления	58
5.2. Основное соединение	59
5.3. Виды структурного резервирования	61
5.4. Общее и раздельное резервирование с постоянно включенным резервом	62
5.5. Резервирование замещением и скользящее резервирование	66

6. Анализ надежности систем со сложной структурой	72
6.1. Расчет показателей произвольной схемы надежности с использованием логических условий безотказности. Метод структурных функций	72
6.2. Мажоритарное резервирование	77
6.3. Мажорирование функциональных блоков ВС	79
6.4. Системы с реконфигурацией и восстановлением	80
7. Оперативный аппаратный контроль ВС	84
7.1. Задачи и виды оперативного контроля	84
7.2. Мажоритарный контроль	85
7.3. Вычеты и их свойства	87
7.4. Цифровые вычеты	90
7.5. Формирование цифровых вычетов (на примере свертки по mod 3)	92
7.6. Контроль по вычетам	93
7.7. Аппаратная реализация цифрового контроля по вычетам	96
8. Задачи и виды технической диагностики	98
8.1. Формы технической диагностики	98
8.2. Показатели качества диагностирования	99
8.3. Понятие состояния объекта диагностирования. Модели неисправностей	100
8.4. Классы неисправностей цифровых устройств	102
8.5. Задачи и классификация систем тестового диагностирования	104
8.6. Уровни диагностирования	105
9. Организация диагностирования вычислительных систем на макроуровне	107
9.1. Метод раскрутки	107
9.2. Диагностическая модель многопроцессорной ВС с распределенным ядром	108
9.3. Диагностическая модель ВС с центральным ядром (на примере микропроцессорной системы)	116
10. Организация диагностирования цифровых устройств на уровне функциональных модулей	120
10.1. Микродиагностика	120
10.2. Проектирование тестовых микропрограмм	122
10.3. Декомпозиция последовательностного устройства	127
10.4. Синтез тестов комбинационных устройств	128
10.5. Исчерпывающее и псевдослучайное тестирование	134
11. Сигнатурный анализ	137
11.1. Анализ реакций	137
11.2. Сигнатурный анализ методом контроля остатков	138
11.3. Маскировка ошибок при сигнатурном анализе по остаткам (на сдвиговом регистре)	141
11.4. Выбор метода сигнатурного анализа	146

11.5. Эквивалентные схемы сигнатурного анализатора на сдвиговом регистре	148
11.6. Генерация ПС-последовательностей	152
12. Алгоритмы диагностирования	153
12.1. Классификация	153
12.2. Безусловные алгоритмы для модели одиночных неисправностей	155
12.3. Условные алгоритмы для модели одиночных неисправностей	156
13. Повышение тестопригодности СВТ	163
13.1. Общие принципы повышения контролепригодности и тестопригодности СВТ	163
13.2. Метод сканирования	165
13.3. Разновидности метода сканирования	167
13.4. Повышение диагностируемости ВС на структурном уровне	171
13.5. Конденсация графа управляемости / наблюдаемости и выделение МСС-подграфов	175
14. Построение самопроверяемых СВТ	179
14.1. Схемы самопроверяемого дублирования	179
14.2. Проектирование ССВК комбинационной логики	181
15. Построение легкотестируемых схем	196
15.1. Идея построения легкотестируемых схем	196
15.2. Порядок построения легкотестируемых схем	198
Контрольные вопросы	201
Список литературы	206

ПРИНЯТЫЕ СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ

Принятые сокращения

ВБР — время безотказной работы объекта;
ВВ — время восстановления объекта;
ВП — выходная последовательность сигналов объекта диагностирования;
ВС — вычислительная система;
ДДУ — дополнительное дискретное устройство;
ДС — диагностический словарь;
КЧ — комбинационная часть;
м.о. — математическое ожидание случайной величины;
м.п. — марковский процесс;
МСС-подграф — максимально сильносвязный подграф;
о/в — отказ/восстановление;
ОД — объект диагностирования;
ОпК — оперативный контроль;
П — элемент памяти, *D*-триггер;
Порт в/в — порт ввода/вывода;
ПС — псевдослучайная (последовательность);
СА — сигнатурный анализ;
с.в. — случайная величина;
СВК — схема встроенного контроля;
СВТ — средства вычислительной техники;
с.с. — случайное событие;
СС — схема сравнения;
ССВК — самопроверяемая схема встроенного контроля;
ТН — таблица неисправностей;
ТПС — таблица пар состояний;
ТФН — таблица функций неисправностей;
ф.р. — функция распределения случайной величины;
ЭУР — элемент управляемого разрыва.

Принятые обозначения

Главы 1—6

θ — случайное время наработки объекта до первого отказа;
 ξ — случайное время восстановления объекта;
 η — случайное время между последовательными моментами восстановления в альтернирующем потоке отказов (с ненулевым временем восстановления);

$F_{\theta}(t), f_{\theta}(t)$ — функция и плотность (соответственно) распределения случайного времени θ ;

$F_{\xi}(t), f_{\xi}(t)$ — функция и плотность (соответственно) распределения времени восстановления ξ ;

$F_{\eta}(t)$ — функция распределения случайного времени η ;

$M[]$ — математическое ожидание случайной величины, заключенной в квадратные скобки;

$P(t), Q(t)$ — вероятность соответственно безотказной работы и отказа объекта в интервале от 0 до t ;

$P(t, t + \Delta t), Q(t, t + \Delta t)$ — условная вероятность соответственно безотказной работы и отказа объекта в интервале от t до $t + \Delta t$, если в момент t объект был работоспособен;

P_c, Q_c — вероятность соответственно безотказной работы и отказа объекта, содержащего несколько элементов;

T — средняя наработка до отказа невосстанавливаемого объекта;

T_c — средняя наработка до отказа невосстанавливаемой многоэлементной системы, содержащей несколько элементов;

$\lambda(t)$ — интенсивность отказа элемента системы в момент времени t ;

$\Lambda_c(t)$ — интенсивность отказа системы, содержащей несколько элементов, в момент t ;

μ — интенсивность восстановления элемента;

t_{γ} — гамма-процентная наработка — интервал времени, при котором обеспечивается вероятность работоспособности объекта не ниже P_{γ} ;

α — параметр масштаба в распределении Вейбулла—Гнеденко;

β — параметр формы распределения Вейбулла—Гнеденко;

$m(t, t + \Delta t)$ — м.о. числа отказов (восстановлений) в точечном потоке отказов с мгновенным восстановлением;

$\omega_0(t)$ — параметр точечного потока отказов при мгновенном восстановлении;

$\omega(t)$ — параметр альтернирующего потока отказов (с ненулевым временем восстановления);

T_0 — средняя наработка на отказ (между отказами) восстанавливаемого объекта;

$\mu(t)$ — интенсивность восстановления объекта в момент времени t ;

$T_{\text{в}}$ — среднее время восстановления;

$T_{\text{ов}}$ — среднее время между отказами (или моментами восстановления) объекта с ненулевым временем восстановления;

$R(t), S(t)$ — вероятность соответственно восстановления и невосстановления объекта в интервале от 0 до t ;

$R(t, t + \Delta t)$, $S(t, t + \Delta t)$ — вероятность соответственно восстановления и невозстановления объекта в интервале от t до $t + \Delta t$ при условии, что в момент t он находился в состоянии отказа;

$k_r(t)$ и k_r — нестационарный и стационарный коэффициенты готовности соответственно;

$k_{\Pi}(t)$ и k_{Π} — нестационарный и стационарный коэффициенты простоя соответственно;

$k_{o,r}(t, t + \Delta t)$ — коэффициент оперативной готовности — совместная вероятность того, что восстанавливаемый объект будет работоспособен в момент времени t и проработает безотказно еще в течение интервала Δt ;

$F(x)$ — структурная функция системы;

$\mathbf{x} = (x_1 \ x_2 \ \dots \ x_n)$ — вектор n переменных;

$\lceil \]$ — верхняя грань — минимальное целое число, не меньшее числа, заключенного в скобки.

Индекс «с» при любом показателе надежности обозначает, что объектом является система, состоящая из некоторого множества элементов.

Главы 7—15

m — модуль (целое положительное число, большее 1);

$r(A) \bmod m$ — целый неотрицательный остаток от деления целого числа A на модуль $m > 1$;

СПД — система передачи данных;

$\lfloor \]$ — нижняя грань — максимальное целое число, не большее числа, заключенного в скобки.

Глава 9

G — множество исправных модулей в модели многопроцессорной системы;

$|G|$ — число исправных модулей (мощность множества G);

F — множество неисправных модулей в системе;

$|F|$ — число неисправных модулей;

t — число неисправных модулей в системе, при котором возможно правильное заключение о состоянии системы;

τ — максимальное значение t , при котором возможно правильное заключение о состоянии системы;

$\Gamma^-(v_i)$ — множество вершин — непосредственных предшественников вершины v_i в графовой модели системы с распределенным ядром;

$|\Gamma^-(v_i)|$ — число таких вершин (мощность множества $\Gamma^-(v_i)$);

$\lceil \]$ — верхняя грань — минимальное целое число, не меньшее числа, заключенного в скобки;

$\lfloor \]$ — нижняя грань — максимальное целое число, не большее числа, заключенного в скобки;

a_{ij} — реакция j -го модуля на тестовое воздействие со стороны i -го модуля (симптом);

A — синдром реакций системы на тестовые воздействия.

Глава 10

μ_k, M — микрокоманда;

$\mu\Pi, P$ — микропрограмма;

y_k^* — константная неисправность в цепи y_k устройства;

y_k' — активизированная цепь устройства,

$dF(y_k)/dy_k$ — булева производная выходной переменной F устройства относительно переменной y_k ;

a_i — i -е состояние объекта диагностирования.

Глава 11

W — сигнатура;

$k_{\text{пп}}$ — коэффициент полноты проверок;

$k_{\text{гд}}$ — коэффициент глубины диагностирования.

Глава 12

$l_{\text{ср}}$ — средняя длина проверки (объем диагностического эксперимента).

ВВЕДЕНИЕ

Надежность является одной из самых важных характеристик технических изделий, что объясняет большой объем исследований по данной тематике во всех сферах промышленности. К настоящему времени издано большое число монографий и учебной литературы, частично приведенной в списке литературы. В учебных заведениях изучение задач и методов обеспечения надежности связано со специальностью и особенностью подготовки учащихся. Наиболее сложными объектами являются средства цифровой техники, особенно использующие элементы высокой степени интеграции. Самым распространенным способом поддержания требуемого уровня надежности является автоматизированная проверка состояний объекта, а также повышение степени ремонтпригодности электронных изделий.

Предлагаемое учебное пособие предназначено в первую очередь для изучения путей обеспечения надежности и диагностики средств вычислительной техники (СВТ). Оно содержит материал, излагаемый в курсе «Основы теории надежности» для студентов специальности «Вычислительные машины, комплексы, системы и сети».

Главы 1—6 посвящены расчету и оценке различных показателей надежности. В главах 7—12 рассматриваются методы контроля и диагностики СВТ, в главах 13—15 — пути повышения ремонтпригодности и построения самопроверяемых цифровых изделий.

Приводимые в книге теоремы в ряде случаев даются без доказательств, доказательства можно найти в приводимой литературе. Для поясняющих модельных примеров частично использовано пособие автора «Обеспечение надежности СВТ: сб. задач [3], а также статьи и монографии различных авторов. При ссылках на пособие [3] оно будет кратко именоваться «Задачник».

Автор выражает благодарность профессору И.И. Ладыгину за работу по редактированию пособия, а также профессору А.Б. Фролову и профессору Б.Г. Теряеву за замечания и пожелания по улучшению рукописи.

1. ПОКАЗАТЕЛИ И ПУТИ ОБЕСПЕЧЕНИЯ НАДЕЖНОСТИ СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

1.1. Основные понятия и определения

Надежность изделий определяется рядом критериев, выбор которых зависит от вида объектов и особенностей их эксплуатации. Эти критерии могут быть противоречивыми и изменяться при изменении условий использования изделий.

1) *Надежность* — свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах.

2) *Отказоустойчивость* — способность объекта выполнять свои функции при наличии неисправностей, но, возможно, с меньшей эффективностью (производительностью). При этом предполагается, что неисправности обусловлены внутренними причинами — надежностью элементов, которую можно прогнозировать, используя вероятностные и статистические методы.

3) *Живучесть* — устойчивость к влиянию внешних факторов (стихийные бедствия, аварийные изменения режимов работы, умышленные повреждения и т.д.). Здесь не применимы вероятностные методы прогнозирования, и для оценки и обеспечения живучести используются методы теории графов и другие методы структурного анализа.

4) *Исправность* — состояние объекта, при котором его характеристики соответствуют всем требованиям нормативно-технической и конструкторской документации.

Повреждение — нарушение хотя бы одного требования документации (т.е. исправности). То же, что *неисправность*.

Дефект — неисправность как физическое явление.

5) *Работоспособность* — состояние объекта, при котором его характеристики соответствуют требованиям документации, определяющим способность системы выполнять заданные функции.

Отказ — нарушение работоспособности. То же, что *неработоспособность*.

Сбой — самоустраняющийся отказ, приводящий к кратковременному нарушению работоспособности.

Перемежающийся отказ — многократно возникающий сбой одного и того же характера.

Устойчивый отказ — невозстанавливаемый самостоятельно отказ (в отличие от сбоя).

Внезапный отказ — отказ, характеризующийся скачкообразным изменением параметров (кратковременным по сравнению с периодом между сеансами контроля).

Постепенный отказ — отказ, характеризующийся медленным изменением параметров.

Ошибка — искаженное логическое значение сигналов на внешних выводах устройства (результат отказов или ошибок пользователя).

Действие сбоев и ошибок обычно приводит к устойчивым отказам вычислительной системы, требующим принудительного восстановления (перезапуск программ или перезагрузка машины). Отметим, что кратность ошибки может быть больше кратности неисправностей (вследствие наличия разветвлений).

6) *Безотказность* — свойство объекта сохранять работоспособность в течение заданной наработки.

Долговечность — свойство объекта сохранять работоспособность в течение срока службы с учетом времени обслуживания и ремонта.

7) *Диагностика* — определение состояния объекта (исправен / неисправен и что неисправно).

8) *Ремонтопригодность* — степень приспособленности объекта к восстановлению. Увеличение ремонтопригодности сокращает время восстановления, которое состоит из времени обнаружения неисправности (обеспечивается устройствами контроля), времени локализации неисправности (обеспечивается системой диагностирования) и времени ремонта.

9) *Контролепригодность, тестопригодность* — степень приспособленности объекта к обнаружению и локализации неисправности (дефекта) в процессе его рабочего функционирования или тестирования. Эти показатели непосредственно влияют на ремонтопригодность.

10) *Основной элемент* — элемент структуры объекта, необходимый для выполнения функций системы при отсутствии отказов других элементов.

Резервный элемент — элемент структуры объекта, выполняющий функции основного элемента при его отказе.

1.2. Показатели надежности

1. Показатели безотказности:

1) *Вероятность безотказной работы* в заданном интервале $P(0, t) \equiv P(t)$ и $P(t, t + \Delta t)$.

Вероятность отказа $Q(0, t) \equiv Q(t)$ и $Q(t, t + \Delta t)$.

2) *Средняя наработка до отказа* (для невосстанавливаемых объектов) T — матожидание наработки до первого отказа.

Средняя наработка на отказ (для восстанавливаемых объектов) T_0 — среднее время между отказами при мгновенном восстановлении, или отношение наработки объекта к матожиданию числа отказов в течение этой наработки при мгновенном восстановлении.

3) *Интенсивность отказов* $\lambda(t)$ — условная плотность распределения времени безотказной работы (ВБР) в момент t , если до этого момента отказа не было. Используется только для невосстанавливаемых объектов.

Параметр потока отказов (восстановлений) ω_0 — среднее число отказов (восстановлений) в единицу времени. Используется только для объектов с мгновенным восстановлением.

Для нестационарных потоков $\omega_0 = \omega_0(t)$, для стационарных — $\omega_0 = 1/T_0$.

4) *γ -процентная наработка* T_γ — наработка невосстанавливаемого объекта, при которой $P(0, t)$ снижается до заданной величины P_γ (или γ в %).

Аналогично определяются показатели, связанные со сбоями: вероятность бессбойной работы в заданном интервале $P_{сб}(t)$, интенсивность сбоев $\lambda_{сб}$, параметр потока сбоев $\omega_{сб}(t)$, γ -процентная наработка на сбой $T_{\gamma сб}$.

2. Показатели ремонтпригодности:

1) *Вероятность восстановления* в заданном интервале $R()$ и невосстановления $S()$, в скобках указывается интервал времени.

2) *Среднее время восстановления* (ВВ) $T_{в}$.

Эти показатели зависят от контролепригодности и тестопригодности объекта.

3. Комплексные показатели надежности:

1) *Коэффициент готовности* $k_r(t)$ — вероятность того, что объект будет в работоспособном состоянии в момент времени t .

Коэффициент простоя $k_{\Pi}(t) = 1 - k_{\Gamma}(t)$.

2) *Коэффициент оперативной готовности $k_{о.г.}(t, t + \Delta t)$ — вероятность того, что объект в момент t будет в работоспособном состоянии и затем в течение интервала Δt не откажет.*

3) *Коэффициент технического использования $k_{ТИ}$ — доля времени работоспособного состояния объекта за полное время его эксплуатации. Этот коэффициент является показателем долговечности восстанавливаемых объектов и равен*

$$k_{ТИ} = T_0 / (T_0 + T_{\text{в}} + T_{\text{проф}}),$$

где $T_{\text{проф}}$ — среднее время профилактики.

Эти коэффициенты являются показателями безотказности с учетом ремонтпригодности восстанавливаемых объектов.

4. Показатели живучести и отказоустойчивости сетей ЭВМ и многопроцессорных вычислительных систем. При анализе живучести сеть или вычислительная система (ВС) представляются в виде неориентированного графа, вершинами которого являются рабочие станции, хост-машины, процессоры, концентраторы и т.д., а дугами — технические или логические средства передачи данных. Статистические характеристики безотказности элементов при этом не учитываются. В этом случае основным показателем живучести является коэффициент связности вершин сети. В свою очередь, задача анализа и обеспечения надежности зависит от назначения сети: это может быть связность отдельной пары объектов, связность сервера с рабочими станциями или взаимная связность всех компьютеров в сети. Изменение конфигурации сети с увеличением числа путей между вершинами графа может повысить ее надежность.

В отличие от живучести, при анализе отказоустойчивости сетей учитываются статистические характеристики безотказности элементов. Показателем отказоустойчивости является вероятность связности вершин графовой модели сети.

1.3. Рекомендации по выбору показателей надежности вычислительных систем

Рассмотренные выше показатели могут оказаться противоречивыми. Пусть из двух объектов с вероятностями безотказной работы $P_1(t)$ и $P_2(t)$ в интервале от 0 до t (рис. 1.1) хотим определить лучший. Из

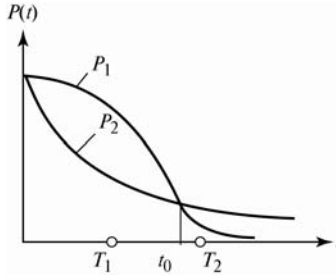


Рис. 1.1

графиков видно, что при $t < t_0$ вероятность безотказной работы первого объекта больше, и следует выбрать именно его, а при $t > t_0$ предпочтительнее оказывается второй объект. Однако если выбор выполняется по критерию средней наработки до отказа T_i , то $T_1 < T_2$, и следует выбрать второй объект.

В таких многокритериальных задачах нужно уметь правильно выбирать главный критерий, а остальные критерии использовать как ограничения.

1) Если ВС восстанавливаемая и систематически решает оперативные задачи большой длительности, то в качестве главного критерия целесообразно использовать $k_{0.г.}$

2) Для восстанавливаемых ВС, решающих задачи незначительной длительности, главным критерием может служить $k_г.$

3) Если ВС при включении диагностируется и при необходимости восстанавливается, а в процессе дальнейшей работы не восстанавливается, то в качестве главного критерия выбирается $P(t)$ или $P(t, t + \Delta t)$.

4) Если прибыль от эксплуатации ВС или эксплуатационные расходы пропорциональны времени использования системы, то главным критерием является T или T_0 (для невосстанавливаемых и восстанавливаемых систем соответственно).

5) При назначении сроков профилактики, гарантийных сроков и при решении ответственных задач, когда уровень $P(t)$ должен оставаться высоким, в качестве главного критерия выбирается $T_г.$

6) При анализе живучести вычислительных сетей и объектов, для которых можно использовать сетевые модели, применяются показатели парной связности, корневой и сетевой связности.

7) При анализе отказоустойчивости ВС используются вероятностные характеристики связности, а также общие критерии надежности, рассмотренные выше.

1.4. Основные средства обеспечения надежности вычислительных систем

1. Повышение надежности элементной базы. Надежность элементной базы связана с совершенствованием технологии производства и повышением степени интеграции. Повышение надежности при этом достигается, в частности, за счет:

- сокращения числа внешних соединений и длины внутренних соединительных проводников, что уменьшает вероятность сбоев;
- уменьшения потребляемой мощности.

Однако для удовлетворения современных требований к надежности ВС интенсивность отказов существующих элементов должна быть уменьшена на 4—5 порядков. Кроме того, требования к надежности систем постоянно возрастают.

2. Резервирование. Целью резервирования является маскирование неисправностей. Различают четыре вида резервирования.

1) Структурное резервирование.

Теоретически при увеличении кратности резервирования вероятность безотказной работы стремится к единице. Однако при этом быстро растут аппаратные затраты и соответственно стоимость. Например, при использовании нагруженного резерва с одним резервным элементом средняя наработка до отказа системы увеличивается в 1,5 раза, при двух резервных элементах — в 1,8 раза, а при 10 резервных элементах — только в 3 раза. Таким образом, структурное резервирование наиболее эффективно при малой кратности резерва.

2) Временное резервирование.

Широкое использование временного резервирования объясняется значительно меньшими аппаратными затратами за счет ухудшения временных показателей (производительности). Примером может служить повторение вычислений или передачи данных. Если же использовать более быстродействующие элементы или просто увеличивать тактовую частоту, то возрастает интенсивность сбоев.

3) Информационное резервирование.

Примером являются корректирующие коды. Недостатки те же, что и для временного резервирования.

4) Функциональное резервирование.

Примеры: системы с перестраиваемой структурой, пакетная передача в сетях ЭВМ, спарки вычислительных машин, при которых в обычном режиме машины работают независимо, а при отказе одной

из них вторая машина берет на себя ее функции, решая при этом и свои задачи.

3. **Диагностика и восстановление.** Этот вид обеспечения надежности является одним из наиболее экономичных и эффективных. Чем меньше интервал между сеансами диагностирования (и соответственно восстановления), тем выше вероятность безотказной работы системы в пределах этого интервала.

4. **Профилактика.** При профилактических работах выявляются скрытые дефекты и постепенные отказы.

5. **Повышение контролепригодности и тестопригодности на стадии проектирования изделия.** Это одно из наиболее мощных средств повышения надежности, позволяет существенно уменьшить T_v и соответственно увеличить k_r .

6. **Использование встроенных средств функционального контроля и диагностирования изделий.** Считается оправданным использование таких средств, если дополнительные аппаратные затраты в расчете на один ventиль не превышают 20 %. Затраты уменьшаются при увеличении степени интеграции. На рис. 1.2 через N_0 обозначено число ventилей в кристалле, через N — число ventилей средств диагностирования. Тогда отношение N/N_0 характеризует дополнительные затраты на ventиль кристалла. При встраивании диагностических средств указанный уровень затрат достигается при числе ventилей в кристалле СБИС порядка 2000 и более (показано стрелкой). Это одно из наиболее перспективных направлений обеспечения надежности вычислительных средств.

7. **Повышение надежности программного обеспечения (ПО).** Ошибки в программном обеспечении приводят к тем же последствиям, что и отказ аппаратуры. Существует ряд путей обеспечения надежности ПО.

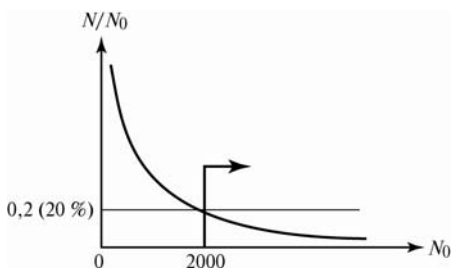


Рис. 1.2



Рис. 1.3

- 1) Структурное программирование.
- 2) Верификация программ (статическая проверка на стадии отладки, т.е. проверка программ без их исполнения).
- 3) Тестирование программ (динамическая проверка на стадии отладки, т.е. запуск программ и сравнение результатов прогона со спецификациями).
- 4) Резервирование ПО. Примером может служить мажорирование ПО, при котором используются три версии программы, промежуточные результаты которых анализируются драйвером голосования (рис. 1.3). При совпадении хотя бы двух результатов выполнение программ продолжается, а при несовпадении всех трех результатов анализируется причина ошибки или фрагмент программы выполняется повторно. Совпадение двух или трех неверных результатов считается весьма маловероятным.

2. МАТЕМАТИЧЕСКИЙ АППАРАТ И МОДЕЛИ НАДЕЖНОСТИ

2.1. Аппарат теории вероятностей. Вероятностные модели надежности

При расчете надежности невосстанавливаемых изделий наиболее часто используются вероятностные модели. Ниже приводятся некоторые понятия теории вероятностей, используемые в теории надежности.

Пусть статистический эксперимент имеет некоторое множество заранее непредсказуемых исходов, называемых *случайными событиями* (с.с.). Под *элементарным* с.с. e_i будем понимать неделимое (атомарное, точечное) с.с. (неопределяемое понятие). Множество E всех возможных элементарных событий образует *полную группу* с.с. и называется *выборочным пространством* или *полем с.с.*:

$$E = \{e_1, e_2, \dots, e_n\}.$$

Рассмотрим множество S всех подмножеств с.с. из E :

$$S = \{s_1, s_2, \dots, s_m\}.$$

Пусть S удовлетворяет следующим требованиям:

- S содержит достоверное событие $\{E\}$;
- S содержит невозможное событие $\{\emptyset\}$ (пустое множество);
- если с.с. A и $B \in S$, то \bar{A} и $\bar{B} \in S$, $A \cap B \in S$, $A \cup B \in S$, причем последнее соотношение справедливо как для конечного числа с.с., так и для бесконечного.

Определенное таким образом множество S называется *борелевским полем с.с.*, или *σ -алгеброй борелевских множеств s_k* . Каждое с.с. A совпадает с одним из борелевских множеств s_k . Если A содержит элементарное с.с. e_i , то появление e_i влечет за собой появление A . В частности, появление любого e_i означает появление E , поэтому E есть достоверное событие, а $\{\emptyset\}$ — невозможное событие, поскольку не содержит ни одного элементарного с.с. e_i .

Отметим, что если множество S конечно, то оно содержит ровно 2^n элементов:

$$S = \{\emptyset, \{e_1\}, \dots, \{e_n\}, \{e_1, e_2\}, \{e_1, e_3\}, \dots, \{e_1, \dots, e_n\}\}.$$

Если события A и $B \in S$ не содержат общих точек, то они называются *несовместимыми*. События e_i всегда несовместимы, так как неделимы, а события s_k могут быть как несовместимыми, так и совместимыми, т.е. содержат общие точечные события.

В пространстве элементарных с.с. E с алгеброй S каждому с.с. A_i ставится в соответствие число $P(A_i)$, называемое его вероятностью, удовлетворяющее следующим требованиям:

- $0 \leq P(A_i) \leq 1$;
- при $A = E$: $P(A) = 1$, при $A = \emptyset$: $P(A) = 0$;
- если A_i — несовместимые события, то $P(\cup A_i) = \sum_i P(A_i)$.

Тройка $\langle E, S, P \rangle$ называется *вероятностным пространством над полем с.с.*

Характеристикой с.с. является *распределение вероятностей* $\{P(e_i)\}$.

A и B называются *зависимыми* с.с., если появление A изменяет вероятность события B . Мерой зависимости является *условная вероятность*, которая определяется как

$$P(B/A) = P(A, B) / P(A),$$

где $P(A, B)$ означает $P(A \cap B)$.

Правило умножения вероятностей:
для независимых событий A, B, C

$$P(A, B, C) = P(A) \cdot P(B) \cdot P(C),$$

для зависимых событий

$$P(A, B, C) = P(A) \cdot P(B/A) \cdot P(C/A, B).$$

Правило сложения вероятностей:
для несовместимых событий A, B, C, D :

$$P(A \cup B \cup C \cup D) = P(A) + P(B) + P(C) + P(D),$$

для совместимых

$$\begin{aligned} P(A \cup B \cup C \cup D) &= [P(A) + P(B) + P(C) + P(D)] - \\ &- [P(A, B) + P(A, C) + P(A, D) + P(B, C) + P(B, D) + P(C, D)] + \\ &+ [P(A, B, C) + P(A, B, D) + P(A, C, D) + P(B, C, D)] - P(A, B, C, D) = \\ &= \sum_{i=1}^m (-1)^{i-1} P_i, \end{aligned}$$

где m — число квадратных скобок, а P_i — сумма в i -й скобке.

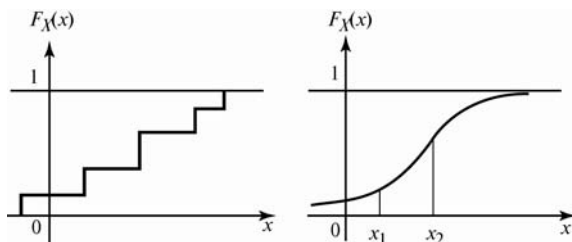


Рис. 2.1

Случайная величина (с.в.) X — вещественная переменная, значения которой определены на каждой точке пространства E . Эти значения (реализации с.в.) будем обозначать строчными буквами x_i . Во многих приложениях, в частности в теории надежности, использование с.в. более удобно, чем с.с.

Характеристики с.в.:

- распределение вероятностей с.в. $\{P(x_i)\}$ (определено для дискретных с.в.);
- функция распределения с.в. $F_X(x) = P[X < x]$, где X — с.в., а x — независимая переменная (определена как для дискретных, так и для непрерывных с.в., рис. 2.1);
- плотность распределения с.в. (определена только для непрерывных с.в., рис. 2.2);

$$f_X(x) = dF_X(x)/dx,$$

- моменты распределения.

Свойства функции распределения:

- $0 \leq F_X(x) \leq 1$;
- $F_X(-\infty) = 0, F_X(\infty) = 1$;
- $F_X(x)$ — неубывающая функция.

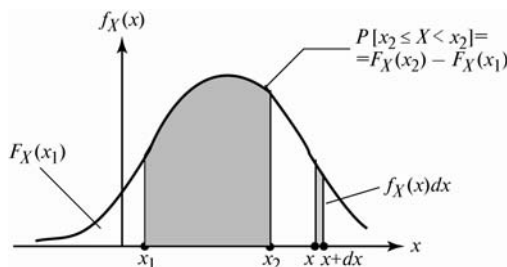


Рис. 2.2

Функция распределения (ф.р.) связана с плотностью выражением

$$F_X(x) = \int_{-\infty}^x f_X(x) dx .$$

Тогда (рис. 2.1, 2.2)

$$P[x_1 \leq X < x_2] = F_X(x_2) - F_X(x_1) = \int_{x_1}^{x_2} f_X(x) dx ,$$

$$F_X(x) = \int_{-\infty}^{\infty} f_X(x) dx = 1 ,$$

$$f_X(x)dx = P[x \leq X < x + dx] = P(dx) \text{ — элемент вероятности.}$$

Математическое ожидание дискретной случайной величины

$$M[X] = \sum_i x_i P(x_i) ,$$

непрерывной —

$$M[X] = \int_{-\infty}^{\infty} f_X(x) dx = \int_{-\infty}^{\infty} x dF_X(x) ,$$

а *дисперсия* —

$$D[X] = M(X - M[X])^2 = M[X^2] - (M[X])^2 .$$

В теории надежности с.в. является случайное время θ от 0 до момента отказа изделия — время безотказной работы (ВБР). При этом всегда предполагается, что при $t = 0$ значение $P(0, t) = P(0, 0) = 1$. Поэтому обычно вместо обозначения $P(0, t)$ используется $P(t)$, а функция и плотность распределения ВБР отлична от 0 только при $t > 0$:

$$F_{\theta}(t) = P[\theta < t],$$

$$f_{\theta}(t) = dF_{\theta}(t)/dt.$$

Кроме функции и плотности распределения, в теории надежности широко используется относительная плотность распределения ВБР

$$\lambda(t) = f_{\theta}(t)/(1 - F_{\theta}(t)),$$

которая называется *интенсивностью отказов*.

Функция $F_{\theta}(t)$ связана с λ соотношением

$$F_{\theta}(t) = 1 - e^{-\varphi(\lambda(t), t)} .$$

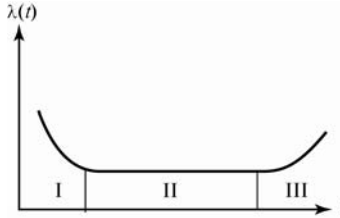


Рис. 2.3

Если $\lambda(t) = \lambda = \text{const}$, то

$$\varphi(\lambda(t), t) = \lambda t$$

$$\text{и } F_0(t) = 1 - e^{-\lambda t}.$$

Это показательная функция распределения ВБР. Для других законов λ зависит от времени.

Для элементов вычислительной техники (ВТ) величина λ меняется за время эксплуатации (рис. 2.3). Период II характерен для приработочного элемента. В течение этого периода λ сохраняет приблизительно постоянное значение, и ВБР элемента распределено экспоненциально. Период I определяется временем приработки, в течение которого λ уменьшается («молодеющий элемент»). В конце срока службы элемента (период III) λ возрастает («стареющий элемент»).

Некоторые полезные формулы теории вероятностей:

- $p(y/x) = p(x, y) / p(x)$.

$$p(z/x, y) = p(x, y, z) / p(x, y) = \frac{p(x)p(y, z/x)}{p(x)p(y/x)} = p(y, z/x) / p(y/x).$$

Отметим, что в общем случае $p(x/y) \neq p(y/x)$.

- Пусть $\{y_1, y_2, \dots\}$ — несовместимые события или величины. Тогда

$$p(x_i) = \sum_j p(x_i, y_j) = \sum_j p(y_j) p(x_i/y_j).$$

Это выражение носит название *формулы полной вероятности*. Аналогично

$$p(x_i) = \sum_j \sum_k p(x_i, y_j, z_k) \text{ и т.д.}$$

- Из $p(x_i/y_j) = p(x_i, y_j)/p(y_j)$, разлагая на множители числитель и используя в знаменателе формулу полной вероятности, находим

$$p(x_i / y_j) = \frac{p(x_i) p(y_j / x_i)}{\sum_j p(x_i) p(y_j / x_i)}.$$

Данное выражение называется *формулой вероятности гипотез* или *формулой Байеса*.

- Пусть $\{x_1, x_2, \dots\}$ — несовместимые события, образующие полную группу, $\{y_1, y_2, \dots\}$ и $\{z_1, z_2, \dots\}$ — тоже несовместимые события и тоже образуют полную группу. Тогда

$$\sum_i p(x_i) = \sum_j p(y_j) = \sum_k p(z_k) = \sum_i \sum_j p(x_i, y_j) = \sum_i \sum_j \sum_k p(x_i, y_j, z_k) = 1.$$

2.2. Аппарат цепей Маркова. Марковские модели надежности

Марковские модели чаще всего используются для анализа восстанавливаемых систем, значительно реже — для невосстанавливаемых. При расчетах характеристик надежности почти всегда предполагается, что функции распределения времени безотказной работы и восстановления элементов системы являются экспоненциальными.

Случайная функция $X(y)$ — это зависимость случайной величины X от независимой переменной y .

Стохастический процесс (с.п.) — это случайный процесс, развивающийся во времени, т.е. $X(t)$. Графически он может быть изображен только в виде своих реализаций $x(t)$ (рис. 2.4).

С.п. $X(t)$ может быть непрерывным (по параметру x) или дискретным. В свою очередь, оба эти процесса могут быть с непрерывным или дискретным временем. На рис. 2.5, *а* изображен дискретный с.п. с непрерывным временем, на рис. 2.5, *б* — с дискретным временем.



Рис. 2.4

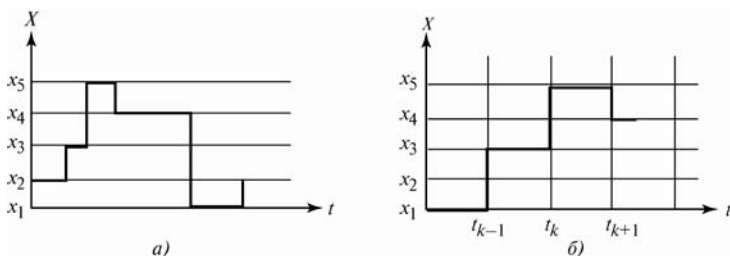


Рис. 2.5

Марковский процесс (м.п.) — это стохастический *дискретный* процесс с непрерывным или дискретным временем, обладающий тем свойством, что вероятность перехода из одного дискретного состояния x_i в другое состояние x_j зависит только от текущего состояния x_i в текущий момент времени и не зависит от состояний, в которых он находился в предшествующие моменты времени:

$$P[X(t_{k+1}) = x_j / X(t_k) = x_i, X(t_{k-1}) = x_l, \dots] = P[X(t_{k+1}) = x_j / X(t_k) = x_i] = p_{ij}.$$

Условные вероятности p_{ij} называются *переходными вероятностями* из состояния i в состояние j .

Процесс с дискретным временем называется *дискретным м.п.*, а с непрерывным временем — *непрерывным м.п.* В теории надежности чаще используются непрерывные м.п., поскольку ВБР — непрерывная с.в.

Рассмотрим два близких значения t и $t + dt$. Тогда вероятность перехода из i -го состояния в j -е за малый интервал dt равна

$$p_{ij}(t, t + dt) = \lambda_{ij}(t)dt,$$

где параметр λ_{ij} называется *интенсивностью перехода* из состояния x_i в состояние x_j за dt .

Отметим, что интенсивность перехода λ_{ij} , в общем случае не совпадает с *интенсивностью отказа* элемента λ_i или *интенсивностью его восстановления* μ_i , но определяется этими характеристиками.

Марковская цепь — изображение в виде графа процесса переходов из состояния в состояние: дуги — переходы, вершины x_i — состояния. Дугам приписываются значения λ_{ij} , вершинам — вероятности состояний $p_i \equiv p(x_i)$ (рис. 2.6).

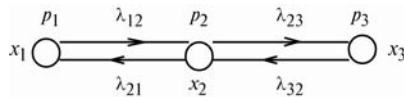


Рис. 2.6

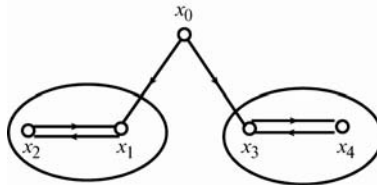


Рис. 2.7

Если интенсивности переходов не зависят от времени, т.е. $\lambda_{ij}(t) = \lambda_{ij}$, то м.п. называется *однородным*. Если все p_i не зависят от t , то процесс называется *стационарным*. Однородный процесс не обязательно является стационарным, но при $t \rightarrow \infty$ однородный процесс становится стационарным. Если при $t \rightarrow \infty$ все p_i не зависят от начального состояния x_k (в момент $t = 0$), то м.п. называется *эргодическим*. Пример неэргодического стационарного процесса приведен на рис. 2.7. Из состояния x_0 процесс случайным образом переходит в одно из двух эргодических подмножеств $\{x_1, x_2\}$ или $\{x_3, x_4\}$, в котором в дальнейшем и остается. В целом же процесс неэргодический, так как, например, p_4 зависит от того, в каком состоянии находился процесс при $t = 0$ — в x_0 , x_1 , x_2 , x_3 или x_4 .

При анализе надежности восстанавливаемых систем обычно используются однородные стационарные эргодические марковские цепи.

Пример 2.1. Система состоит из двух ЭВМ, одна из которых находится в нагруженном резерве (рис. 2.8); λ_1 и λ_2 — интенсивности отказов основной и резервной машин соответственно, μ_1 и μ_2 — интенсивности восстановления. Для составления марковской цепи должна быть задана *дисциплина восстановления*. Пусть в данном случае — это восстановление одним ремонтником сразу после отказа любой машины, причем если отказали обе машины, то сначала ремонтируется резервная машина, а затем основная.

Требуется определить стационарный коэффициент готовности системы.

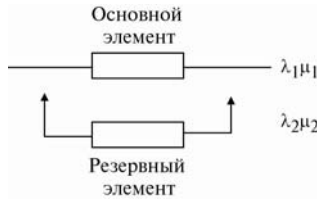


Рис. 2.8

Решение. Введем подходящий для данной задачи набор состояний $\{x_i\}$ (если таких наборов может быть несколько, выберем наиболее экономичный):

- x_0 — оба элемента системы (машины) работоспособны;
- x_1 — основной элемент отказал, резервный — работоспособен;
- x_2 — резервный элемент отказал, основной работоспособен;
- x_3 — оба элемента отказали.

Цепь Маркова для данных состояний приведена на рис. 2.9. Вероятности состояний $p(x_i)$ представим как функцию времени: $p(x_i) = p_i(t)$. Для стационарного процесса (при $t \rightarrow \infty$) p_i не зависит от времени.

Составим дифференциальные уравнения Чепмена—Колмогорова. В левой части каждого уравнения помещаем производную dp_i/dt , а в правой — разность сумм $\sum_j \lambda_{ji} p_j - \sum_k \lambda_{ik} p_i$, где p_i или p_j — вероят-

ность состояния вершины x_i или x_j , из которой выходит стрелка, а λ входит в первую сумму со знаком «+», если стрелка входит в вершину x_i , и во вторую сумму со знаком «-», если выходит из вершины x_i . Уравнения для изображенной цепи (для нестационарного режима) принимают вид:

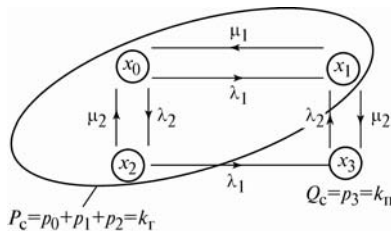


Рис. 2.9

$$\begin{aligned}
dp_0(t)/dt &= \mu_1 p_1(t) + \mu_2 p_2(t) - \lambda_1 p_0(t) - \lambda_2 p_0(t); \\
dp_1(t)/dt &= \lambda_1 p_0(t) + \mu_2 p_3(t) - \mu_1 p_1(t) - \lambda_2 p_1(t); \\
dp_2(t)/dt &= \lambda_2 p_0(t) - \mu_2 p_2(t) - \lambda_1 p_2(t); \\
dp_3(t)/dt &= \lambda_2 p_1(t) + \lambda_1 p_2(t) - \mu_2 p_3(t); \\
\sum_i p_i(t) &= 1 \text{ (уравнение нормировки)}.
\end{aligned}$$

Для стационарного режима (при $t \rightarrow \infty$) $p_i(t) = p_i = \text{const}$ и $dp_i/dt = 0$, тогда

$$\begin{aligned}
0 &= \mu_1 p_1 + \mu_2 p_2 - \lambda_1 p_0 - \lambda_2 p_0; \\
0 &= \lambda_1 p_0 + \mu_2 p_3 - \mu_1 p_1 - \lambda_2 p_1; \\
0 &= \lambda_2 p_0 - \mu_2 p_2 - \lambda_1 p_2; \\
0 &= \lambda_2 p_1 + \lambda_1 p_2 - \mu_2 p_3; \\
\sum_i p_i &= 1.
\end{aligned}$$

Отметим, что уравнений пять, а переменных четыре. Поэтому одно из уравнений (любое, кроме уравнения нормировки) можно исключить, поскольку оно линейно зависит от остальных.

Решая эту алгебраическую систему уравнений, найдем все p_i . Стационарный коэффициент готовности есть вероятность P_c работоспособности ВС в любой момент времени, т.е.

$$k_T = P_c = p_0 + p_1 + p_2,$$

а коэффициент простоя —

$$k_{\Pi} = p_3 = Q_c = 1 - P_c.$$

Пример 2.2. Рассматривается та же ВС, но при $\lambda_1 = \lambda_2$, $\mu_1 = \mu_2$ и с другой дисциплиной обслуживания: восстановление выполняется двумя ремонтниками, по одному на основной и резервный элемент.

Тогда для решения задачи марковскую цепь можно упростить, введя другое определение состояний:

- x_0 — обе машины работоспособны;
- x_1 — отказала одна (любая) машина;
- x_2 — отказали обе машины.

Цепь Маркова для этого случая приведена на рис. 2.10. Такой характер цепи (с последовательно расположенными вершинами и двусторонними стрелками между ними) описывает так называемый *процесс гибели и размножения*, при котором обычно существенно упрощаются расчеты.

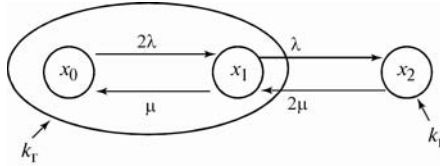


Рис. 2.10

Составив уравнения Чепмена—Колмогорова и решив их, можно убедиться, что коэффициент готовности будет больше, чем в предыдущей задаче.

2.3. Аппарат булевой алгебры. Логические модели надежности

При построении булевой модели надежности каждому элементу системы ставится в соответствие логическая переменная x_i :

$$x_i = \begin{cases} 1, & \text{если элемент работоспособен;} \\ 0 & \text{— в противном случае.} \end{cases}$$

Затем формулируются логические условия работоспособности системы, выражаемые в виде логической функции от этих переменных:

$$F(\hat{x}) \equiv F(x_1, \dots, x_n) = \begin{cases} 1, & \text{если система работоспособна;} \\ 0 & \text{— в противном случае.} \end{cases}$$

Пример 2.3. Пусть невозстанавливаемая система работоспособна, если работоспособны элементы x_1 и x_2 или x_1 и x_3 . Тогда

$$F(x) = x_1 x_2 \vee x_1 x_3 = x_1 (x_2 \vee x_3).$$

Вероятность работоспособности системы определяется как

$$P_c = P[F(x) = 1].$$

Функцию $F(x)$ иногда удобно представить в *арифметизированной форме* (алгебра Порецкого): если $A, B \in \{0, 1\}$, то

$$A = 1 - \bar{A}$$

$A \& B = A \cdot B$ (при этом умножение обладает свойством *идемпотентности*, т.е. $AAB = AB$),

$$A \vee B = A + B - AB,$$

$$A \oplus B = A + B - 2 \cdot AB.$$

Функция $F(x)$ называется *структурной функцией системы* и позволяет рассчитывать характеристики надежности системы по соответствующим характеристикам элементов. В приведенном примере функция $F(x)$ имеет два термина:

$A = x_1 x_2$ и $B = x_1 x_3$, поэтому

$$F(x) = A \vee B = A + B - AB = x_1 x_2 + x_1 x_3 - x_1 x_2 x_3.$$

Подставив вместо x_i вероятности безотказной работы элементов p_i , получим вероятность безотказной работы системы:

$$P = p_1 p_2 + p_1 p_3 - p_1 p_2 p_3.$$

Вместо арифметизации логической функции можно ортогонализировать $F(x)$, т.е. преобразовать ее в такую форму, при которой логическое произведение любых двух термов равно нулю. В данном случае (в примере)

$$F(x) = x_1 x_2 \vee x_1 x_3 = x_1 x_2 \vee x_1 \bar{x}_2 x_3.$$

После этого вместо x_i и \bar{x}_i подставляются соответственно p_i и $q_i = 1 - p_i$.

Тогда

$$P = p_1 p_2 + p_1 (1 - p_2) p_3,$$

что совпадает с полученным ранее результатом.

Для более сложных логических функций существуют различные процедуры ортогонализации [7], например приведение $F(x)$ к совершенной дизъюнктивной нормальной форме, где все термы взаимно ортогональны, метод Фратта—Монтанари и др.

Некоторые полезные формулы булевой алгебры, используемые при преобразованиях структурных функций:

$x_i \vee x_j = \overline{x_i \bar{x}_j}$, $x_i x_j = \overline{\bar{x}_i \vee \bar{x}_j}$ — формулы де-Моргана

$$x_i \vee f(x) = x_i \vee \bar{x}_i f(x).$$

$$x_i f(x_1, \dots, x_i, \dots, x_n) = x_i f(x_1, \dots, 1, \dots, x_n).$$

$$\bar{x}_i f(x_1, \dots, x_i, \dots, x_n) = \bar{x}_i f(x_1, \dots, 0, \dots, x_n).$$

$$x_i \vee f(x_1, \dots, x_i, \dots, x_n) = x_i \vee f(x_1, \dots, 0, \dots, x_n).$$

$$\bar{x}_i \vee f(x_1, \dots, x_i, \dots, x_n) = x_i \vee f(x_1, \dots, 1, \dots, x_n).$$

$f(x_1, \dots, x_i, \dots, x_n) = x_i f(x_1, \dots, 1, \dots, x_n) \vee \bar{x}_i f(x_1, \dots, 0, \dots, x_n)$ — формула разложения Шеннона.

$xf(x) \vee \bar{x} \varphi(x) = xf(x) \vee \bar{x} \varphi(x) \vee f(x) \varphi(x)$ — формула Порецкого.

Приведем примеры использования вышеперечисленных формул:

$$x_1 \cdot \overline{x_1 x_2 x_3} = x_1 \cdot \overline{1 \cdot x_2 x_3} = x_1 \cdot \overline{x_2 x_3} ;$$

$$\overline{x_1} \cdot \overline{x_1 x_2 x_3} = \overline{x_1} \cdot \overline{0 \cdot x_2 x_3} = \overline{x_1} ;$$

$$x_1 \vee \overline{x_1 x_2 x_3} = x_1 \vee \overline{1} = 1 ;$$

$$x_1 \cdot \overline{x_2 x_3} = x_2 (x_1 \cdot \overline{1 \cdot x_3}) \vee \overline{x_2} (x_1 \cdot \overline{0 \cdot x_3}) = x_1 x_2 \overline{x_3} \vee x_1 \overline{x_2} .$$

2.4. Аппарат теории графов. Структурно-графовые модели надежности

Графовое представление используется в различных моделях надежности, например при построении марковской модели, при логическом описании условий безотказности с последующим применением булевой модели и в других случаях. Некоторые модели изначально имеют сетевую структуру. К ним относятся модели живучести и отказоустойчивости многопроцессорных систем и вычислительных сетей. Если вероятностные характеристики системы получить не удастся, то возникает ситуация непредсказуемости отказов. В этом случае применяются структурно-сетевые модели надежности. Основной характеристикой такой системы является связность ее элементов. Система представляется в виде структурного графа, вершины которого соответствуют процессорам или вычислительным машинам, а дуги — линиям передачи данных.

Для количественной оценки уровня надежности используется понятие k -связности вершин графа. Вершины v_i и v_j являются k -связными, если удаление любого подмножества, содержащего не более $k - 1$ других вершин вместе с инцидентными дугами, не нарушает связности. Иногда рассматривают дуговую связность, для которой связность выделенной пары вершин не нарушается при удалении $k - 1$ дуг. Если вершины k -связны по вершинам, то они всегда k -связны и по дугам (но не наоборот). Поэтому будем рассматривать только вершинную k -связность. Чем больше значение k , тем выше надежность.

Путем в графе между вершинами v_i и v_j называется последовательность дуг, соединяющая эти вершины. Для сетей ЭВМ, в большинстве случаев, все дуги и пути графовой модели имеют двустороннюю ориентацию, поэтому граф предполагается неориентированным. Элементарным называется путь, проходящий через содержащиеся в нем вершины только по одному разу. В теории надежности элементарный путь называют также минимальным путем.

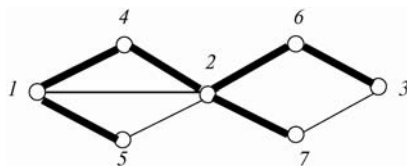


Рис. 2.11

Теорема 2.1. Необходимым и достаточным условием k -связности вершин v_i и v_j является наличие k непересекающихся элементарных путей между этими вершинами.

В зависимости от назначения и функций сети различают три формы связности: парную, корневую и сетевую.

1) *Парная связность двух выделенных вершин* имеет место, если между ними существует хотя бы один неотказавший путь. В общем случае, для определения k нужно найти минимальное число непересекающихся путей, соединяющих выделенные вершины, которое и определяет k -связность.

В графе, изображенном на рис. 2.11, вершины 1 и 2 3-связны, вершины 2 и 3 2-связны, вершины 1 и 3 1-связны.

2) *Корневая связность выделенной вершины с сетью* обеспечивается при наличии хотя бы одного остовного дерева с корнем в данной вершине.

На рис. 2.11 вершина 1 связана с сетью, так как существует остовное дерево (обозначено жирными линиями) 1-4-2-6-3, 1-5, 2-7. Это дерево — не единственное.

Этот показатель используется, например, при анализе и обеспечении надежности связи центральной машины с рабочими станциями пользователей.

3) *Сетевая связность* — наличие парной связности каждой вершины с любой другой.

Сетевая k -связность является наиболее универсальным показателем. Существует несколько методов ее определения. Рассмотрим метод, основанный на алгоритме Клейтмана.

Алгоритм проверки сетевой k -связности:

- 1) Задаемся некоторым значением k , начиная с $k = 1$.
- 2) Выбираем любую вершину графа (обозначим ее v_1) и проверяем ее k -связность с сетью по наличию k непересекающихся элементарных

путей от v_1 до любой другой вершины графа. При выполнении этого условия вершина v_1 удаляется вместе с инцидентными ей дугами.

3) В оставшейся части графа выбирается любая другая вершина v_2 и проверяется наличие $k - 1$ элементарных непересекающихся путей между v_2 и оставшимися вершинами, после чего v_2 удаляется вместе с инцидентными дугами.

.....
 $k + 1$) Выбирается вершина v_k и проверяется ее 1-связность с остальными вершинами оставшейся части графа (по наличию хотя бы одного пути).

Если на всех шагах условия связности выполняются, то граф является k -связным.

Увеличивая значение k ($k := k+1$) и используя данный алгоритм, можно проверить, является ли граф $(k+1)$ -связным и определить максимальный коэффициент связности сети.

Пример 2.4. Рассмотрим графовое представление сети, изображенное на рис. 2.12, а.

Задаемся $k = 1$. Очевидно, что сеть является 1-связной, так как между любыми вершинами существует элементарный путь. Поэтому в данном случае можно не применять алгоритм.

Задаемся $k = 2$. Здесь также достаточно очевидно наличие двух непересекающихся путей между любой парой вершин.

Задаемся $k = 3$. Применим алгоритм Клейтмана. Выбираем (произвольно) вершину v_1 . Поскольку между ней и остальными вершинами существуют по 3 непересекающиеся пути, то условие шага 2 выполняется. После удаления v_1 и дуг получаем сокращенный граф (рис. 2.12, б). Убеждаемся, что условие шага 3 удовлетворяется (между выбранной вершиной v_2 и остальными вершинами существует по 2 непересекающихся пути). После удаления v_2 получаем граф, изображенный на рис. 2.12, в. В нем условие $(k+1)$ -го шага (в данном случае

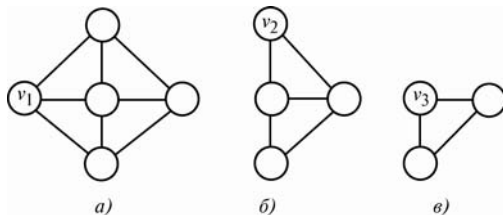


Рис. 2.12

$k + 1 = 4$), т.е. условие проверки 1-связности v_3 с сетью), также выполняется.

Задаемся $k = 4$ и убеждаемся, что не существует четырех непересекающихся элементарных путей, связывающих v_1 с каждой из оставшихся вершин графа. Значит максимальный коэффициент связности сети равен $k = 3$.

Если известны статистические показатели надежности, то можно рассчитать отказоустойчивость сети в вероятностной форме:

- 1) вероятность парной связности,
- 2) вероятность корневой связности выделенной вершины,
- 3) вероятность сетевой связности.

2.5. Аппарат статистического моделирования. Имитационные модели надежности

При расчете надежности ВС обычно предполагается, что ВБР и ВВ элементов имеют показательное (экспоненциальное) распределение. Отдельные показатели надежности удастся рассчитать и при произвольном распределении. Но в общем случае приходится прибегать к имитационному моделированию. Имитационные модели используются также для сложных систем с показательными ВБР и ВВ, если расчеты оказываются слишком трудоемкими.

Моделирование начинается с генерации случайных величин X с заданной функцией распределения. Для этого используется *метод статистического моделирования* (Монте-Карло). Затем рассчитываются характеристики, связанные с этими случайными величинами.

Один из распространенных способов генерации носит название *метода обратных функций*. Его сущность состоит в следующем. Пусть случайная переменная X имеет непрерывную и строго монотонную функцию распределения $F_X(x)$ (рис. 2.13).

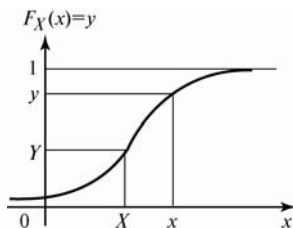


Рис. 2.13

Обозначим эту функцию через y : $y = F(x)$ (без индекса X) и будем рассматривать ее не как распределение, а как отображение x в y . Тогда случайная переменная X имеет образ $Y = F(X)$, Y также является случайной величиной. Обратную функцию (прообраз) обозначим G : $x = G(y)$ и $X = G(Y)$. Тогда

$$Y = F[G(Y)].$$

Отметим, что $F(x)$ совпадает с функцией распределения $F_X(x)$, но $G(y)$ не является функцией распределения $F_Y(y)$ случайной величины Y . При моделировании $F_Y(y)$ может быть задана независимо от $F_X(x)$.

Решим следующую задачу: пусть Y имеет заданное нами распределение $F_Y(y)$, требуется определить функцию распределения $F_{G(Y)}(x)$ случайной переменной $G(Y)$, являющейся функцией случайной величины Y .

Теорема 2.2. Если величина Y распределена равномерно на интервале $[0,1]$, т.е. имеет плотность распределения, изображенную на рис. 2.14, то $G(Y)$ имеет распределение, определяемое кривой $F(x)$.

Действительно, распределение

$$F_{G(Y)}(x) = P[G(Y) < x] = P[F(G(Y)) < F(x)] = P[Y < F(x)],$$

где Y — случайная переменная, а x и $F(x)$ — независимые переменные.

Но Y , по условию, имеет равномерную плотность распределения в интервале $[0,1]$, поэтому вероятность

$$P[Y < F(x)] = \int_0^{F(x)} 1 \cdot dy = F(x),$$

а график этой функции совпадает с $F_X(x)$. Значит, сгенерировав значение равномерно распределенной с.в. Y , получим по графику $F(x)$ (рис. 2.13) значение с.в. $X = G(Y)$, имеющей распределение $F_X(x)$.

Моделирование начинается с генерации равномерно распределенной величины Y (с помощью генератора случайных чисел). Затем по заданному нами графику (рис. 2.13) определяется значение функции

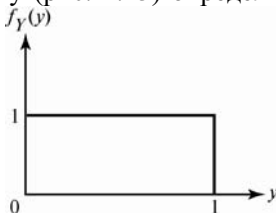


Рис. 2.14

$X = G(Y)$, имеющей (как было показано) распределение $F_X(x)$. Эта процедура носит название *разыгрывания с.в. X*. Полученные реализации случайных чисел используются для расчета характеристик ВС.

Пример 2.5. Вычислительная система, содержащая две ЭВМ, принимает из сети и обрабатывает информационные пакеты. Если одна из машин занята обработкой, пакет поступает на другую машину. При занятости обеих машин пакеты вводятся в буферный накопитель и выстраиваются в очередь на обработку. При отказе одной из машин все пакеты обрабатываются оставшейся работоспособной машиной, но с вдвое меньшей производительностью, а отказавшая машина ремонтируется. Тем самым обеспечивается функциональное резервирование, повышающее *отказоустойчивость* системы. Если отказали обе машины, производительность становится нулевой, обе машины ремонтируются, а очередь пакетов возрастает.

Пусть ВБР и ВВ каждой машины имеют показательные распределения:

$$Q(t) = F_\theta(t) = 1 - e^{-\lambda t} \text{ (вероятность отказа в } [0, t])$$

и

$$R(t) = F_\xi(t) = 1 - e^{-\mu t} \text{ (вероятность восстановления в } [0, t]).$$

При этом роль случайной величины X выполняют случайные переменные θ (случайная наработка на отказ) или ξ (случайное время восстановления), а роль Y — функции $F(\theta)$ и $F(\xi)$, совпадающие с функциями распределения.

Пусть задан интервал $[0, \tau]$, в течение которого система должна работать (принимать и обрабатывать информацию). Требуется определить среднюю производительность системы за этот интервал.

Процедура моделирования. Функциональная зависимость между Y и θ имеет вид:

$$Y = F(\theta) = 1 - e^{-\lambda \theta},$$

а обратная функция

$$\theta = G(Y) = (-1/\lambda) \cdot \ln(1 - Y).$$

Аналогично, обратная функция для времени восстановления

$$\xi = G(Y) = (-1/\mu) \cdot \ln(1 - Y).$$

Обозначим случайные значения θ и ξ для первой машины через θ' и ξ' , и для второй — θ'' и ξ'' .

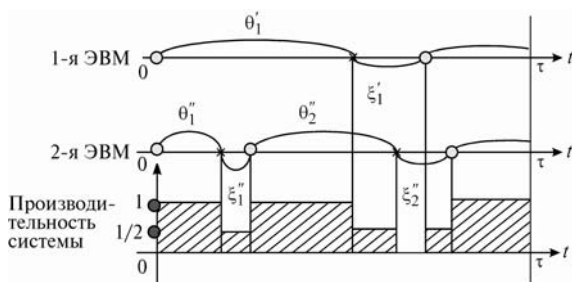


Рис. 2.15

Сначала рассматриваются состояния первой машины (рис. 2.15) и фиксируются моменты отказов (отмечены крестиками) и восстановлений (отмечены кружками).

Для получения первого момента отказа θ'_1 разыгрывается случайная величина θ' , имеющая экспоненциальное распределение. Для этого генерируется равномерно распределенная величина Y и находится $\theta' = G(Y)$. Если $\theta' < \tau$, то фиксируется момент отказа θ'_1 , если же $\theta' > \tau$, то в интервале $[0, \tau]$ отказа не наблюдается и θ'_1 не отмечается. После фиксации θ'_1 начало временной координаты переносится в точку отказа, при этом интервал $[0, \tau]$ сокращается. Затем разыгрывается случайная величина ξ' и находится первый момент восстановления первой машины, отстоящий от нового начала координат на расстоянии ξ'_1 . После этого для первой машины находятся интервалы θ'_2 и ξ'_2 и т.д.

Аналогичная процедура выполняется для второй машины, в результате чего определяются моменты времени θ'' и ξ'' .

По окончании разыгрывания времен θ и ξ определяются значения производительности системы, зависящие от состояний машин (третья диаграмма на рис. 2.15). Если производительность системы при работоспособности обеих машин обозначить через 1, то при отказе одной из машин она составляет 1/2, а при отказе обеих машин становится равной 0. Усреднение в интервале $[0, \tau]$ определяет *среднюю производительность системы в одном эксперименте*.

После проведения серии подобных имитационных экспериментов, определяется *средняя производительность для всей серии*. Для определения необходимого числа таких экспериментов используются статистические оценки, применяемые в методе Монте-Карло.

3. РАСЧЕТ И ОЦЕНКА ПОКАЗАТЕЛЕЙ НАДЕЖНОСТИ НЕРЕЗЕРВИРОВАННЫХ НЕВОССТАНАВЛИВАЕМЫХ ОБЪЕКТОВ

3.1. Объекты с произвольной функцией распределения времени безотказной работы

3.1.1. Вероятность безотказной работы в заданном интервале

Предполагается известной ф.р. ВБР $F_{\theta}(t)$. Определим $P(0, t) \equiv P(t)$ и $P(t, t + \Delta t)$ (рис. 3.1).

Вероятности безотказной работы объекта в интервале $[0, t]$:

$$P(0) = 1 \text{ (по определению);}$$

$$P(t) = P[\theta \geq t] = 1 - P[\theta < t] = 1 - F_{\theta}(t).$$

Вероятность отказа

$$Q(t) = 1 - P(t) = F_{\theta}(t).$$

Условная вероятность безотказной работы объекта в интервале $[t, t + \Delta t]$, если до момента t отказа не было,

$$\begin{aligned} P(t, t + \Delta t) &= \\ &= P[\theta \geq t + \Delta t / \theta \geq t] = \frac{P[\theta \geq t + \Delta t, \theta \geq t]}{P[\theta \geq t]} = \\ &= \frac{P[\theta \geq t + \Delta t] \cdot P[\theta \geq t / \theta \geq t + \Delta t]}{P[\theta \geq t]}. \end{aligned}$$

Но $P[\theta \geq t / \theta \geq t + \Delta t] = 1$, поэтому

$$P(t, t + \Delta t) = \frac{P[\theta \geq t + \Delta t]}{P[\theta \geq t]} = \frac{P(t + \Delta t)}{P(t)}.$$

Вероятность отказа

$$Q(t, t + \Delta t) = 1 - P(t, t + \Delta t) = 1 - \frac{P(t + \Delta t)}{P(t)}.$$

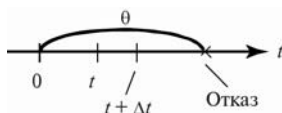


Рис. 3.1

3.1.2. Плотность распределения ВБР и интенсивность отказов

Плотность распределения ВБР определяется следующим образом:

$$f_{\theta}(t) = dF_{\theta}(t)/dt = dQ(t)/dt = d(1 - P(t))/dt = -P(t)/dt.$$

Выразим P и Q через $f_{\theta}(t)$:

$$Q(t) = \int_0^t f_{\theta}(x)dx, \quad P(t) = 1 - \int_0^t f_{\theta}(x)dx = \int_t^{\infty} f_{\theta}(x)dx.$$

Заштрихованные площади на рис. 3.2 равны вероятностям отказа и безотказной работы объекта.

По определению, $\lambda(t)$ — условная плотность распределения ВБР, если до момента t отказа не было. Тогда условная вероятность отказа в интервале dt , если ранее отказа не было, равна

$$Q(t, t + dt) = \lambda(t) dt,$$

а безусловная вероятность отказа в этом интервале

$$f_{\theta}(t) dt = P[t \leq \theta < t + dt] = P[\theta < t + dt, \theta \geq t].$$

Поскольку безусловная вероятность работоспособного состояния в интервале $[0, t]$ составляет

$$P[\theta \geq t] = P(t),$$

то согласно правилу умножения вероятностей

$$f_{\theta}(t)dt = P(t) \cdot P[\theta < t + dt/\theta \geq t] = P(t) \cdot Q(t, t + dt) = P(t) \cdot \lambda(t)dt,$$

откуда

$$\lambda(t) = f_{\theta}(t)/P(t).$$

Выразим P через $\lambda(t)$:

$$\lambda(t) = -\frac{dP(t)/dt}{P(t)} = -\frac{d \ln P(t)}{dt},$$

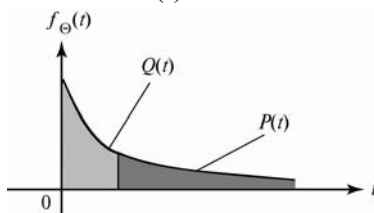


Рис. 3.2

откуда

$$d \ln P(t) = -\lambda(t) dt.$$

Проинтегрировав обе части равенства от 0 до t , находим:

$$\int_0^t d \ln P(x) = -\int_0^t \lambda(x) dx ; \ln P(t) - \ln P(0) = -\int_0^t \lambda(x) dx .$$

Поскольку $\ln P(0) = 0$, то

$$P(t) = e^{-P(t)} = e^{-\int_0^t \lambda(x) dx} .$$

При $\lambda = \text{const}$ имеем

$$P(t) = e^{-\lambda t}, \quad Q(t) = 1 - e^{-\lambda t} .$$

Статистические оценки. Статистические оценки всегда рассчитываются по множеству реализаций. Пусть на испытание поставлены $N(0)$ объектов (в данном случае $N(0) = 6$), рис. 3.3. Крестиками обозначены моменты отказа соответствующих объектов. Через $N(t)$ обозначим число работоспособных объектов в момент времени t , а через $n(0, t)$ и $n(t, t + \Delta t)$ — число изделий, отказавших в соответствующем интервале. Оценки показателей будем снабжать штрихами (например, P' , λ' и т.д.).

Оценки определяются выведенными выше соотношениями для показателей надежности и совпадают с показателями при бесконечном числе изделий. В скобках указаны численные значения оценок для рис. 3.3.

$$P'(t) = N(t) / N(0) = 4/6,$$

$$Q'(t) = 1 - P(t) = n(0, t) / N(0) = 2/6,$$

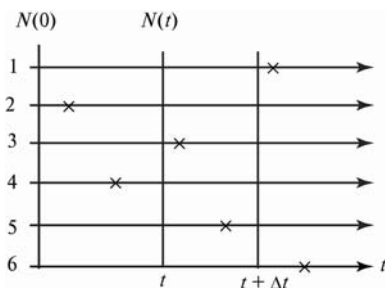


Рис. 3.3

$$P'(t, t + \Delta t) = \frac{N(t + \Delta t) / N(0)}{N(t) / N(0)} = N(t + \Delta t) / N(t) = 2/4,$$

$$Q'(t, t + \Delta t) = \frac{N(t) - N(t + \Delta t)}{N(t)} = n(t, t + \Delta t) / N(t) = 2/4,$$

при этом Δt может быть выбрано произвольно.

Однако для оценок $f'_0(t)$ и $\lambda'(t)$ величина Δt должна быть выбрана достаточно малой по сравнению со средней наработкой до отказа, а число изделий $N(0)$ достаточно большим, так чтобы в интервале Δt было зафиксировано не менее 10 отказов различных изделий. Тогда

$$f'_0(t)\Delta t = Q'(t + \Delta t) - Q'(t) = \frac{n(0, t + \Delta t) - n(0, t)}{N(0)} = \frac{n(t, t + \Delta t)}{N(0)},$$

или

$$f'_0(t) = \frac{n(t, t + \Delta t)}{N(0)\Delta t} = \frac{2}{6\Delta t},$$

$$\lambda'(t)\Delta t = Q'(t, t + \Delta t) = \frac{n(t, t + \Delta t)}{N(t)},$$

или

$$\lambda'(t) = \frac{n(t, t + \Delta t)}{N(t)\Delta t} = \frac{2}{4\Delta t}.$$

Вывод: $f'_0(t)$ отличается от $\lambda(t)$ только тем, что в первом случае выполняется деление на $N(0)$, а во втором случае — на $N(t)$.

3.1.3. Средняя наработка до отказа

Определим среднюю наработку до отказа:

$$\begin{aligned} T = M[\theta] &= \int_0^{\infty} t \cdot f_0(t) dt = \int_0^{\infty} t \cdot dQ(t) = - \int_0^{\infty} t \cdot dP(t) = \\ &= -tP(t) \Big|_0^{\infty} + \int_0^{\infty} P(t) dt = \int_0^{\infty} P(t) dt, \end{aligned}$$

так как первое слагаемое равно 0 в силу того, что при $t \rightarrow \infty$ $P(t)$ убывает быстрее, чем возрастает t , поэтому произведение стремится к 0. Из этого соотношения видно, что T есть площадь под кривой $P(t)$ (рис. 3.4).

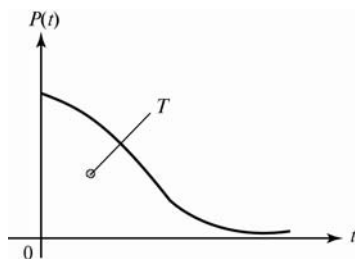


Рис. 3.4

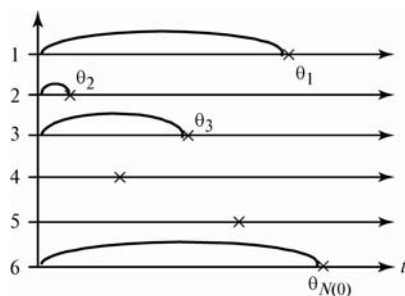


Рис. 3.5

Статистические оценки. Возможно использование трех видов оценок, основанных на приведенных соотношениях:

а) Оценка, основанная на выражении $T = M[\theta]$ (рис. 3.5):

$$T' = \frac{\sum_{i=1}^{N(0)} Q_i}{N(0)}. \quad (3.1)$$

Достоинство — высокая точность. Недостаток — необходимость наблюдения момента отказа каждого изделия, поставленного на испытание.

б) Оценка, основанная на выражении $T = \int_0^{\infty} t dQ(t)$:

$$T' = \sum_{i=1}^m t_i \Delta Q(t_i), \quad (3.2)$$

где $\Delta Q(t_i)$ — доля изделий, отказавших в интервале $\Delta t_i = t_i - t_{i-1}$. Величина $\Delta Q(t_i)$ задается, обычно полагается $\Delta Q(t_i) = \text{const} = \Delta Q$. Значе-

ния Δt_i и t_i заранее неизвестны и определяются по результатам испытаний.

ΔQ определяет некоторую группу из $n(t_{i-1}, t_i) = n$ изделий, состав которой заранее неизвестен, а величина $n = \text{const}$ задается до испытаний и определяет точность оценки. Число таких групп $m = N(0)/n = 1/\Delta Q$. Чем больше m , тем выше точность оценки, но при этом усложняется процедура эксперимента, так как при испытании требуется зафиксировать отказ последнего (априорно неизвестного) элемента каждой группы. При указанных условиях (3.2) принимает вид:

$$T' = \Delta Q \sum_{i=1}^m t_i = \frac{1}{m} \sum_{i=1}^m t_i. \quad (3.3)$$

Наибольшая точность оценки имеет место при $n = 1$ и $m = N(0)$, тогда $t_i = \theta_i$ и формула (3.3) совпадает с (3.1) для оценки, приведенной в п. а.

Порядок проведения эксперимента:

- Задается число групп m , при этом в каждой группе будет $n = N(0)/m$ изделий.
- Наблюдается момент t_1 отказа n -го изделия; наблюдается момент t_2 отказа $2n$ -го изделия; наблюдается момент t_3 отказа $3n$ -го изделия и т.д. В результате становится известным состав 1, 2, ..., m -й групп и значения t_1, t_2, \dots, t_m .
- По формуле (3.3) рассчитывается значение оценки T' .

На рис. 3.6 приведен график $F_0(t) = Q(t)$. Ординаты выше этого графика есть $P(t)$, площадь выше графика есть точное значение средней наработки до отказа T , а заштрихованная площадь определяет верхнюю границу оценки, полученную данным методом.

Недостатком использования данной оценки является необходимость фиксации неизвестных моментов отказов каждого n -го изделия группы.

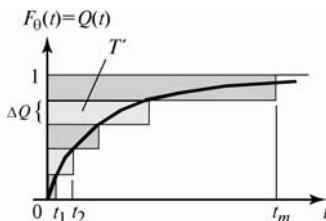


Рис. 3.6

в) Оценка, основанная на выражении $T = \int_0^{\infty} P(t)dt$:

$$T' = \sum_{i=1}^m P'(t_{i-1})\Delta t_i ,$$

где $\Delta t_i = t_i - t_{i-1}$. Обычно $\Delta t_i = \Delta t = \text{const}$. Тогда

$$T' = \Delta t \sum_{i=1}^m \frac{N(t_{i-1})}{N(0)} = \frac{\Delta t}{N(0)} \sum_{i=1}^m T(t_{i-1}) . \quad (3.4)$$

Чем меньше Δt , тем выше точность оценки. Достоинство — простота эксперимента: интервалы Δt и, следовательно, моменты времени наблюдений t_i фиксированы и заранее известны. Например, можно проводить наблюдения в конце каждого месяца. Поэтому данная оценка используется наиболее часто. Недостаток — заранее неизвестно число m отсчетов (моментов наблюдений).

Порядок проведения эксперимента:

- Задается интервал Δt .
- В конце каждого интервала Δt_i наблюдается число оставшихся исправных изделий $N(t_i)$ (при $t_{i-1} = 0$ $N(t_{i-1}) = N(0)$, через Δt , т.е. в момент t_1 , число исправных изделий составляет $N(t_1)$ и т.д.).
- По формуле (3.4) рассчитывается оценка T' .

На рис. 3.7 приведен график, подобный изображенному на рис. 3.6. Площадь каждого заштрихованного прямоугольника равна $\Delta t \frac{N(t_{i-1})}{N(0)}$, а общая площадь определяет верхнюю границу оценки T' ,

полученную данным методом.

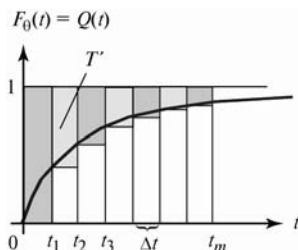


Рис. 3.7

3.1.4. γ -процентная наработка T_γ объекта

T_γ есть наработка, в течение которой гарантируется вероятность безотказной работы изделия в вероятностью P_γ (или γ в процентах от $P = 1$).

Рис. 3.8 иллюстрирует определение T_γ по заданному значению P_γ .

Пример 3.1. Завод выпускает в год 1000 изделий, ВБР которых имеет экспоненциальное распределение. В течение года на завод поступает в среднем 84 рекламации. Требуется определить гарантийный срок на изделия, при котором доля рекламаций не превышала бы 10 % (допустимые потери завода, носящие название «риск изготовителя»).

Решение. Допустимые потери в вероятностном выражении есть P_γ , а гарантийный срок — T_γ . Определим сначала параметр распределения λ . В практике расчета надежности интенсивность отказов λ выражается в 1/ч. В году 8760 ч. При круглосуточной эксплуатации изделий на предприятиях заказчиков

$$P(8760) = e^{-\lambda \cdot 8760} = N(8760) / N(0) = (1000 - 84) / 1000 = 0,916.$$

Отсюда

$$\lambda = -\ln 0,916 / 8760 = 10^{-5} \text{ 1/ч.}$$

По условию задачи

$$P_\gamma = 0,9 \text{ (90 \%)} = e^{-\lambda T_\gamma}.$$

Отсюда $T_\gamma = -\ln 0,9 / 10^{-5} = 10\,536 \approx 1,2$ года.

Ответ: $T_\gamma = 1,2$ года.

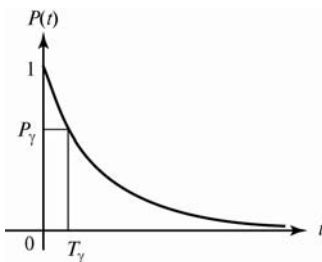


Рис. 3.8

3.2. Объекты с показательной функцией распределения времени безотказной работы («экспоненциальные» объекты)

При показательной ф.р. ВБР:

$$\begin{aligned}\lambda &= \text{const}, \\ F_{\theta}(t) &= 1 - e^{-\lambda t}, \quad f_{\theta}(t) = \lambda e^{-\lambda t}, \\ P(t) &= e^{-\lambda t}, \quad Q(t) = 1 - e^{-\lambda t}, \\ P(t, t + \Delta t) &= P(t + \Delta t)/P(t) = e^{-\lambda(t + \Delta t)}/e^{-\lambda t} = e^{-\lambda \Delta t}, \\ T = M[\theta] &= \int_0^{\infty} e^{-\lambda t} dt = 1/\lambda.\end{aligned}$$

Признаки экспоненциальности распределения ВБР объектов:

1) Дисперсия экспоненциального распределения с плотностью $f_{\theta}(t) = 1 - e^{-\lambda t}$ равна

$$\sigma^2[\theta] = \int_0^{\infty} (t - T)^2 f_{\theta}(t) dt = 1/\lambda^2,$$

а среднеквадратичное отклонение

$$\sigma[\theta] = 1/\lambda = T = M[\theta].$$

Таким образом, если $\sigma[\theta] \approx T$, то распределение можно считать экспоненциальным.

2) Другой признак вытекает из соотношения

$$P(T) = e^{-\lambda T} = e^{-1} = 0,368.$$

Таким образом, если $P(T) \approx 0,368$, то объект «экспоненциальный».

Пример 3.2. В процессе ускоренных статистических испытаний установлено, что $T' = 10000$ ч, $P'(10000) = 0,365$. Поскольку 0,365 мало отличается от 0,368 (на 0,08 %), то объект также можно считать «экспоненциальным».

Для экспоненциального распределения разложение в ряд величины $e^{-\lambda t}$ при $\lambda t \leq 0,1$ ($t \leq 0,1T$) дает

$$e^{-\lambda t} = 1 - \lambda t + \lambda^2 \cdot t^2 / 2! - \lambda^3 t^3 / 3! + \dots \approx 1 - \lambda t, \quad (3.5)$$

при этом верхняя оценка ошибки, возникающей при отбрасывании третьего и последующих членов ряда равна

$$\frac{\lambda^2 t^2 / 2!}{e^{-\lambda t}} \approx 0,005 \text{ (0,5 \%)}.$$

Если положить $t = 0,1T$, то

$$P(0,1T) = e^{-0,1T\lambda} = e^{-0,1} = 0,9048.$$

Таким образом, формулой (3.5) можно пользоваться, если:

а) $t \leq 0,1T$, б) $P(t) > 0,9$.

Пример 3.3. $P_\gamma = 0,99$, $\lambda = 10^{-5}$ 1/ч. Найти гамма-процентную наработку T_γ .

Решение. $P_\gamma = e^{-\lambda T_\gamma}$. Поскольку $P_\gamma > 0,9$ (по условию), то $P_\gamma \approx 1 - \lambda T_\gamma = 0,99$, откуда

$$T_\gamma = (1 - P_\gamma) / \lambda = 0,01 / 10^{-5} = 10^3 \text{ ч} \approx 42 \text{ дня}.$$

4. ПОКАЗАТЕЛИ НАДЕЖНОСТИ ВОССТАНАВЛИВАЕМЫХ ОБЪЕКТОВ

4.1. Показатели процесса отказов и восстановлений

Обычно время восстановления много меньше времени от момента восстановления до отказа. В этом случае временем восстановления (ВВ) можно пренебречь, считая его нулевым (мгновенное восстановление). В этом случае рассматривается *точечный процесс (поток) отказов/восстановлений* (о/в) (рис. 4.1). В противном случае (при конечном времени восстановления) процесс называется *альтернирующим* и на рисунке изображается в виде кружочков (моменты восстановления) и крестиков (моменты отказов) (рис. 4.2), либо в виде изменения состояний случайного процесса $X(t)$ с исходами x_0 (работоспособное состояние) и x_1 (состояние отказа) (рис. 4.3).

На рисунках через θ обозначено время работоспособного состояния, через ξ — время нахождения в состоянии отказа и через $\eta = \theta + \xi$ — случайное время безотказной работы (ВБР) плюс время восстановления (ВВ).



Рис. 4.1

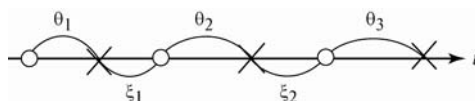


Рис. 4.2

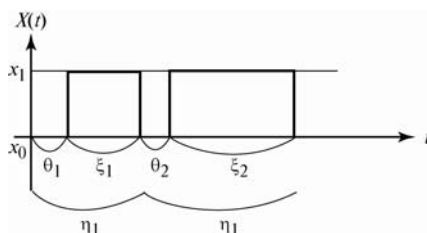


Рис. 4.3

При анализе показателей предполагаются известными функции распределения $F_{\theta}(t)$, $F_{\xi}(t)$, $F_{\eta}(t) = F_{\theta} + \xi(t)$. Рассмотрим применяемые показатели.

1) Вероятность восстановления и невосстановления в альтернирующем потоке:

$R(t) = F_{\xi}(t)$ — вероятность восстановления в интервале от 0 до t (аналог $Q(t)$ для невосстанавливаемых объектов),

$S(t) = 1 - F_{\xi}(t)$ — вероятность невосстановления (аналог $P(t)$).

2) Плотность распределения ВВ и интенсивность восстановления:

$$f_{\xi}(t) = dF_{\xi}(t)/dt \text{ (аналог } f_{\theta}(t)),$$

$$\mu(t) = f_{\xi}(t)/S(t) = \frac{dR(t)/dt}{S(t)} = - \frac{dS(t)/dt}{S(t)} \text{ (аналог } \lambda(t)).$$

3) Среднее время восстановления:

$$T_{\text{в}} = \int_0^{\infty} S(t)dt \text{ (аналог } T).$$

4) Параметр точечного потока ω_0 . Введем следующие обозначения.

- $K(0, t)$ — случайное число отказов в интервале $[0, t]$ (дискретная случайная величина) определена для одного объекта (рис. 4.4).
- $m(0, t) = M[K(0, t)]$ — м.о. числа отказов (усреднение выполняется по множеству объектов, поставленных на испытание), $m(0, t)$ является непрерывной неслучайной функцией, зависящей от t . В теории потоков этот показатель называется *функцией восстановления*.
- Приращение функции восстановления за время dt составляет

$$dm(t) = m(0, t + dt) - m(0, t).$$

Тогда параметр точечного потока определяется как матожидание числа отказов в единицу времени

$$\omega_0(t) = dm(t) / dt.$$

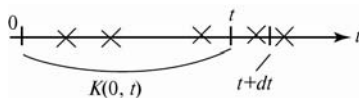


Рис. 4.4

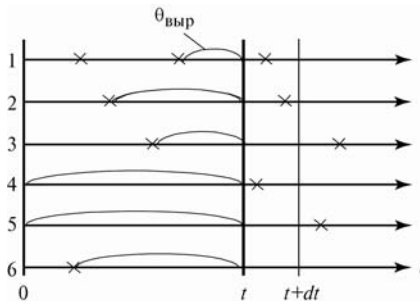


Рис. 4.5

Приведем пример статистической оценки параметра точечного потока. Пусть на испытание поставлено $N(0)$ изделий, которые отказывают и мгновенно восстанавливаются. На рис. 4.5 $N(0) = 6$. Для восстанавливаемых изделий число работоспособных изделий $N(t) = N(0)$. Отметим, что на малом интервале Δt (т.е. dt) изделие может отказать не более одного раза.

Для примера, изображенного на рис. 4.5:

$$\begin{aligned} K_1(0, t) = 2, & \quad K_1(0, t + \Delta t) = 3, & K_4(0, t) = 0, & \quad K_4(0, t + \Delta t) = 1, \\ K_2(0, t) = 1, & \quad K_2(0, t + \Delta t) = 2, & K_5(0, t) = 0, & \quad K_5(0, t + \Delta t) = 0, \\ K_3(0, t) = 1, & \quad K_3(0, t + \Delta t) = 1, & K_6(0, t) = 1, & \quad K_6(0, t + \Delta t) = 1. \end{aligned}$$

Средние значения в соответствующих интервалах

$$m'(0, t) = 5/6, \quad m'(0, t + \Delta t) = 8/6,$$

а среднее число изделий, отказавших в интервале Δt , равно $\Delta m'(t) = 3/6$. Тогда оценка параметра потока отказов

$$\omega'_0(t) = \Delta m'(t) / \Delta t = 3 / (6\Delta t).$$

Замечание. Приведенная формула для оценки ω_0 формально совпадает с формулой оценки интенсивности отказа λ для невосстанавливаемых объектов. Однако физическая интерпретация их различна и, в общем случае, ω_0 и λ различаются. Причина заключается в том, что вероятность отказа в интервале $[t, t + \Delta t]$ зависит от выработанного изделием ресурса $\theta_{\text{выр}}$ (рис. 4.5) к моменту времени t . Для восстанавливаемых изделий, поставленных на испытание, значения $\theta_{\text{выр}}(t)$ в момент t различаются, что видно на рисунке, а для невосстанавливаемых изделий (если они не отказали до момента t) выработанный ресурс одинаковый.

Показатели ω_0 и λ совпадут (при любом распределении ВБР) только при «экзотической» дисциплине восстановления, когда при отказе какого-либо элемента все изделия, поставленные на испытание, заменяются на новые (с невыработанным ресурсом).

Однако, если все элементы изделий имеют экспоненциальное распределение ВБР (и только в этом случае), вероятность отказа в интервале Δt не зависит от выработанного ресурса и

$$\omega_0(t) = \lambda(t) = \lambda = \omega_0 = \text{const}$$

для любого значения t .

5) Средняя наработка на отказ (между отказами) T_0 :

$$T_0 = 1/\omega_0$$

(определена только при $\omega_0 = \text{const}$, т.е. для стационарного потока отказов).

6) Среднее время между отказами в альтернирующем потоке:

$$T_{\text{ов}} = M[\eta] = M[\theta] + M[\xi] = T_0 + T_{\text{в}}.$$

7) Среднее число отказов в единицу времени в альтернирующем потоке:

$$\omega = 1/T_{\text{ов}} = 1/(T_0 + T_{\text{в}}).$$

4.2. Показатели процесса отказов/восстановлений с показательной функцией распределения ВБР и ВВ

Для таких процессов $\lambda = \text{const}$, $\mu = \text{const}$, где $\mu < \infty$ — интенсивность восстановления объекта в альтернирующем потоке и $\mu \rightarrow \infty$ в случае точечного процесса.

$$R(t) = 1 - e^{-\mu t}, \quad S = e^{-\mu t},$$

$$\mu = -\frac{dS(t)/dt}{S(t)},$$

при этом (только для экспоненциальных распределений):

$$T_0 = 1/\lambda, \quad T_{\text{в}} = 1/\mu, \quad T_{\text{ов}} = T_0 + T_{\text{в}} = \frac{\mu + \lambda}{\mu\lambda}, \quad \omega = \frac{\mu\lambda}{\mu + \lambda}.$$

4.3. Комплексные показатели надежности

4.3.1. Коэффициенты готовности и простоя

Коэффициент готовности есть вероятность работоспособного состояния объекта в любой заданный нами момент времени t . Обозначим $X(t) = x_0$ — работоспособное состояние объекта; $X(t) = x_1$ — состояние отказа (период восстановления).

Тогда:

$k_T(t) = P[X(t) = x_0]$ — нестационарный коэффициент готовности;

$k_n(t) = 1 - k_T(t) = P[X(t) = x_1]$ — коэффициент простоя объекта.

Стационарный коэффициент получается в установившемся режиме:

$$k_T = \lim_{t \rightarrow \infty} k_T(t).$$

Теорема 4.1. Для эргодических марковских процессов в стационарном режиме имеет место эргодическое свойство: доля времени пребывания объекта в состоянии x_0 (исправное состояние) равна доле работоспособных объектов (находящихся в процессе испытаний) в произвольно выбранный момент времени, т.е. для коэффициента готовности усреднение по времени и по множеству реализаций приводит к одинаковому результату.

Для одной реализации (рис. 4.6) имеем:
среднее по времени

$$\bar{k}_T = T_0 / (T_0 + T_B),$$

а оценка этой величины

$$\bar{k}'_T = \sum_{i=1}^m \theta_i / \sum_{i=1}^m (\theta_i + \xi_i),$$

где θ_i — случайное время безотказной работы; ξ_i — случайное время восстановления (моменты отказов отмечены крестиками, восстановления — кружочками); m — число отказов (или восстановлений) за большой промежуток времени T .

Для множества реализаций (рис. 4.7) имеем: $N(0) = 6$, $N(t) = 4$ и $k'_T = N(t) / N(0) = 4/6$. Эта оценка похожа на оценку $P(t)$.

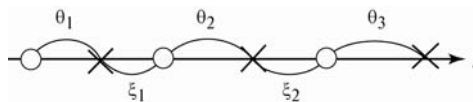


Рис. 4.6

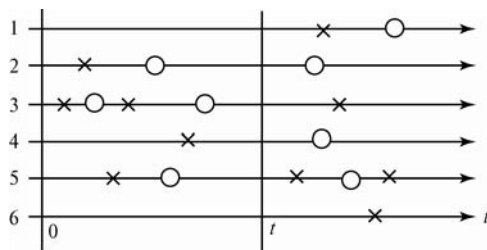


Рис. 4.7

Отметим, что $\lim k'_T(t) = k_T$ при $N(0) \rightarrow \infty$ и не зависит от времени. Эргодическое свойство для коэффициента готовности означает, что $k_T = \bar{k}_T$.

Примечание: формула $\bar{k}_T = T_0 / (T_0 + T_B)$ справедлива для эргодического марковского процесса с любой функцией распределения ВБР и ВВ.

4.3.2. Коэффициент оперативной готовности

Коэффициент оперативной готовности представляет собой совместную вероятность того, что в заданный момент времени t объект окажется в работоспособном состоянии и проработает безотказно еще в течение заданного нами интервала Δt :

$$k_{o.T} = k_T P(t, t + \Delta t),$$

где $P(t, t + \Delta t)$ — условная вероятность безотказной работы восстанавливаемого объекта в интервале Δt , если в момент t он находился в работоспособном состоянии.

Замечание. Вероятность $P(t, t + \Delta t)$ похожа на аналогичный показатель для невосстанавливаемых объектов, но в общем случае не совпадает с ним, так как зависит от выработанного ресурса $\theta_{\text{выр}}$ от момента восстановления до момента t , а в случае невосстанавливаемых объектов этот ресурс одинаков для всех неотказавших изделий, поставленных на испытание, т.е. при расчете $P(t, t + \Delta t)$ ранее предполагалось, что изделие проработало безотказно в интервале от 0 до t .

Таким образом, для восстанавливаемых объектов $P(t, t + \Delta t)$ есть условная вероятность $P_{\text{ост}}(\Delta t)$ того, что остаточное время работоспособности $\theta_{\text{ост}}$ от момента t до момента очередного отказа будет больше, чем интервал Δt (рис. 4.8). Следовательно, $k_{o.T} = k_T P_{\text{ост}}(\Delta t)$.

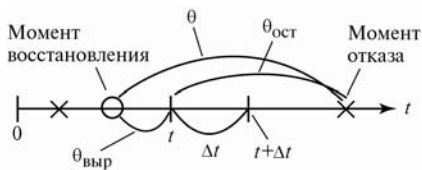


Рис. 4.8

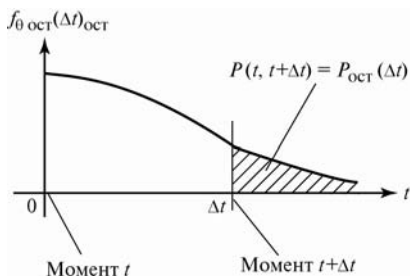


Рис. 4.9

Пусть остаточное время имеет некоторую плотность распределения $f_{\theta_{\text{ост}}}(\Delta t)$, зависящую от $\theta_{\text{выр}}$ (рис. 4.9):

$$P_{\text{ост}}(\Delta t) = P[\theta_{\text{ост}} \geq \Delta t] = \int_{\Delta t}^{\infty} f_{\theta_{\text{ост}}}(x) dx.$$

Поскольку распределение $f_{\theta_{\text{ост}}}(\Delta t)$ зависит как от Δt , так и от выработанного ресурса $\theta_{\text{выр}}$, то оно не совпадает с распределением $f_{\theta}(\Delta t)$ времени θ между моментом восстановления и моментом отказа. Однако для экспоненциальных объектов (и только для них) эти распределения совпадают:

$$f_{\theta_{\text{ост}}}(\Delta t) = f_{\theta}(\Delta t) = \lambda e^{-\lambda \Delta t}.$$

При этом

$$P(t, t + \Delta t) = e^{-\lambda \Delta t}.$$

4.3.3. Коэффициент технического использования

Коэффициент технического использования определяется как

$$k_{\text{ТИ}} = T_0 / (T_0 + T_{\text{в}} + T_{\text{п}}),$$

где $T_{\text{п}}$ — среднее время профилактических работ.

Оценка этого коэффициента равна

$$k'_{\text{ТИ}} = \sum_{i=1}^m \theta_i / \left(\sum_{i=1}^m (\theta_i + \zeta_i) + \sum_{j=1}^n \varphi_j \right),$$

где φ_j — случайное время выполнения j -й профилактической работы;
 n — число профилактик за некоторое большое время T .

4.4. Марковская модель восстанавливаемого объекта

Использование марковской модели возможно только, если ВБР и ВВ имеют экспоненциальные распределения. Марковская цепь в этом случае имеет простейший вид (рис. 4.10).

Уравнения Чепмена—Колмогорова в дифференциальной форме имеют вид:

$$dp_0 / dt = -\lambda p_0(t) + \mu p_1(t),$$

$$dp_1 / dt = +\lambda p_0(t) - \mu p_1(t),$$

$$p_0(t) + p_1(t) = 1.$$

Начальные условия при этом могут быть следующими:

$$1) p_0(0) = 1, p_1(0) = 0,$$

$$2) p_0(0) = 0, p_1(0) = 1.$$

Решением данной системы линейных дифференциальных уравнений 1-го порядка являются следующие функции.

Для начальных условий 1):

$$p_0(t) = \mu/(\mu + \lambda) + \lambda/(\mu + \lambda) e^{-(\mu+\lambda)t} = k_{\Gamma}(t),$$

$$p_1(t) = \lambda/(\mu + \lambda) - \lambda/(\mu + \lambda) e^{-(\mu+\lambda)t} = k_{\Pi}(t).$$

Отсюда видно, что установившимися значениями являются

$$k_{\Gamma} = \mu/(\mu + \lambda), \quad k_{\Pi} = \lambda/(\mu + \lambda).$$

Поэтому

$$k_{\Gamma}(t) = k_{\Gamma} + k_{\Pi} e^{-(\mu+\lambda)t},$$

$$k_{\Pi}(t) = k_{\Pi} - k_{\Pi} e^{-(\mu+\lambda)t}.$$

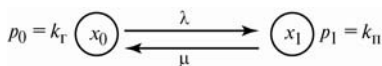


Рис. 4.10

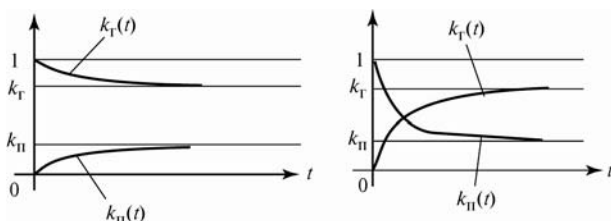


Рис. 4.11

Примечание. Стационарные коэффициенты k_T и $k_П$ могут быть получены также путем решения алгебраических уравнений Чепмена—Колмогорова.

Для начальных условий 2):

$$p_0(t) = \mu/(\mu + \lambda) - \mu/(\mu + \lambda) e^{-(\mu + \lambda)t} = k_T(t),$$

$$p_1(t) = \lambda/(\mu + \lambda) + \mu/(\mu + \lambda) e^{-(\mu + \lambda)t} = k_П(t),$$

или

$$k_T(t) = k_T - k_T e^{-(\mu + \lambda)t},$$

$$k_П(t) = k_П + k_T e^{-(\mu + \lambda)t}.$$

Соответствующие графики приведены на рис. 4.11: слева — для начальных условий 1), справа — для условий 2).

4.5. Время установления стационарного режима

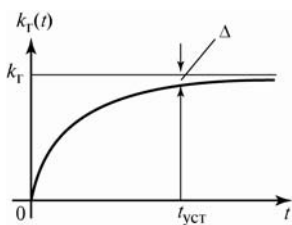


Рис. 4.12

Обычно средняя наработка на отказ T_0 составляет годы, а среднее время восстановления изделия $T_в$ (часы или дни), т.е. $\lambda \ll \mu$. Посмотрим, как быстро в этом случае затухает переходный процесс. Для определенности рассмотрим процесс, изображенный на рис. 4.12 для начальных условий 2) (см. § 4.4). Поскольку стационарный режим для $k(t)$ устанавливается лишь при $t \rightarrow \infty$, следу-

ет задаться малой относительной погрешностью

$$\delta = \Delta / k_T = [k_T - k_T(t_{уст})] / k_T,$$

при которой процесс можно считать практически установившимся.

Но

$$k_T(t_{уст}) = k_T - k_T e^{-(\mu+\lambda)t_{уст}}.$$

Поэтому

$$\delta = [k_T - k_T(t_{уст})] / k_T = 1 - [k_T - k_T e^{-(\mu+\lambda)t_{уст}}] / k_T = e^{-(\mu+\lambda)t_{уст}},$$

откуда

$$t_{уст} = - [1 / (\mu + \lambda)] \ln \delta \cong - (1/\mu) \ln \delta.$$

Следовательно, при $T_B \ll T_0$ (т.е. при $\mu \gg \lambda$) можно считать

$$t_{уст} = - T_B \ln \delta.$$

Положим $\delta = 0,01$ (1 % стационарного значения k_T). Тогда

$$t_{уст} = 4,6 T_B.$$

При начальных условиях 1) время установления будет тем же. Таким образом, можно сделать вывод, что переходный процесс, как правило, устанавливается очень быстро по сравнению с T_0 (примерно — через $5T_B$).

Пример 4.2.4 в задачнике. Наиболее частой неисправностью ПЭВМ является ее «зависание». Пусть ВБР машины имеет показательное распределение, а время перезагрузки, удаления испорченных файлов и восстановления информации составляет в среднем 10 мин (1/6 ч), а сбой в среднем случаются один раз в 5 дней (120 ч). Требуется определить вероятность того, что к моменту назначенной презентации программного продукта (через 4 ч) машина будет в работоспособном состоянии и проработает безотказно еще 2 ч.

Решение. Сначала определим, можно ли через 4 ч считать процесс установившимся:

$$t_{уст} = 4,6 T_B = 46 \text{ мин.}$$

Следовательно, через 4 ч k_T не будет зависеть от времени и равен

$$k_T = T_0 / (T_0 + T_B) = 120 / (120 + 1/6) = 0,9986.$$

Условная вероятность того, что объект проработает еще 2 ч:

$$P(t, t + \Delta t) = e^{-\lambda \Delta t} = e^{-1/120 \cdot 2} = 0,9835.$$

Тогда коэффициент оперативной готовности (совместная вероятность)

$$k_{о.г} = k_T e^{-\lambda \Delta t} = 0,9986 \cdot 0,9835 = 0,982,$$

т.е. примерно только в двух случаях из 100 машина может отказать в течение периода презентации.

5. АНАЛИЗ НАДЕЖНОСТИ НЕВОССТАНАВЛИВАЕМЫХ МНОГОЭЛЕМЕНТНЫХ СИСТЕМ

5.1. Схема надежности. Нерезервированная система без восстановления

Для удобства анализа надежностных характеристик удобно использовать схему надежности системы, которая изображается в виде неориентированного графа, обладающего следующими свойствами:

- 1) имеются две выделенные вершины A и B — вход и выход;
- 2) каждая дуга графа отражает физический элемент, отказ которого характеризуется разрывом дуги;
- 3) объект работоспособен, если существует хотя бы один путь от A к B с работоспособными элементами — дугами графа.

Нерезервированная система всегда отображается последовательным соединением элементов (рис. 5.1).

Последовательная схема называется также *основным соединением*. Если отказы — независимые события, то по правилу умножения вероятностей

$$P_c(t) = \prod_j p_j(t).$$

Теорема 5.1. Интенсивность отказов последовательной схемы надежности равна сумме интенсивностей отказов ее элементов (независимо от функции распределения ВБР):

$$\Lambda_c(t) = \sum_j \lambda_j(t).$$

Доказательство:

$$\begin{aligned} P_c(t) &= \prod_j p_j(t) = \exp \left(- \sum_j \int_0^t \lambda_j(x) dx \right) = \\ &= \exp \left(- \int_0^t \sum_j \lambda_j(x) dx \right) = \exp \left(- \int_0^t \Lambda_c(x) dx \right). \end{aligned}$$

Отсюда следует, что $\Lambda_c(t) = \sum_j \lambda_j(t)$.

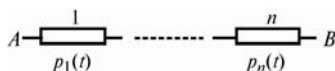


Рис. 5.1

5.2. Основное соединение

1. Система с экспоненциальными элементами (т.е. с элементами, имеющими показательное распределение ВБР).

Теорема 5.2. Последовательное соединение экспоненциальных элементов имеет показательное распределение ВБР.

Доказательство:

$$P_c(t) = \prod_j e^{-\lambda_j t} = e^{-\sum_j \lambda_j t} = e^{-\Lambda_c t}.$$

Пример 5.1. Нерезервированная схема содержит 8 транзисторов (имеющих интенсивность отказов λ_1), 4 диода (λ_2), 15 резисторов (λ_3) и 11 конденсаторов (λ_4). Физическое соединение элементов при расчете надежности знать не требуется. Определить Λ_c , $P_c(1000 \text{ ч})$ и T_c , если $\lambda_j = \lambda = 2 \cdot 10^{-6} \text{ 1/ч}$.

Решение. Поскольку λ не зависит от t , то все элементы — экспоненциальные. Находим (по теореме)

$$\Lambda_c = 8\lambda_1 + 4\lambda_2 + 15\lambda_3 + 11\lambda_4 = 76 \cdot 10^{-6} \text{ 1/ч}.$$

$$P_c(1000) = e^{-\Lambda_c \cdot 1000} = 0,924.$$

В силу второй теоремы вся система является экспоненциальной, поэтому

$$T_c = 1/\Lambda_c = 13158 \text{ ч} \approx 1,5 \text{ года}.$$

2. Система с элементами, имеющими вейбулловское распределение ВБР. Функция распределения Вейбулла—Гнеденко имеет вид:

$$F_\theta(t) = Q(t) = 1 - e^{-\alpha t^\beta},$$

где α — коэффициент масштаба; β — коэффициент формы распределения.

Плотность распределения равна производной от $F_\theta(t)$:

$$f_\theta(t) = \alpha \beta t^{\beta-1} \cdot e^{-\alpha t^\beta},$$

Вид плотности для различных значений β изображен на рис. 5.2.

Вероятность безотказной работы в интервале $[0, t]$

$$P(t) = e^{-\alpha t^\beta},$$

а интенсивность отказов

$$\lambda(t) = f_\theta(t) / P(t) = \alpha \beta t^{\beta-1}$$

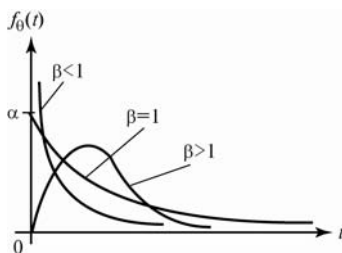


Рис. 5.2

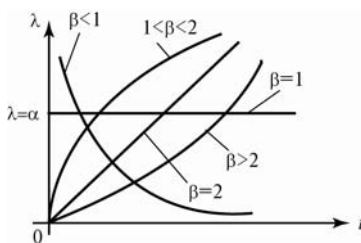


Рис. 5.3

На рис. 5.3 приведены графики $\lambda(t)$. В зависимости от коэффициента β интенсивность принимает значения:

- 1) при $\beta = 1$: $\lambda = \alpha = \text{const}$ (экспоненциальное распределение),
- 2) при $\beta < 1$: λ изменяется в пределах от ∞ до 0 («молодеющее» распределение),
- 3) при $\beta > 1$: λ изменяется в пределах от 0 до ∞ («стареющее» распределение).

Для новых электронных элементов $0,2 \leq \beta \leq 0,5$, для изношенных элементов и механических изделий $1,2 \leq \beta \leq 1,5$.

Средняя наработка до отказа

$$T = \int_0^{\infty} P(t) dt = \int_0^{\infty} e^{-\alpha t^{\beta}} dt.$$

Этот интеграл непосредственно не берется и является табличным.

Теорема 5.3. Последовательное соединение вейбулловских элементов с одинаковым коэффициентом формы имеет ВБР, распределенное по Вейбуллу.

Доказательство:

$$P_c(t) = \prod_j e^{-\alpha t^{\beta}} = e^{-\sum_j \alpha t^{\beta}} = e^{-\alpha_c t^{\beta}}.$$

5.3. Виды структурного резервирования

Рассмотрим восемь видов структурного резервирования.

1. При резервировании используют простые схемы надежности (рис. 5.4) и структурно-сложные схемы (рис. 5.5). Простые схемы надежности в свою очередь делятся:

- на параллельные (рис. 5.4, а);
- последовательно-параллельные (рис. 5.4, б);
- параллельно-последовательные (рис. 5.4, в).

Схема надежности может совпадать или не совпадать с электрическим соединением (обычно не совпадает).

Схема надежности зависит от вида отказов. Рассмотрим параллельное соединение двух диодов, в котором один из диодов — резервный. Для диодов существует два вида отказов — отказ типа «Обрыв» и отказ типа «Короткое замыкание». В первом случае имеем параллельную схему надежности, во втором — последовательную (хотя электрическая схема — параллельная).

2. По способу включения резерва (рис. 5.6) схемы резервирования могут быть:

- с постоянно включенным резервом (рис. 5.6, а);
- резервирование замещением (рис. 5.6, б);
- скользящее резервирование (рис. 5.6, в).

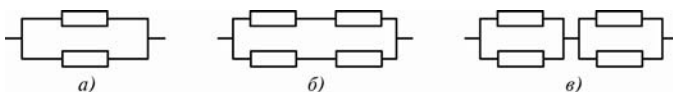


Рис. 5.4

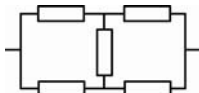


Рис. 5.5

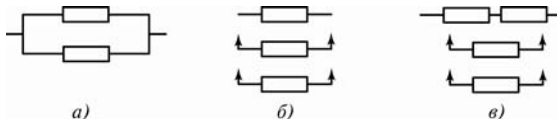


Рис. 5.6

3. По режиму резервированных элементов различают схемы:
 - с нагруженным резервом ($\lambda_{рез} = \lambda_{осн} > 0$);
 - с ненагруженным резервом ($\lambda_{рез} = 0, \lambda_{осн} > 0$);
 - с облегченным резервом ($0 < \lambda_{рез} < \lambda_{осн}$).
4. По восстанавливаемости схемы могут быть:
 - без восстановления;
 - с восстановлением.
5. По времени восстановления:
 - с мгновенным восстановлением ($\xi = 0$);
 - с задержанным восстановлением ($\xi > 0$).
6. По числу одновременно восстанавливаемых элементов:
 - с неограниченным восстановлением (одновременно восстанавливаются все отказавшие элементы);
 - с полностью ограниченным восстановлением (отказавшие элементы восстанавливаются последовательно);
 - с частично ограниченным восстановлением (одновременно восстанавливается ограниченное число отказавших элементов).
7. По дисциплине восстановления (устанавливается очередность восстановления).
8. По полноте восстановления:
 - с полным восстановлением (при отказе любого элемента изделия производится замена всего изделия на новое — с невыработанным ресурсом);
 - с неполным восстановлением (заменяется только отказавший элемент).

5.4. Общее и раздельное резервирование с постоянно включенным резервом

5.4.1. Общее резервирование

Схема надежности — параллельная или последовательно-параллельная (рис. 5.7).

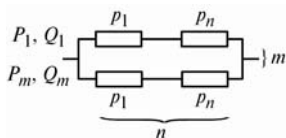


Рис. 5.7

Для этой схемы

$$P_i = \prod_{j=1}^n p_j, \quad Q_i = 1 - \prod_{j=1}^n p_j,$$

$$Q_c = \prod_{i=1}^m Q_i = \prod_{i=1}^m \left(1 - \prod_{j=1}^n p_j \right), \quad P_c = 1 - Q_c = 1 - \prod_{i=1}^m \left(1 - \prod_{j=1}^n p_j \right).$$

При $P_i = P, Q_i = Q$

$$P_c = 1 - \left(1 - \prod_{j=1}^n p_j \right)^m = 1 - Q^m.$$

Отсюда видно, что $P_c \rightarrow 1$ при $m \rightarrow \infty$.

Средняя наработка до отказа определяется формулой

$$T_c = \int_0^{\infty} P_c(t) dt,$$

а интенсивность отказов системы

$$\Lambda_c = -\frac{dP_c/dt}{P_c(t)}; \quad \Lambda_c \neq 1/T_c.$$

Рассмотрим параметры надежности системы с экспоненциальными элементами.

Вероятность безотказной работы системы

$$P_c(t) = 1 - (1 - e^{-\Lambda t})^m,$$

где $\Lambda = \sum_{j=1}^n \lambda_j$ — интенсивность отказа последовательной цепочки.

Средняя наработка до отказа

$$T_c = \int_0^{\infty} P_c(t) dt = \int_0^{\infty} \left[1 - (1 - e^{-\Lambda t})^m \right] dt.$$

Обозначим

$$z = 1 - e^{-\Lambda t}.$$

При $t = 0$ получим $z = 0$, при $t \rightarrow \infty$: $z \rightarrow 1$. Имеем

$$dz = \Lambda e^{-\Lambda t} dt, \quad dt = \frac{1}{\Lambda} \frac{1}{1-z} dz,$$

$$T_c = \frac{1}{\Lambda} \int_0^1 \frac{1-z^m}{1-z} dz.$$

Но под знаком интеграла стоит сумма членов геометрической прогрессии со знаменателем z :

$$\frac{1-z^m}{1-z} = 1 + z + z^2 + \dots + z^{m-1}.$$

Подставив эту сумму в формулу для T_c и взяв интеграл, получим:

$$T_c = \frac{1}{\Lambda} (1 + 1/2 + 1/3 + \dots + 1/m) = T (1 + 1/2 + 1/3 + \dots + 1/m),$$

где $T = 1/\Lambda$.

Интенсивность отказов системы

$$\Lambda_c = -\frac{dP_c / dt}{P_c}.$$

Рассмотрим систему, состоящую из элементов с *вейбулловским распределением* ВБР

При $p_j = p$

$$p_j = e^{-\alpha_j t^\beta}, \quad P_c = 1 - \left(1 - e^{\alpha t^\beta}\right)^m,$$

где $\alpha_c = \sum_{j=1}^n \alpha_j$ — масштабный параметр последовательной цепочки.

Каждая последовательная цепочка с экспоненциальными элементами имеет ВБР, распределенное экспоненциально. Однако резервированная система уже не является экспоненциальной. Аналогично — для вейбулловских элементов: последовательная цепочка с вейбулловскими элементами имеет такое же распределение ВБР, но функция распределения ВБР резервированной системы — другое.

5.4.2. Раздельное резервирование

Схема надежности — параллельно-последовательная (рис. 5.8).

При $q_i = q$

$$Q_j = q^m, \quad P_j = 1 - q^m, \quad P_c = \prod_{j=1}^n P_j = \prod_{j=1}^n [1 - (1 - p)^m]_j.$$

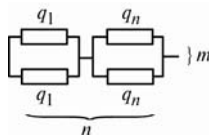


Рис. 5.8

P_c также стремится к 1 при неограниченном увеличении числа резервных элементов.

Рассмотрим схему с экспоненциальными элементами.

Вероятность безотказной работы системы

$$P_c = \prod_{j=1}^n [1 - (1 - e^{-\lambda t})^m]_j.$$

Средняя наработка до отказа вычисляется по обычной формуле:

$$T_c = \int_0^{\infty} P_c(t) dt,$$

а интенсивность отказов системы — по формуле

$$\Lambda_c = -\frac{dP_c(t)/dt}{P_c(t)}; \quad \Lambda_c \neq 1/T_c.$$

5.4.3. Сравнение по надежности схем с общим и раздельным резервированием

Хотя при неограниченном увеличении резервных элементов $P_c(t) \rightarrow 1$, схемы общего и раздельного резервирования с одинаковым конечным числом элементов различаются по надежности. Для определенности выполним сравнение по случайному времени отказа θ для схем с 4 элементами при любой функции распределения ВБР (рис. 5.9). Используя индукцию, можно убедиться в справедливости выводов для любого числа элементов.

Обозначим: $\theta_1, \theta_2, \theta'_1, \theta'_2$ — случайное время до отказа соответствующих элементов; τ_1 — случайное время до отказа системы с общим

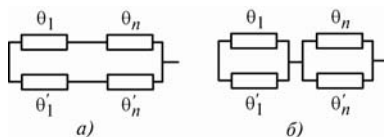


Рис. 5.9

резервированием; τ_2 — случайное время до отказа системы с раздельным резервированием.

Тогда первая схема (рис. 5.9, а) откажет, если откажет по одному элементу в верхней и нижней ветви:

$$\tau_1 = \max (\min(\theta_1, \theta_2), \min(\theta'_1, \theta'_2)).$$

Но

$$\min(\theta_1, \theta_2) \leq \theta_1,$$

$$\min(\theta'_1, \theta'_2) \leq \theta'_1.$$

Следовательно,

$$\tau_1 \leq \max (\theta_1, \theta'_1).$$

Аналогично получаем

$$\tau_1 \leq \max (\theta_2, \theta'_2).$$

Следовательно,

$$\tau_1 \leq \min [\max (\theta_1, \theta'_1), \max (\theta_2, \theta'_2)].$$

Вторая схема (рис. 5.9, б) откажет, если откажут оба элемента в левой или в правой секции. Поэтому

$$\tau_2 = \min [\max (\theta_1, \theta'_1), \max (\theta_2, \theta'_2)].$$

Отсюда следует, что

$$\tau_1 \leq \tau_2.$$

Вывод: чем мельче масштаб резервирования, тем выше надежность системы.

5.5. Резервирование замещением и скользящее резервирование

5.5.1. Резервирование замещением

При анализе предполагается мгновенное замещение. Рассмотрим сначала систему, состоящую из одного основного и m_0 резервных элементов (рис. 5.10). При этом общее число элементов $n = m_0 + 1$.

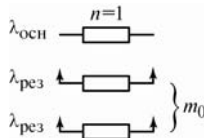


Рис. 5.10

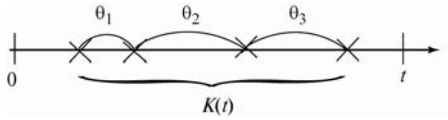


Рис. 5.11

Нагруженный резерв. В этом случае показатели надежности системы совпадают с показателями при общем нагруженном резервировании с постоянно включенным резервом.

Ненагруженный резерв (экспоненциальные элементы). Рассмотрим случай, когда все элементы системы имеют *экспоненциальное распределение ВБР*. В этом случае $\lambda_{\text{осн}} = \lambda > 0$, $\lambda_{\text{рез}} = 0$.

Для вывода показателей воспользуемся теорией случайных потоков. На рис. 5.11 изображен точечный поток последовательных отказов элементов с замещением их резервными (пока они не будут исчерпаны). Пусть θ — *непрерывная*, а $K(t)$ — *дискретная* случайная величина — число отказов в интервале $[0, t]$.

Поскольку переменная θ распределена экспоненциально, то $K(t)$ имеет пуассоновское распределение:

$$p_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t},$$

где $p_k(t)$ — вероятность $P[K = k \text{ в интервале от } 0 \text{ до } t]$, $k \in [0, \infty]$.

В силу несовместимости событий $K(t)$ вероятность безотказной работы системы

$$P_c(t) = p_0(t) + p_1(t) + \dots + p_{m-1}(t) = e^{-\lambda t} \sum_{k=0}^{m-1} \frac{(\lambda t)^k}{k!} = e^{-\lambda t} \sum_{k=0}^{m_0} \frac{(\lambda t)^k}{k!}, \quad (5.1)$$

где m_0 — число резервных элементов.

Найдем среднюю наработку T_c до отказа системы. Средняя наработка до отказа основного элемента $T = M[\theta]$. Такие же значения имеют наработки резервных элементов после замещения ими основного элемента. Поэтому

$$T_c = mT = (m_0 + 1) T,$$

где $T = 1/\lambda$.

Интенсивность отказов системы определяется по формуле:

$$\Lambda_c = -\frac{dP_c(t)/dt}{P_c(t)}.$$

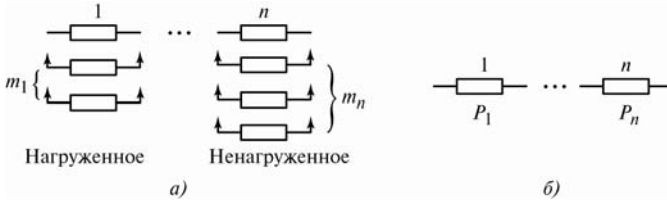


Рис. 5.12

Сравним значения T_c для нагруженного и ненагруженного резервирования.

$$T_{c. \text{нагр}} = T(1 + 1/2 + 1/3 + \dots + 1/m) < mT = T_{c. \text{ненагр}},$$

Вывод: если пренебречь ненадежностью коммутатора, подключающего резервные элементы, то применение ненагруженного резервирования вместо нагруженного повышает надежность, в противном случае нужно учитывать характеристики коммутатора.

Для отдельного резервирования замещением с различными видами нагрузки (рис. 5.12, а) расчет ведется поэтапно: сначала рассчитывается надежность резервированных групп, затем надежность последовательного соединения этих групп (рис. 5.12, б).

Для общего ненагруженного резервирования замещением с экспоненциальными элементами (последовательно-параллельная схема) (рис. 5.13), расчет вероятности безотказной работы системы проводится по формуле (5.1), в которой вместо λ используется интенсив-

ность отказа последовательного соединения элементов $\Lambda = \sum_{j=1}^n \lambda_j$.

Тогда

$$P_c = e^{-\Lambda t} \sum_{k=1}^{m_0} \frac{(\Lambda t)^k}{k!}, \quad (5.2)$$

Средняя наработка до отказа такой системы

$$T_c = (m_0 + 1) T,$$

где $T = 1/\Lambda$.

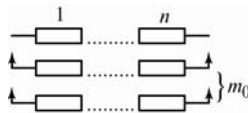


Рис. 5.13

5.5.2. Скользящее резервирование

Используется в системах с одинаковыми элементами. Схема скользящего резервирования имеет вид, приведенный на рис. 5.14

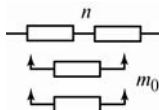


Рис. 5.14

Нагруженный резерв. Система работоспособна, если отказало $k \leq m_0$ элементов (основных или резервных). Общее число элементов в системе равно $N = n + m_0$. Число возможных отказов k элементов из N составляет $C_{n+m_0}^k$, а вероятность каждой комбинации равна $q^k p^{n+m_0-k}$. Тогда

$$P_c = \sum_{k=0}^{m_0} C_{n+m_0}^k q^k p^{n+m_0-k}.$$

Рассмотрим систему с экспоненциальными элементами.
Вероятность безотказной работы

$$P_c = \sum_{k=0}^{m_0} C_{n+m_0}^k (1 - e^{-\lambda t})^k e^{-(n+m_0-k)\lambda t}.$$

Для определения T_c составим *марковскую модель гибели*, в которой состояние x_i соответствует отказу i элементов (рис. 5.15):

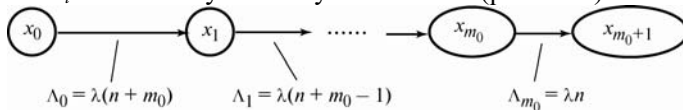


Рис. 5.15

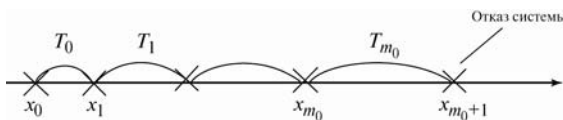


Рис. 5.16

Средняя наработка системы до перехода в состояние x_1 составляет

$$T_0 = \frac{1}{\Lambda_0} = \frac{1}{\lambda(n + m_0)},$$

до перехода в состояние x_2

$$T_1 = \frac{1}{\lambda(n + m_0 - 1)},$$

до перехода в состояние x_{m_0+1}

$$T_{m_0} = \frac{1}{\lambda n}.$$

Таким образом, $T_0 < T_1 < T_2 < \dots$ (рис. 5.16).

Средняя наработка системы до отказа составляет

$$\begin{aligned} T_c &= \sum_{i=0}^{m_0} T_i = \frac{1}{\lambda} [1/n + 1/(n + 1) + \dots + 1/(n + m_0)] = \\ &= T \cdot [1/n + 1/(n + 1) + \dots + 1/(n + m_0)], \end{aligned}$$

где $T = 1/\lambda$ (для отдельного элемента).

Ненагруженный резерв (экспоненциальные элементы). Условия работоспособности системы те же: $k \leq m_0$, но отказать могут только элементы, находящиеся в данный момент в нагруженном состоянии, т.е. в последовательной цепочке. Обозначим интенсивность отказа этой цепочки $\Lambda = n\lambda$. Вероятность k -кратного отказа экспоненциальной цепи с мгновенным замещением отказавшего элемента подчиняется пуассоновскому закону:

$$p_k(t) = \frac{(\Lambda t)^k}{k!} e^{-\Lambda t}.$$

Тогда вероятность безотказной работы системы

$$P_c(t) = p_0(t) + p_1(t) + p_{m_0}(t) = e^{-\Lambda t} \sum_{k=0}^{m_0} \frac{(\Lambda t)^k}{k!}.$$

Эта формула совпадает с выражением (5.2) для последовательно-параллельной схемы с ненагруженным резервом, но с одинаковыми элементами. Однако в случае скользящего резервирования число элементов намного меньше.

Рассчитаем среднюю наработку до отказа. Для последовательной цепочки имеем:

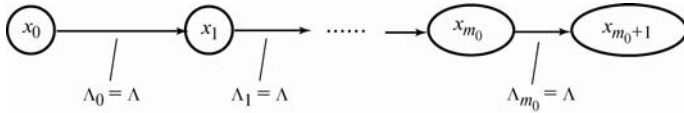


Рис. 5.17

$$T = 1/\Lambda = 1/(n\lambda).$$

Отказ системы произойдет, когда будет исчерпан весь резерв, т.е. при отказе $(m_0 + 1) = m$ элементов. На рис. 5.17 этот процесс представлен в виде марковской схемы гибели. Из рисунка видно, что

$$T_c = T_1 + T_2 + \dots + T_{m_0+1} = (m_0 + 1)(1/\Lambda) = (m_0 + 1) T.$$

Эта формула также совпадает с формулой последовательно-параллельного ненагруженного резервирования.

Выводы: 1) при одинаковом числе элементов скользящее резервирование обеспечивает большую надежность, чем последовательно-параллельная схема. Можно также показать, что при том же условии надежность будет выше, чем при использовании параллельно-последовательной схемы; 2) концентрация резервов повышает надежность системы.

6. АНАЛИЗ НАДЕЖНОСТИ СИСТЕМ СО СЛОЖНОЙ СТРУКТУРОЙ

6.1. Расчет показателей произвольной схемы надежности с использованием логических условий безотказности. Метод структурных функций

Ранее рассматривались невосстанавливаемые системы с относительно простой структурой, сводящейся к последовательно-параллельным или параллельно-последовательным схемам с нагруженным и ненагруженным резервом. Исключением была структура скользящего резервирования.

Во многих случаях условиям безотказности невозможно или затруднительно сопоставить схему надежности с независимыми путями, т.е. с путями, не содержащими одноименных элементов (дуг). В этом случае обычно вообще не используется схема надежности, а применяются другие подходы к описанию условий безотказности. Одним из них является *метод структурных функций*, заключающийся в составлении логической модели безотказности системы с последующим вычислением вероятностных характеристик. С его помощью можно вычислять следующие показатели надежности: для невосстанавливаемых объектов — вероятности безотказной работы системы в заданном интервале $P_c(t)$ или $P_c(t, t + \Delta t)$; для восстанавливаемых — коэффициенты готовности и простоя $k_{г.с}(t)$, $k_{п.с}(t)$ или коэффициент оперативной готовности $k_{о.г.с}(t, t + \Delta t)$.

Рассмотрим в качестве примера *скользящее нагруженное резервирование* с двумя основными и двумя резервными элементами (рис. 6.1).

Сначала выполняется словесное описание условий безотказности: «Система работоспособна, если работоспособны блоки 1 и 2 или 1 и 3, или 1 и 4, или 2 и 3, или 2 и 4, или 3 и 4, или 1 и 2 и 3 и т.д.»

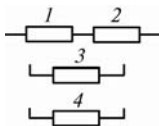


Рис. 6.1

Затем словесное описание представляется в форме логической структурной функции:

$$F(x) = F(x_1, x_2, \dots, x_n),$$

где $x_i = 1$ (или $F = 1$), если i -й элемент системы (или весь объект) работоспособен, и $x_i = 0$ — если отказал:

$$F(x) = x_1 x_2 \vee x_1 x_3 \vee x_1 x_4 \vee x_2 x_3 \vee x_2 x_4 \vee x_3 x_4 \vee x_1 x_2 x_3 \vee \dots \vee x_1 x_2 x_3 x_4.$$

После сокращений получим

$$F(x) = x_1 x_2 \vee x_1 x_3 \vee x_1 x_4 \vee x_2 x_3 \vee x_2 x_4 \vee x_3 x_4.$$

Каждый терм обозначим буквой A_i . Тогда

$$F(x) = A_1 \vee A_2 \vee A_3 \vee A_4 \vee A_5 \vee A_6.$$

Переход к вероятностному представлению безотказности можно выполнить двумя путями.

1) Первый путь состоит в использовании правила сложения вероятностей совместимых событий. Для этого структурную функцию удобно представить в арифметизированной форме, используя правило объединения множеств (которому соответствует правило сложения совместимых событий):

$$A_1 \vee A_2 \vee \dots \vee A_m = (A_1 + A_2 + \dots + A_m) - (A_1 A_2 + A_1 A_3 + \dots + A_{m-1} A_m) + (A_1 A_2 A_3 + A_1 A_2 A_4 + A_{m-2} A_{m-1} A_m) - \dots + A_1 A_2 \dots A_m = \sum_{k=1}^s (-1)^k B_k,$$

где B_k — выражение в k -й скобке; s — число скобок, а произведение $A_i A_j$ обладает свойством идемпотентности: $x \cdot x = x$. Например,

$$A_1 A_2 = x_1 x_2 \cdot x_1 x_3 = x_1 x_2 x_3.$$

Эта формула по структуре совпадает с формулой сложения вероятностей совместимых событий A_i .

После арифметизации вместо x_i подставляем показатель надежности элемента p_i , а вместо $F(x_1, \dots, x_m)$ — показатель надежности P_c системы. Тогда

$$P_c = F(p_1, \dots, p_n).$$

Аналогично при расчете коэффициента готовности $k_{г.с}$ (в случае восстанавливаемой системы) вместо x_i подставляется $k_{г. i}$ соответствующего элемента:

$$k_{Г.с} = F(k_{Г1}, \dots, k_{Гn}).$$

Например, если система — восстанавливаемая с неограниченным режимом восстановления, то

$$k_{Г.с} = (k_{Г1}k_{Г2} + k_{Г1}k_{Г3} + \dots) - (k_{Г1}k_{Г2}k_{Г3} + k_{Г1}k_{Г2}k_{Г4} + \dots) + \dots - k_{Г1}k_{Г2}k_{Г3}k_{Г4}.$$

Приведя подобные члены и подставив значения $k_{Гi}$, находим $k_{Г.с}$.

Этот способ расчета, связанный с арифметизацией $F(x)$, используется при малом числе термов A_i .

2) Второй путь не требует арифметизации и заключается в ортогонализации термов булевой функции $F(x)$, которую можно выполнить различными путями. Ортогонализация состоит в нахождении эквивалентной логической функции $F(x)$, все термы которой попарно ортогональны, т.е. $A_i \cdot A_j \equiv 0$. Например, совершенная дизъюнктивная нормальная форма (СДНФ) функции содержит только ортогональные термы. Тогда термы отображают несовместимые события, и их вероятности просто складываются:

$$P_c = P(A_1) + P(A_2) + \dots,$$

где $P(A_i)$ представляют собой произведения вероятностей p_k элементов, отображаемых переменными x_k терма A_i . Таким же образом выполняется расчет коэффициента готовности системы $k_{Г.с}$ или оперативной готовности $k_{ог.с}$.

Данный способ (ортогонализация) применяется при незначительном числе переменных x_j .

Ограничения на использование метода структурных функций.

1. При расчете вероятности безотказной работы системы предполагается, что отказы элементов системы — независимые события. Это означает, что метод применим только при нагруженном резервировании, так как при ненагруженном резервировании вероятность отказа резервных элементов зависит от их режима нагрузки. Действительно, при нахождении элемента в резерве интенсивность его отказа равна нулю, а при замене им отказавшего основного элемента она возрастает в связи с изменением его режима. Другими словами, вероятность отказа резервного элемента зависит от состояния основных элементов.

2. При расчете коэффициента готовности должны быть независимыми как отказы элементов, так и процедуры их восстановления. Это означает, что каждый отказавший элемент должен восстанавливаться

независимо от состояния других, т.е. число обслуживающих (восстанавливающих) устройств должно быть равно числу отказавших элементов системы.

Применение структурных функций для анализа схем надежности. Построение структурной функции облегчается, если предварительно построена схема надежности с независимыми путями. В этом случае применяется *метод минимальных путей* либо *метод минимальных сечений*.

В первом случае выполняются следующие шаги:

- 1) Определяется множество минимальных путей в графе от входной до выходной вершины схемы надежности. Минимальным называется элементарный путь (без повторяющихся дуг и контуров) между указанными вершинами.
- 2) Каждому найденному минимальному пути соответствует терм, состоящий из конъюнкции переменных, сопоставляемых дугам этого пути: $A_i = x_i x_j x_k \dots$
- 3) Составляется структурная функция, представляющая собой дизъюнкцию полученных термов.

$$F(x) = A_1 \vee A_2 \vee \dots$$

Далее по полученной структурной функции находится выражение для расчета показателей надежности.

Пример 6.1. Для схемы надежности, приведенной на рис. 6.2 построить структурную функцию.

В данном случае имеем 8 минимальных путей: (1-3-7), (1-3-5-6-8), (1-4-8), (1-4-6-5-7), (2-6-8), (2-6-4-3-7), (2-5-7), (2-5-3-4-8). Каждому минимальному пути сопоставляется терм структурной функции. В результате получаем выражение

$$F(x) = x_1 x_3 x_7 \vee x_1 x_3 x_5 x_6 x_8 \vee x_1 x_4 x_8 \vee x_1 x_4 x_6 x_5 x_7 \vee x_2 x_6 x_8 \vee x_2 x_6 x_4 x_3 x_7 \vee x_2 x_5 x_7 \vee x_2 x_5 x_3 x_4 x_8.$$

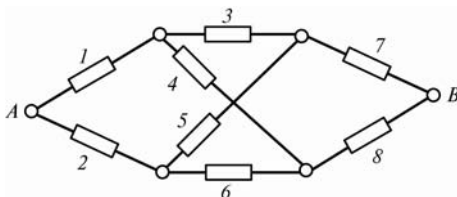


Рис. 6.2

Далее рассчитывается надежность P_c , $k_{г.с}$ или $k_{о.г.с}$. Для этого вместо x_i подставляется p_i либо $k_{г.с}$, либо $k_{о.г.с}$.

Иногда вместо надежности удобнее рассчитывать показатель ненадежности Q_c или $k_{п.с}$. Для этого используется метод минимальных сечений графа, при котором выполняются следующие шаги:

- 1) Определяется множество минимальных сечений. Минимальным сечением называется минимальная совокупность дуг, разрыв которых приводит к потере связности вершин A и B . На рис. 6.2 такими сечениями являются $(1,2)$, $(1,5,6)$ и т.д. Однако сечение $(1,4,5,6)$ не является минимальным, так как содержит в себе минимальное сечение $(1,5,6)$.
- 2) Каждому минимальному сечению сопоставляется соответствующий терм.
- 3) Составляется структурная функция в форме отрицания:

$$\bar{F} = x_1 x_2 \vee x_1 x_5 x_6 \vee \dots$$

Затем рассчитывается Q_c , но вместо p_k подставляются $q_k = 1 - p_k$, а вместо \bar{F} ненадежность Q_c . Тогда $P_c = 1 - Q_c$. Аналогично рассчитываются $k_{п}$ и $k_{г}$ для восстанавливаемых систем.

Минимальные сечения используются, если их число меньше, чем число минимальных путей.

Пример 6.2. Рассчитать P_c схемы общего нагруженного резервирования (рис. 6.3).

Здесь имеются два пути и одно сечение, поэтому целесообразно использовать метод сечений.

$$\bar{F} = x_1 x_2.$$

Вместо x_i подставляем q_i и находим

$$Q_c = q_1 q_2 = (1 - p_1) (1 - p_2) = 1 - p_1 - p_2 + p_1 p_2,$$

$$P_c = 1 - Q_c = p_1 + p_2 - p_1 p_2 = 2p - p^2.$$

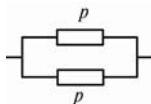


Рис. 6.3

6.2. Мажоритарное резервирование

Мажоритарная схема приведена на рис. 6.4. Она состоит из нечетного числа n одинаковых функциональных блоков (каналов), в данном случае $n = 3$, и кворум-элемента θ , сравнивающего выходы каналов между собой. Все сигналы z_i при исправных каналах совпадают. При отказе некоторых каналов их выходы не будут совпадать с выходами исправных каналов. Кворум-элемент формирует сигнал F , совпадающий с большинством сигналов z_i . Если число исправных каналов $k \geq m$, где

$$m = \lceil n/2 \rceil = (n + 1)/2,$$

то решение будет правильным. Для кворум-элементов принято обозначение « m/n » (2/3, 3/5, 4/7 и т.д.) или « $\geq m$ ».

Мажоритарное резервирование можно анализировать как схему скользящего резервирования, содержащую m основных и $n - m$ резервных элементов, но можно использовать метод структурных функций.

Рассмотрим схему троирования ($m/n = 2/3$) и составим структурную функцию на основе логических условий безотказности системы:

$$F(x) = x_1 x_2 \vee x_1 x_3 \vee x_2 x_3 \vee x_1 x_2 x_3 = x_1 x_2 \vee x_1 x_3 \vee x_2 x_3.$$

Полученные термы обозначим A_1 , A_2 и A_3 . Тогда структурная функция в арифметизированной форме имеет вид

$$\begin{aligned} & (A_1 + A_2 + A_3) - (A_1 A_2 + A_1 A_3 + A_2 A_3) + A_1 A_2 A_3 = \\ & = (x_1 x_2 + x_1 x_3 + x_2 x_3) - (x_1 x_2 x_3 + x_1 x_2 x_3 + x_1 x_2 x_3) + x_1 x_2 x_3 = \\ & = x_1 x_2 + x_1 x_3 + x_2 x_3 - 2x_1 x_2 x_3. \end{aligned}$$

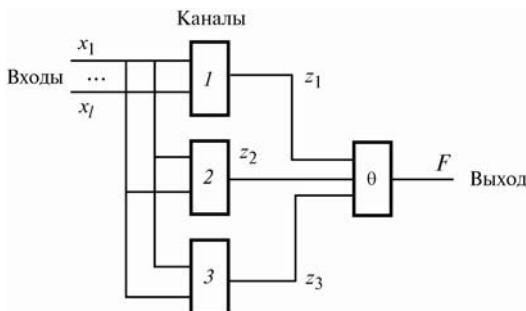


Рис. 6.4

Поскольку каналы одинаковые, то все $p_k = p$ и

$$P_{2/3} = 3p^2 - 2p^3.$$

При $p_0 < 1$ вероятность $P_c = P_{2/3}/p_0$, а при $p_0 = 1$ вероятность $P_c = P_{2/3}$. Зависимость $P_c = P_{2/3}$ от надежности одного канала p изображена на рис. 6.5.

Нетрудно показать, что для любой мажоритарной схемы с нечетным n и $m = (n + 1)/2$ (т.е. для схем 2/3, 3/5, 4/7 и т.д.)

$$P_{m/n} = \sum_{k=0}^{(n-1)/2} C_n^k q^k p^{n-k}.$$

Схема с экспоненциальными элементами. Рассмотрим схему троирования, в которой ВБР каналов имеют показательное распределение с параметром λ . Тогда

$$P_c = 3e^{-2\lambda t} - 2e^{-3\lambda t},$$

$$T_c = \int_0^{\infty} (3e^{-2\lambda t} - 2e^{-3\lambda t}) dt = (5/6)1/\lambda = (5/6)T,$$

где T — среднее время до отказа канала.

Из графика видно, что при $p(t) > 0,5$ значение $P_c(t) > p(t)$, а из последней формулы следует, что $T_c < T$ при любом p , т.е. имеют место противоречивые показатели и нужно выбирать — что важнее.

При большом t может оказаться, что $p(t) < 0,5$, и тогда $P_c(t) < p(t)$, при этом оба показателя для системы троирования будут хуже, чем для одного канала. Критическим является значение $t = t_0$, для которого $p(t_0) = e^{-\lambda t_0} = 0,5$. Отсюда

$$t_0 = \frac{+\ln 2}{\lambda} = 0,7T.$$

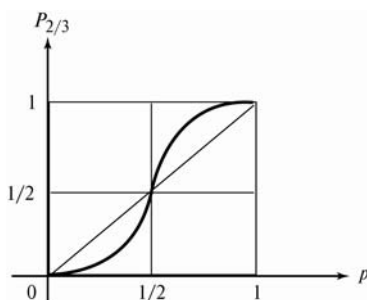


Рис. 6.5

Для обеспечения условия $t < t_0$ выполняются сеансы контроля состояния каналов с интервалами $t_k < t_0$. Еще лучше обеспечивать непрерывный аппаратный контроль.

Отметим, что троирование применяется в основном для маскировки сбоев, поскольку они возникают гораздо чаще устойчивых дефектов, а их последствиями являются ошибки вычислений или зависания системы (что эквивалентно устойчивым отказам). Мажоритарное резервирование практически исключает влияние сбоев, а аппаратный контроль позволяет следить за частотой возникающих сбоев, позволяющей судить о степени износа каналов.

6.3. Мажорирование функциональных блоков ВС

Нерезервированная ВС имеет последовательную схему надежности (рис. 6.6, а).

Резервированная имеет мажорированные блоки (m/n) и кворум-элементы θ (рис. 6.6, б). В этом случае надежность системы

$$P = p_1 p_{\theta 1} + p_2 p_{\theta 2} + \dots,$$

где p_j характеризует надежность мажорированных блоков, а $p_{\theta j}$ — надежность схем голосования. Кроме того, в эту последовательную цепочку включаются схемы контроля состояния каналов (соответствующие вероятности включены в показатели $p_{\theta j}$).

Слабым звеном такой схемы мажорирования являются нерезервированные кворум-элементы (со схемами контроля), при отказе которых отказывает вся система. Этого недостатка лишена *слоистая схема*, в которой эти элементы также мажорируются. Отметим, что в большинстве случаев вероятность отказа кворум-элемента значительно меньше вероятности отказа канала, и ею можно пренебречь.

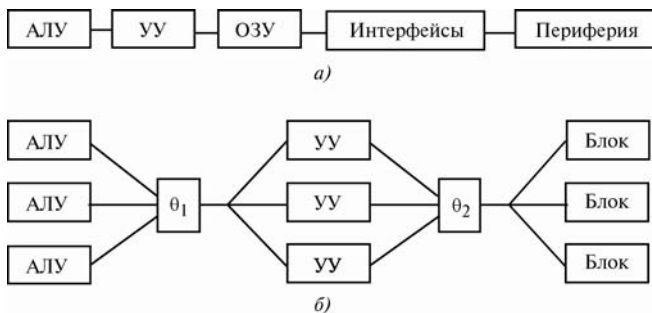


Рис. 6.6

6.4. Системы с реконфигурацией и восстановлением

6.4.1. Системы с реконфигурацией

Реконфигурация позволяет повысить надежность системы. Рассмотрим систему 2/3 без восстановления (ненадежностью кворум-элемента пренебрежем). После отказа одного из трех каналов система (еще не отказавшая) переходит к состоянию без резервирования. Для этого отключается второй (не отказавший) канал, т.е. осуществляется реконфигурация системы.

Обозначим состояния системы:

x_0 — работоспособны все три канала;

x_1 — отказал один канал, переход к работе с одним из двух работоспособных каналов;

x_2 — отказ оставшегося канала и всей системы.

Составим схему гибели (рис. 6.7) и запишем уравнения Колмогорова:

$$dp_0(t)/dt = -3\lambda p_0(t),$$

$$dp_1(t)/dt = 3\lambda p_0(t) - \lambda p_1(t),$$

$$dp_2(t)/dt = \lambda p_1(t),$$

$$p_0(t) + p_1(t) + p_2(t) = 1.$$

Решением этой системы при начальных условиях $p_0(0) = 1, p_1(0) = 0, p_2(0) = 0$ являются функции:

$$p_0(t) = e^{-3\lambda t},$$

$$p_1(t) = (3/2)(e^{-\lambda t} - e^{-3\lambda t}),$$

$$p_2(t) = 1 - p_0(t) - p_1(t).$$

Вероятность безотказной работы системы равна

$$P_c(t) = p_0(t) + p_1(t) = 3/2 e^{-\lambda t} - 1/2 e^{-3\lambda t}.$$

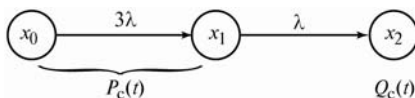


Рис. 6.7

Для системы без реконфигурации было

$$P_c(t) = 3 e^{-2 \lambda t} - 2 e^{-3 \lambda t}.$$

При $t = 0$ вероятность $P_c(0) = P_{2/3}(0) = 1$, а при $t > 0$ производная $dP_{2/3}(t)/dt$ более отрицательная функция, чем $dP_c(t)/dt$ для системы с реконфигурацией (рис. 6.8).

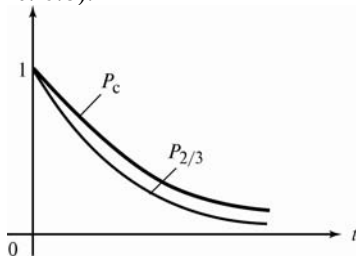


Рис. 6.8

Следовательно, $P_c(t) > P_{2/3}(t)$ при любом $t > 0$.

Средняя наработка до отказа такой системы

$$T_c = \int_0^{\infty} P_c(t) dt = (4/3)T,$$

где $T = 1/\lambda$ — наработка для одного канала.

Сравнивая полученные выражения для T_c и $T_{2/3}$ (см. с. 78), видим, что $T_c = (4/3)T > T_{2/3} = (5/6)T$, т.е. T_c больше $T_{2/3}$ в 1,6 раза.

6.4.2. Системы с восстановлением

Восстановление всегда позволяет намного повысить надежность системы. Для анализа надежности обычно используются марковские модели и иногда аппарат структурных функций.

Пример 6.3. В мажоритарной системе 2/3 с независимым восстановлением каналов средняя наработка на отказ каждого канала составляет $T_0 = 10\,000$ ч, а среднее время восстановления $T_v = 20$ ч. Требуется определить стационарный коэффициент готовности системы $k_{г.с}$ и среднюю наработку на отказ T_0 с и сравнить показатели надежности с показателями невосстанавливаемой системы и системы с реконфигурацией. Предполагается, что ВБР каждого канала имеет экспоненциальное распределение. Ненадежностью кворум-элемента пренебречь.

Решение. Структурная функция системы (при $x_i = x$) имеет вид:

$$F(x) = x_1 x_2 + x_1 x_3 + x_2 x_3 - 2 x_1 x_2 x_3 = 3x^2 - 2x^3.$$

Подставляя k_{Ti} вместо x_i , получим (при $k_{Ti} = k_T$)

$$k_{T.c} = 3 k_T^2 - 2 k_T^3.$$

Введем для удобства обозначение

$$\rho = \lambda/\mu = T_B/T_0 = 0,002.$$

Тогда коэффициент готовности канала

$$k_T = \mu/(\mu + \lambda) = 1/(1 + \rho) = 0,998,$$

а коэффициент готовности системы

$$k_{T.c} = 3/(1 + \rho)^2 - 2/(1 + \rho)^3 = (1 + 3\rho)/(1 + \rho)^3 = 0,999988.$$

Определим среднюю наработку системы на отказ T_{0c} .

Частота отказов системы (параметр альтернирующего потока)

$$\omega_c = 1/(T_{0c} + T_{B.c}).$$

Поскольку

$$k_{T.c} = T_{0c}/(T_{0c} + T_{B.c}),$$

то

$$T_{0c} = k_{T.c} / \omega_c.$$

Чтобы найти ω_c , составим марковскую модель гибели и размножения (рис. 6.9). Вершине x_i соответствует состояние отказа i каналов. Интенсивность перехода $\lambda_{01} = 3\lambda$, поскольку может отказать любой из трех каналов. Соответственно $\lambda_{12} = 2\lambda$, $\lambda_{23} = \lambda$. Аналогично $\mu_{32} = 3\mu$ (одновременно восстанавливаются все три канала), $\mu_{21} = 2\mu$, $\mu_{10} = \mu$. Коэффициент готовности системы $k_{T.c} = p_0 + p_1$, коэффициент простоя $k_{п.с} = p_2 + p_3$, так как при отказе двух или трех элементов происходит отказ системы. Составим первое уравнение Колмогорова (в алгебраической форме):

$$\mu p_1 - 3\lambda p_0 = 0,$$

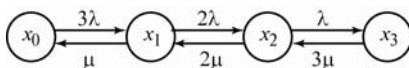


Рис. 6.9

откуда

$$p_1 = 3\rho p_0.$$

Тогда

$$k_{г.с} = p_0 + p_1 = p_0 (1 + 3\rho).$$

Система может отказать, только находясь в состоянии x_1 . Интенсивность отказа (условная частота отказов системы, находящейся в этом состоянии) $\lambda_{12} = 2\lambda$, а безусловная частота отказов

$$\omega_c = p_1 \lambda_{12} = p_1 2\lambda = 3\rho p_0 2\lambda = 6\rho\lambda p_0.$$

Тогда средняя наработка на отказ системы

$$T_{0с} = k_{г.с} / \omega_c = (1 + 3\rho)p_0 / (6\rho\lambda p_0) = (1 + 3\rho) / (6\rho\lambda) \approx 838\,000 \text{ ч},$$

т.е. около 95 лет при непрерывной работе.

Для невосстанавливаемой системы средняя наработка до отказа одного канала равна $T = 10\,000$ ч, а для системы $T_c = (5/6) \cdot T = 8\,333$ ч, т.е. меньше одного года.

Для троированной мажоритарной системы с реконфигурацией имеем

$$T_c = (4/3) T = (4/3) 10\,000 = 13\,300 \text{ ч} = 1,52 \text{ года}.$$

7. ОПЕРАТИВНЫЙ АППАРАТНЫЙ КОНТРОЛЬ ВС

7.1. Задачи и виды оперативного контроля

В резервированных системах отказ основного элемента должен быть быстро обнаружен и локализован, только тогда можно пренебречь временем подключения резерва.

Если система невосстанавливаемая, то оперативный контроль (ОпК) обычно резко сокращает время проявления отказа в системе $\theta_{\text{обн. отк.}}$.

Для восстанавливаемой системы существенно уменьшается среднее время восстановления, которое равно

$$T_{\text{в}} = T_{\text{обн. отк.}} + T_{\text{локал. отк.}} + T_{\text{ремонта}},$$

и, следовательно, увеличивается коэффициент готовности

$$k_{\text{г}} = T_{\text{в}} / (T_0 + T_{\text{в}})$$

и коэффициент оперативной готовности $k_{\text{о.г.}}$.

В общем случае средства ОпК подключаются к функциональному блоку, как показано на рис. 7.1.

Схема ОпК может иметь два выхода: ε — сигнализация об обнаруженном отказе или сбое и α — индикация о локализации отказавшего канала или блока.

Обычно отказ средств ОпК идентифицируется с отказом всей системы, поэтому *схема надежности* системы — последовательная (рис. 7.2). Основные виды оперативного контроля приведены в табл. 7.1.

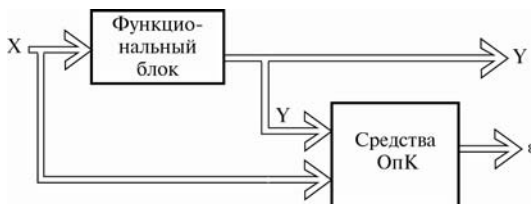


Рис. 7.1



Рис. 7.2

Таблица 7.1

Виды контроля	Формы контроля	Области применения
Контроль операций передачи и хранения	Различные виды помехоустойчивого кодирования и декодирования	СПД, средства хранения информации
Контроль операций передачи	Контроль по четности (по mod 2)	Регистровая логика и средства хранения информации
Контроль резервированных систем	Мажоритарный контроль	Средства ВТ с мажоритарным резервированием
Контроль вычислительных и преобразовательных операций	Контроль по вычетам (по модулю)	Преобразовательная логика

7.2. Мажоритарный контроль

Мажоритарный контроль применяется при мажоритарном резервировании.

Цель контроля состоит в локализации отказавших или сбойных каналов и индикации наличия неисправности. Схемы контроля выявляют m отказавших каналов в системах m/n . На рис. 7.3 изображена система 2/3 с аппаратными средствами контроля. Анализатор А локализует дефектный канал, а индикатор неисправности ИН вырабатывает сигнал ε , что позволяет оперативно заменить неисправное устройство.

Схема оперативного контроля $B = A + \text{ИН}$ состоит из четырех блоков $B_1 + B_2 + B_3 = A$ и $B_4 = \text{ИН}$. Рассмотрим их функции подробнее. В исправной схеме $z_1 = z_2 = z_3 = 1$ или $z_1 = z_2 = z_3 = 0$. При наличии неисправности на выходе одного из каналов соответствующий сигнал будет инвертирован (вместо 1 появится 0 или вместо 0 появится 1).

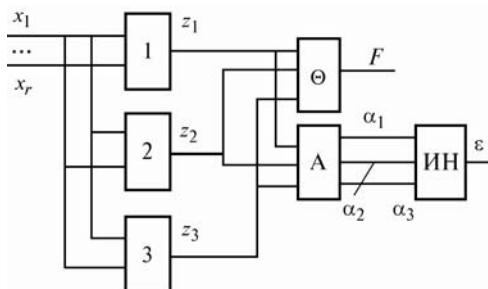


Рис. 7.3

- 1) Блок B_1 локализует появление 0 вместо 1. Для этого в зависимости от номера неисправного канала реализуются функции

$$\alpha'_1 = \bar{z}_1 z_2 z_3,$$

$$\alpha'_2 = z_1 \bar{z}_2 z_3,$$

$$\alpha'_3 = z_1 z_2 \bar{z}_3.$$

- 2) Блок B_2 локализует 1 вместо 0, реализуя функции

$$\alpha''_1 = z_1 \bar{z}_2 \bar{z}_3,$$

$$\alpha''_2 = \bar{z}_1 z_2 \bar{z}_3,$$

$$\alpha''_3 = \bar{z}_1 \bar{z}_2 z_3.$$

- 3) Блок B_3 объединяет эти функции, локализуя канал с инвертированным сигналом

$$\alpha_1 = \alpha'_1 \vee \alpha''_1,$$

$$\alpha_2 = \alpha'_2 \vee \alpha''_2,$$

$$\alpha_3 = \alpha'_3 \vee \alpha''_3.$$

При этом $\alpha_i = 1$ при появлении неисправности в i -м канале.

- 4) Блок $B_4 = \text{ИН}$ — схема индикации реализует функцию

$$\varepsilon = \alpha_1 \vee \alpha_2 \vee \alpha_3.$$

Логическая схема, выполняющая указанные функции, приведена на рис. 7.4.

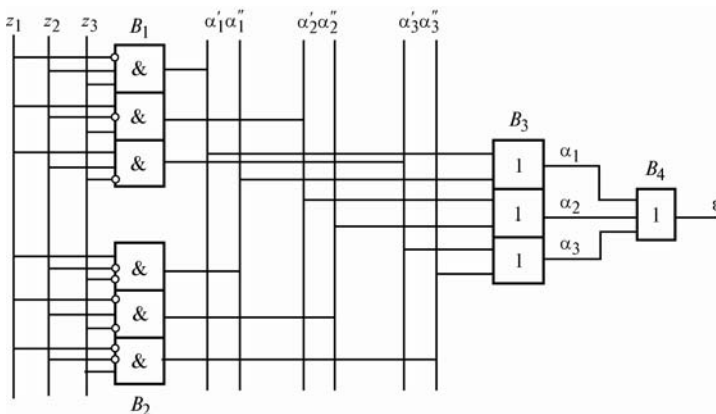


Рис. 7.4

7.3. Вычеты и их свойства

Вычеты применяются для обнаружения ошибок преобразовательных операций (в первую очередь — вычислительных).

Любые данные, содержащиеся в регистре или ячейке памяти, можно интерпретировать как некоторое целое число A . Это число можно представить в виде

$$A = qm + r(A),$$

где $m \geq 2$ — некоторый заданный нами целый делитель числа A , носящий название модуля (всегда положительный); $q = \lfloor A/m \rfloor$ — целое частное (может быть положительным или отрицательным в зависимости от знака числа A); $r(A) = A - \lfloor A/m \rfloor \cdot m$ — остаток (всегда неотрицательный): $0 \leq r(A) \leq m - 1$.

Остаток $r(A)$ носит название вычета или числового вычета числа A по модулю m , т.е. $r(A) = \text{Res } A \bmod m$, $0 \leq r(A) \leq m - 1$.

Два целых числа A и B называются *сравнимыми по модулю m* , что обозначается

$$A \equiv B \bmod m,$$

если их остатки (Res) равны:

$$r(A) = r(B).$$

Примеры вычетов и сравнения чисел по модулю:

- $A = 10, m = 3, q = 3, r(A) = 1$;
- $A = -10, m = 3, q = -4, r(A) = 2$;
- $-22 \equiv +8 \bmod 5$, так как $r(-22) = -22 - \lfloor -22/5 \rfloor \cdot 5 = -22 - (-5) \cdot 5 = +3$ и $r(8) = 8 - \lfloor 8/5 \rfloor \cdot 5 = +3$;
- $22 \equiv 17 \bmod 5$, но эти числа несравнимы между собой по $\bmod 2$.

Свойства вычетов и сравнений:

1) $r(A) \equiv A \bmod m$, так как $A = qm + r(A) \bmod m$ и $r(A) = 0 \cdot m + r(A) \bmod m$, откуда видно, что остатки у A и $r(A)$ одинаковые.

2) $km \equiv 0 \bmod m$ при любом целом k , так как $km = km + 0$ и, в соответствии со свойством 1, $km \equiv 0$.

3) $A \pm km \equiv A \equiv r(A) \equiv r(A) \pm km \bmod m$, что следует из свойств 1 и 2.

4) Если $C = A \pm B$, то $r(C) = r(A) \pm r(B) \bmod m$.

Действительно,

$$C = (q_A \pm q_B) \cdot m + (r(A) \pm r(B)).$$

Но первое слагаемое равно $0 \bmod m$, а второе слагаемое (скобка) равно $r(C)$.

5) Если $C = A \cdot B$ (арифметическое умножение), то $r(C) = r(A)r(B) \bmod m$.
Имеем:

$$\begin{aligned} C &= (q_A m + r(A)) \cdot (q_B m + r(B)) = \\ &= q_A \cdot m \cdot q_B \cdot m + r(A) \cdot q_B \cdot m + r(B) \cdot q_A \cdot m + r(A) \cdot r(B). \end{aligned}$$

Но в соответствии со свойством 2 первые три слагаемых сравнимы с 0 и остается только четвертое слагаемое.

6) Из $A \equiv B$, $C \equiv D$ следует $AC \equiv BD$.

Действительно, $AC \equiv r_A \cdot r_C$, $BD \equiv r_B \cdot r_D$. Но $r_A = r_B$, $r_C = r_D$, значит $r_A \cdot r_C = r_B \cdot r_D$.

7) Из сравнения $A \equiv B \bmod m$ следует сравнение $kA \equiv kB \bmod m$ при любом целом k . Это видно из свойства 6, если заменить C и D на k .

Отметим, что обратное не всегда верно: деление на k может нарушить сравнимость. Аналогично для степеней: из $A^n \equiv B^n$ не всегда следует сравнимость $A \equiv B$.

8) Если $kA \equiv kB \bmod m$ и при этом m и k взаимно просты, то $A \equiv B \bmod m$, т.е. в этом случае сравнимость не нарушается при делении на k . Это достаточное, но не необходимое условие (без доказательства).

9) Если $A \equiv B$, то $A^n \equiv B^n \bmod m$ при любом целом положительном n . Это следует из свойства 6, если перемножить сравнение $A \equiv B$ на $A \equiv B$ нужное число раз.

10) Если $A = a^m$, где a — любое целое число, а m — любое простое число, то $A \equiv a \bmod m$ (теорема Ферма — без доказательства).

11) Если $A = a^{m-1}$, где a — целое число, m — простое число и a не делится на m , то $A \equiv 1 \bmod m$ (следствие теоремы Ферма — без доказательства).

Примеры использования свойств вычетов.

- $48 \equiv 4 \bmod 11 \Rightarrow 12 \equiv 1 \bmod 11$, поскольку $m = 11$ и $k = 4$ взаимно просты (свойство 8).
- $22 \equiv -2 \bmod 8$ (остатки равны 6). Посмотрим, сохранится ли сравнение при делении на 2. Если принять $k = 2$, то $m = 8$ и $k = 2$ не являются взаимно простыми числами и сокращение на 2 нарушает сравнимость: остаток от деления 11 на 8 равен +3, а деление (-1) на 8 дает остаток +7.

- $A = 2^{11} \equiv 2 \pmod{11}$ (согласно свойству 10).
- $A = 2^{11-1} = 2^{10} \equiv 1 \pmod{11}$ (согласно свойству 11).
- Пусть требуется определить, является ли число $N = 341$ составным. Для этого воспользуемся свойством 11: проверим, имеет ли место сравнимость $A = a^{m-1} \equiv 1$, где модуль $m = N = 341$, а в качестве a можно использовать любое целое число, не делящееся на 341. Выберем небольшое число, например, $a = 2$. При этом условие сравнимости принимает вид $A = a^{N-1} \equiv 2^{340} \equiv 1$. Представим число 340 в виде суммы степеней двойки: $N - 1 = 340 = 2^8 + 2^6 + 2^4 + 2^2$. Тогда

$$2^{N-1} = 2^{240} = 2^{256} \cdot 2^{64} \cdot 2^{16} \cdot 2^4.$$

Для упрощения вычислений определим вычеты для каждого из сомножителей по модулю $m = 241$:

- $2^4 = 16 \pmod{341}$.
- $2^{16} = 2^8 \cdot 2^8$. В соответствии со свойством 3, сравнение чисел не изменится, если прибавить или отнять вычет с любым целым коэффициентом k . Поэтому

$$2^8 = 256 \equiv 256 - 341 = -85 \pmod{341}, \text{ а } 2^{16} \equiv (-85)^2 = 7225 \equiv \\ \equiv 7225 - 21 \cdot 341 = 64 \pmod{341},$$

где множитель 21 — это коэффициент k , который подбирается так, чтобы величина $\pm km$ была возможно ближе (сверху или снизу) к значению 7225. Таким образом, имеем $2^{16} \equiv 64 \pmod{341}$.

- $2^{64} = (2^{16})^4 = (2^{16})^2 \cdot (2^{16})^2$. Но $(2^{16})^2 \equiv 64 \cdot 64 \equiv 4096 - 12 \cdot 341 = 4 \pmod{341}$,

где множитель 12 есть целый коэффициент k . Поэтому $2^{64} = 4 \cdot 4 = 16 \pmod{341}$.

- $2^{256} = (2^{64})^4 = (2^{64})^2 \cdot (2^{64})^2$. Но $(2^{64})^2 \equiv 16 \cdot 16 = 256 \equiv 256 - 341 = -85 \pmod{341}$, а $(2^{64})^4 \equiv (-85)^2 = 7225 \equiv 7225 - 21 \cdot 341 = 64 \pmod{341}$.

Следовательно, $2^{256} \equiv 64 \pmod{341}$

- Подставляя полученные вычеты вместо множителей, получим: $2^{N-1} = 2^{240} = 2^{256} \cdot 2^{64} \cdot 2^{16} \cdot 2^4 \equiv 64 \cdot 16 \cdot 64 \cdot 16 = 1024 \cdot 1024$.

Но $1024 \equiv 1024 - 3 \cdot 341 = 1024 - 1023 = 1$. Поэтому $2^{240} \equiv 1$.

Из свойства 11 следует, что в этом случае нельзя утверждать, что число 341 простое, оно может быть как простым, так и составным. В действительности, оно составное: $341 = 11 \cdot 31$. Отметим, что не существует способа определить множители, кроме перебора. Это обстоятельство используется в различных алгоритмах защиты информации.

7.4. Цифровые вычеты

Рассмотренные вычеты являются остатками от деления числа A на m . В этом смысле их называют еще *числовыми вычетами*. На практике при контроле вычислительных операций чаще используются остатки от деления суммы цифр числа A на модуль m . Такие остатки носят название *цифровых вычетов*. Будем обозначать их как $r_{\text{ц}}(A)$ по mod m . Их вычисление требует меньших аппаратных и временных затрат и поэтому более надежно.

Любое неотрицательное число можно представить в виде

$$A = \sum_{i=0}^{l-1} a_i h^i,$$

где l — число разрядов кода с основанием h ; $a_i \in \{0, 1, 2, \dots, h-1\}$ — разрядные коэффициенты; h^i — вес i -го разряда.

Тогда

$$r(A) = r\left(\sum_{i=0}^{l-1} a_i h^i\right) \text{ — числовой вычет,} \quad (7.1)$$

а

$$r_{\text{ц}}(A) = r\left(\sum_{i=0}^{l-1} a_i\right) \text{ — цифровой вычет.} \quad (7.2)$$

В общем случае $r_{\text{ц}}(A) \neq r(A)$ и, следовательно, для цифровых вычетов не выполняются рассмотренные свойства. Однако существует условие, при котором $r_{\text{ц}}(A) = r(A)$.

Теорема 7.1. Если выполняется условие

$$h = km + 1, \quad (7.3)$$

(k — любое целое положительное число), то

$$r_{\text{ц}}(A) = r(A).$$

Действительно, в соответствии со свойствами 4 и 5 имеем

$$r(A) = r\left(\sum_{i=0}^{l-1} a_i h^i\right) = \sum_{i=0}^{l-1} [r(a_i) r(h^i)] = \sum_{i=0}^{l-1} [r(a_i) r^i(h)].$$

Но $r(h) = r(km+1) = 1$ (по свойству 3) и, следовательно, $r^i(h) = 1$. Поэтому

$$r(A) = \sum_{i=0}^{l-1} r(a_i) = r\left(\sum_{i=0}^{l-1} a_i\right) = r_{\Pi}(A).$$

При машинной обработке информации, использующей двоичную арифметику, двоичный код удобно преобразовывать в код с основанием

$$h = 2^s \ (s = 2, 3, 4, \dots), \quad (7.4)$$

т.е. $h = 4, 8, 16, 32, \dots$, а в качестве модуля выбирать значения

$$m = h - 1,$$

хотя возможен любой выбор, удовлетворяющий соотношению (7.3), т.е.

$$m = (h - 1)/k.$$

Пример 7.1. Преобразуем число $A = 359$, выбирая различный модуль m . Пусть $h = 8, m = h - 1 = 7$.

Двоичные числа разбиваются на триады и преобразуются в восьмеричный код:

$$A = 359_{10} = 101\ 100\ 111_2 = 547_8.$$

Затем находится цифровой вычет по mod 7:

$$r_{\Pi}(A) = r(A) = r(5 + 4 + 7) = 16 \equiv 2 \pmod{7}.$$

Определение остатка можно выполнять поэтапно:

$$5 + 4 = 9 \equiv 2 \pmod{7}, \quad 2 + 7 \equiv 2 \pmod{7}.$$

При выборе модуля $m = 3$ двоичное число нужно представить в четверичной системе ($h = 4$), для этого его нужно разбить на диады (по 2 разряда в группе). Тогда

$$A = 359_{10} = 1\ 01\ 10\ 01\ 11_2 = 11213_4$$

и

$$r(A) = r(1 + 1 + 2 + 1 + 3) = 2 \pmod{3}.$$

Для $m = 15$ двоичное число разбивается на тетрады и представляется в 16-ричном коде, затем находится остаток.

Если же выбран модуль $m = 2$, то число необходимо преобразовать в троичный код, что требует более сложной процедуры, поскольку условие (7.4) не выполняется.

7.5. Формирование цифровых вычетов (на примере свертки по mod 3)

Для аппаратного формирования цифровых вычетов обычно применяются простые и надежные комбинационные схемы, одна из которых (для свертки по mod 3) приведена на рис. 7.5. Регистр числа A (один байт) разбивается на два полубайта по 4 разряда в каждом — A_1 и A_2 . Схема свертки состоит из трех одинаковых блоков (1, 2 и 3), на вход которых поступают двоичные сигналы с выходов A_1 и A_2 : $x_0x_1x_2x_3$ и $x_4x_5x_6x_7$. На выходе схемы свертки формируется двоичный код остатка — младший разряд r_0 и старший разряд r_1 . Для получения вычетов в каждый регистр ВС добавляется по 2 разряда, в которые записывается остаток от деления содержимого регистра на m . При модуле 7 требуется 3 дополнительных разряда, при модуле 15 — 4 разряда и т.д.

Для построения каждого блока можно использовать таблицу истинности (ТИ) (табл. 7.2). Поскольку все три блока — 1, 2 и 3 формирователя вычетов одинаковые, то ТИ дана для одного блока (блок 1 для полубайта A_1). В столбце A_1 таблицы перечислены все возможные числа тетрады в десятичном и двоичном виде. В столбце $r(A_1)$ даны значения остатков $r(A_1)$ по модулю 3 в троичном и двоичном кодах (r_1 — старший разряд остатка; r_0 — младший).

Пример для $m = 3$:

$$A = 9_{10} = 1001_2, \quad r = 2 + 1 \equiv 0_3 = 00_2 \pmod{3}.$$

$$A = 10_{10} = 1010_2, \quad r = 2 + 2 \equiv 1_3 = 01_2 \pmod{3}.$$

Двоичные коды A_1 и разряды r_0 и r_1 остатков образуют две таблицы истинности. На основании данных ТИ составляются логические функции $r_0(x)$ и $r_1(x)$ в виде совершенной дизъюнктивной формы (СДНФ).

$$r_1(x) = \bar{x}_3 \bar{x}_2 x_1 \bar{x}_0 \vee \bar{x}_3 x_2 \bar{x}_1 x_0 \vee x_3 \bar{x}_2 \bar{x}_1 \bar{x}_0 \vee x_3 \bar{x}_2 x_1 x_0 \vee x_3 x_2 x_1 \bar{x}_0.$$

Аналогичная функция составляется для $r_0(x)$. Значения r_0 и r_1 приведены в таблице. Если возможно, выполняется минимизация логических функций. По этим функциям составляются функциональные и принципиальные схемы каждого блока свертки.

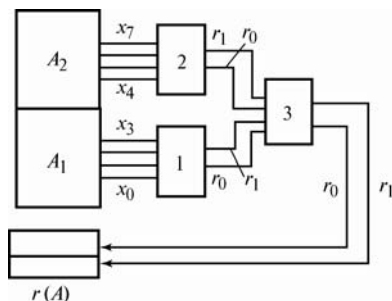


Рис. 7.5

Таблица 7.2

Совмещенные ТИ для $r_1(x)$ и $r_0(x)$

A_1			$r(A_1) \bmod 3$		
Десятичный код	$x_3 x_2$	$x_1 x_0$	Троичный код	$r_1(x)$	$r_0(x)$
0	0 0	0 0	0	0	0
1	0 0	0 1	1	0	1
2	0 0	1 0	2	1	0
3	0 0	1 1	0	0	0
4	0 1	0 0	1	0	1
5	0 1	0 1	2	1	0
6	0 1	1 0	0	0	0
7	0 1	1 1	1	0	1
8	1 0	0 0	2	1	0
9	1 0	0 1	0	0	0
10	1 0	1 0	1	0	1
11	1 0	1 1	2	1	0
12	1 1	0 0	0	0	0
13	1 1	0 1	1	0	1
14	1 1	1 0	2	1	0
15	1 1	1 1	0	0	0

7.6. Контроль по вычетам

7.6.1. Контроль арифметических операций $C = A*B$,

где $*$ $\in \{+, -, \times\}$

Если $C = A*B$, то выполняются следующие действия.

- Вычисляются $r(A)$ и $r(B) \bmod m$.
- Вычисляется $A*B = C'$ (возможно с ошибкой).
- Вычисляется $r(C') \bmod m$.

- Независимо вычисляется $r(A) * r(B) = r'(C) \bmod m$ (штрих указывает на возможность ошибки).
- Выполняется сравнение $r'(C)$ и $r(C')$. Если эти значения равны, то считается, что ошибка отсутствует. В противном случае вычисления повторяются.

Если выполняется арифметическая операция умножения $C = AB$ и результат округляется с обнулением s последних разрядов, то округленное число

$$C_1 = AB - E,$$

где E — число, записанное в s округляемых разрядах. В этом случае при отсутствии ошибок вычисления $r(C_1) = r(A) \cdot r(B) - r(E)$, а при проверке находится C'_1 , затем $r(C'_1)$ сравнивается с $r(A) \cdot r(B) - r(E) = r'(C_1)$.

7.6.2. Контроль арифметических операций деления

Контроль операции $C = A/B$ осуществляется косвенным путем. Представим делимое как

$$A = BC = QB + R,$$

где $Q = \lfloor A/B \rfloor = \lfloor C \rfloor$ — целая часть частного, а $R = A - \lfloor A/B \rfloor \cdot B = A - QB$ — целый неотрицательный остаток. Тогда

$$r(A) = r(B) \cdot r(Q) + r(R) \bmod m.$$

При контроле выполняются следующие операции.

- Рассчитываются $r(A)$ и $r(B)$.
- Определяется частное $A/B = C'$ и целое частное Q' (может быть с ошибкой).
- Выделяется целый остаток от деления A на B : $R' = A - Q'B$.
- Находится $r'(A) = r(B) \cdot r(Q') + r(R') \bmod m$.
- Выполняется сравнение $r'(A)$ и $r(A)$. При равенстве считается, что деление выполнено без ошибки.

Отметим, что при контроле деления ошибка обнаруживается, только если неправильно вычислены как частное Q , так и остаток R .

7.6.3. Контроль логических операций $C = A * B$,

$$\text{где } * \in \{ \&, \vee, \oplus \}$$

Поскольку логические операции выполняются над двоичными числами, то их можно арифметизировать, используя следующие фор-

мулы (справедливость которых легко проверяется подстановкой вместо переменных значений 0 или 1):

$$A \& B = A \cdot B,$$

$$A \vee B = A + B - AB,$$

$$A \oplus B = A + B - 2AB.$$

Из первых двух равенств следует, что

$$(A \& B) + (A \vee B) = A + B.$$

Поэтому

$$C = A \& B = A + B - A \vee B,$$

$$C = A \vee B = A + B - A \& B,$$

$$C = A \oplus B = A + B - 2(A \& B).$$

Отсюда

$$r(A \& B) = r(A) + r(B) - r(A \vee B),$$

$$r(A \vee B) = r(A) + r(B) - r(A \& B),$$

$$r(A \oplus B) = r(A) + r(B) - 2r(A \& B).$$

Контроль осуществляют в следующем порядке.

- Выполняется логическая операция $A * B = C'$.
- Вычисляется $r(C')$.
- Вычисляется $D = A \& B$ — для операции логического сложения, $D = 2 A \& B$ — для операции сложения по модулю 2 или $D = A \vee B$ — для операции логического умножения.
- Находится остаток $r(D)$.
- Выполняется суммирование $r(A) + r(B) - r(D) = r'(C)$.
- Сравниваются полученные значения $r'(C)$ и $r(C')$.

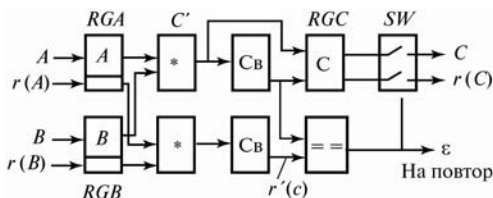
7.6.4. Контроль операций пересылок

($RG \rightarrow RG, M \rightarrow M, RG \rightarrow M, M \rightarrow RG$)

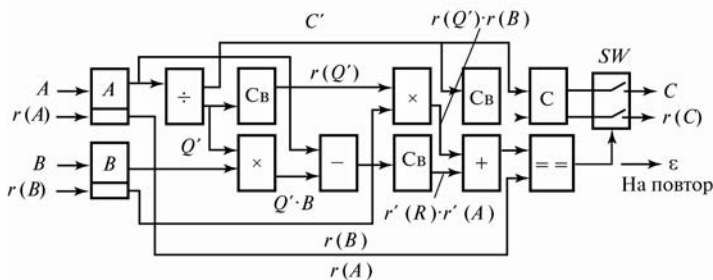
Контроль пересылок осуществляется по цифровым вычетам, которые, однако, не совпадают с числовыми, поскольку модуль ($m = 2$) совпадает с основанием системы счисления ($h = 2$) и условие (7.3) не выполняется. Но для пересылок данных такого совпадения не требуется, так как вычисления не производятся. Схема свертки по модулю 2 формирует цифровой вычет принятой кодовой комбинации, который сравнивается с содержимым контрольного разряда регистра приемной стороны. В случае несовпадения формируется сигнал ошибки. В процессе контроля обнаруживаются наиболее вероятные одиночные ошибки пересылок, а также ошибки нечетной кратности.

7.7. Аппаратная реализация цифрового контроля по вычетам

На рис. 7.6 приведены схемы оперативного аппаратного контроля по цифровым вычетам (цифрового контроля) арифметических и логических операций, а также операций пересылок. Порядок обработки данных встроенными средствами соответствует описанным ранее процедурам. Для сокращения аппаратной избыточности обычно применяется контроль по модулю 3 или 2 (для операций пересылок). Рассмотрим контроль операции пересылок (рис. 7.6, *з*). Схемой свертки является сумматор $M2$ (исключающее ИЛИ). На передающей стороне регистр RGA содержит вектор A и дополнительный разряд четности, в который заносится 0 или 1, в зависимости от четности числа единиц в A . Содержимое RGA (включая разряд четности) пересылается в регистр RGB , который также содержит разряд четности. При передаче может возникнуть одиночная ошибка либо в разрядах A , либо и разряде четности. Принятый вектор B свертывается по модулю 2. В случае ошибки результат свертки будет отличаться от содержимого разряда четности. Это обнаруживается компаратором, формирующим сигнал на повтор передачи.

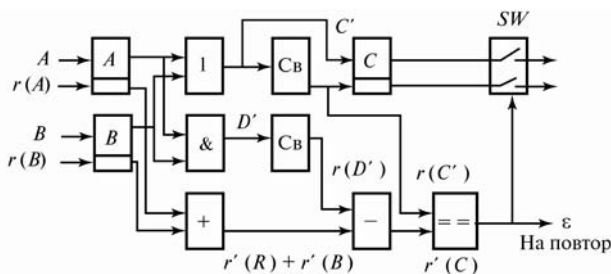


а) схема контроля арифметических операций $*$ = (+, −, ×)

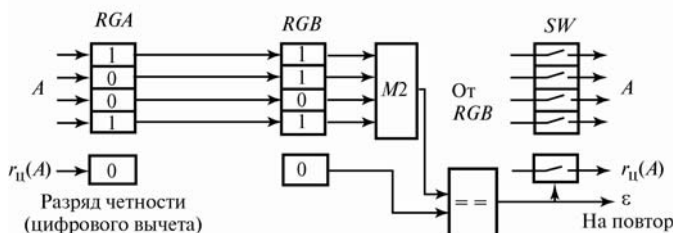


б) схема контроля арифметической операции деления

Рис. 7.6 (Начало)



в) схема контроля логической операции «ИЛИ»



з) схема цифрового контроля операций пересылок по модулю 2 (ошибки нечетной кратности)

Рис. 7.6. (Окончание)

На рисунке приняты следующие обозначения:

- в схемах контроля вычислительных операций: $Cв$ — схема свертки по модулю m ; A и B — регистры операндов, C — регистр результата; SW — ключ, управляемый компаратором, выводящий содержимое регистра C и остаток $r(C)$ для использования в других функциональных блоках BC ; при отрицательном результате сравнения данных, поступающих на компаратор, выдается сигнал ε на повторение вычислений.

- в схеме контроля операций пересылок: RGA и RGB — регистры на передающей и приемной стороне; $M2$ — схема свертки по модулю 2.

8. ЗАДАЧИ И ВИДЫ ТЕХНИЧЕСКОЙ ДИАГНОСТИКИ

8.1. Формы технической диагностики

С точки зрения надежности *состояние объекта диагностирования (ОД)* — это элементарное событие (неопределяемое понятие). Все состояния — несовместимые события. Например, кратная неисправность — это одно из состояний. Состояния делятся на исправное, неисправное, работоспособное и неработоспособное.

Введем обозначения:

- * — исправное состояние,
- — неисправное состояние,
- — неработоспособное состояние.

На рис. 8.1 приведены три формы диагностики (по стандартам ЕСКД) — контроль исправности, контроль работоспособности и локализация дефектов. Контурами обозначены элементарные проверки π (тесты), которые охватывают неисправные (неработоспособные) состояния, но никогда не охватывают исправное состояние. Если состояние находится внутри контура, результатом проверки будет результат «Негоден» (НГ), если — вне контура, будет получен результат «Годен» (Г). Под *проверкой (тестом) T* понимается совокупность элементарных проверок $\{\pi_i\}$.

Пусть при локализации дефектов результаты трех элементарных проверок получили значения

$$\text{Рез. } T = (\text{НГ}, \text{НГ}, \text{Г}).$$

Тогда имеет место состояние a_2 . При результате

$$\text{Рез. } T = (\text{Г}, \text{Г}, \text{Г})$$

имеет место одно из состояний, находящихся вне контура (в данном случае — это исправное состояние a_0).

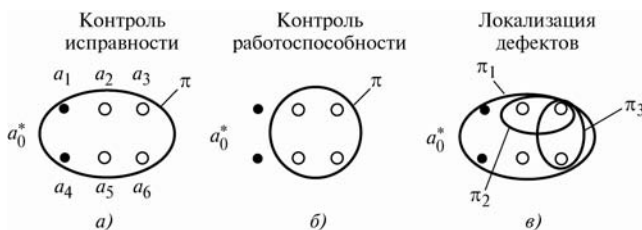


Рис. 8.1

Назовем областью локализации состояний любую область пересечения отдельных контуров, а также внешнюю область. На рис. 8.1, в таких областей 5:

$$\{a_1, a_4, a_5\}, \{a_2\}, \{a_3\}, \{a_6\} \text{ и } \{a_0\}.$$

Все состояния могут быть локализованы, если каждая область содержит только по одному состоянию. Если в области — более одного состояния, то они называются *эквивалентными* и неразличимы выбранной системой проверок. В примере множество $\{a_1, a_4, a_5\}$ образует *класс эквивалентных состояний*.

8.2. Показатели качества диагностирования

1. Глубина диагностирования $k_{ГД}$. Используется при локализации дефектов.

а) С точностью до класса эквивалентных состояний.

Обозначим: N — число классов, различимых данной системой проверок $T = \{\pi_1, \pi_2, \dots, \pi_m\}$, N_0 — максимальное число классов (областей локализации) на принятом уровне деления. Тогда

$$k_{ГД} = N / N_0, 0 \leq k_{ГД} \leq 1.$$

Пусть для рис. 8.1, в принятый уровень деления — «с точностью до класса эквивалентных состояний». Тогда $N_0 = 7$, $N = 5$ и $k_{ГД} = 5/7$.

б) С точностью до функционального блока.

в) С точностью до конструктивного блока.

2. Полнота проверки $k_{ПП}$. Используется при контроле исправности или работоспособности.

Если N — число охватываемых данной системой проверок неисправных состояний (или неисправных функциональных или конструктивных блоков), а N_0 — общее число неисправных состояний (или блоков) на принятом уровне деления, то

$$k_{ПП} = N / N_0, 0 \leq k_{ПП} \leq 1.$$

Другое определение связано с интенсивностью отказов. Если Λ — суммарная интенсивность отказов блоков, охватываемых проверками, а Λ_0 — интенсивность отказов всего изделия, то

$$k_{ПП} = \Lambda / \Lambda_0, 0 \leq k_{ПП} \leq 1.$$

Достоинство данного определения состоит в независимости от принятого уровня деления.

3. Достоверность результата диагностирования:

$$P_{\text{достовер}} = P[\text{правильного заключения о состоянии объекта}].$$

4. Эффективность диагностирования:

а) Среднее число элементарных проверок при определении состояния ОД (средняя длина проверки)

$$l_{\text{ср}} = \sum_i p(a_i) l_i.$$

б) Среднее время диагностирования

$$t_{\text{ср}} = \sum_i p(a_i) t_i.$$

в) Средняя стоимость диагностирования.

$$C_{\text{ср}} = \sum_i p(a_i) C_i.$$

8.3. Понятие состояния объекта диагностирования. Модели неисправностей

Неисправности функциональных или конструктивных блоков могут проявляться по-разному: это может быть изменение функции передачи блока, изменение логического состояния в какой-либо цепи схемы и т.д. Поэтому введем понятие «узла».

Под узлом будем понимать:

- функциональный блок,
- конструктивный блок,
- электрическую цепь схемы между блоками, а также между точкой разветвления цепи и соответствующим блоком,
- входную или выходную цепь.

Состояние ОД — это событие, состоящее в неисправности одного или нескольких узлов на принятом уровне деления или исправности всего ОД.

Обозначим через A_i — состояние исправности узла A_i , — состояние неисправности узла A_i . Состояние a_i ОД отображается конъюнкцией этих переменных. Пусть (для определенности) число узлов $n = 3$. В табл. 8.1 приведены возможные состояния устройства. В правой части таблицы состояния объединены в множества I , J и I_0 . Общее число состояний

$$N = |I| + |J| + |I_0| = 2^n.$$

Таблица 8.1

События	Состояния ОД	Множество состояний M
$\bar{A}_1 A_2 A_3$	a_1	I $(I = n)$
$A_1 \bar{A}_2 A_3$	a_2	
$A_1 A_2 \bar{A}_3$	a_3	
$\bar{A}_1 \bar{A}_2 A_3$	a_4	J $(J = N - n - 1)$
.....	
$\bar{A}_1 \bar{A}_2 \bar{A}_3$	a_7	
$A_1 A_2 A_3$	a_0	$I_0 = \{a_0\} (I_0 = 1)$

В зависимости от используемых в расчетах состояний рассматриваются следующие модели неисправностей.

- Модель кратных зависимых неисправностей. Для нее используется множество $M = I \cup J \cup I_0$.

Вероятности состояний рассчитываются по формулам:

$$p(a_1) = p(\bar{A}_1) p(A_2 / \bar{A}_1) p(A_3 / \bar{A}_1, A_2),$$

$$p(a_2) = p(A_1) p(\bar{A}_2 / A_1) p(A_3 / A_1, \bar{A}_2),$$

.....

$$p(a_7) = p(\bar{A}_1) p(\bar{A}_2 / \bar{A}_1) p(\bar{A}_3 / \bar{A}_1, \bar{A}_2),$$

$$p(a_0) = p(A_1) p(A_2 / A_1) p(A_3 / A_1, A_2),$$

$$\sum p(a_i) = 1.$$

Поскольку условные вероятности обычно не известны, данная модель практически не применяется.

- Модель кратных независимых неисправностей. Множество состояний – то же, т.е. ($M = I \cup J \cup I_0$), но все вероятности состояний – безусловные:

$$p(a_1) = p(\bar{A}_1) p(A_2) p(A_3),$$

$$p(a_2) = \dots\dots,$$

$$p(a_0) = \dots\dots,$$

$$\sum p(a_i) = 1.$$

Значения вероятностей $p(A_k)$ можно рассчитать по λ -характеристикам функциональных блоков.

- Модель одиночных неисправностей. Эта модель в расчетах используется чаще всего в силу их распространенности. Кроме того, тесты, ориентированные на покрытие одиночных неисправностей, обычно покрывают большую часть кратных неисправностей. Множество состояний при этом составляет $M = I \cup I_0$, а вероятности состояний рассчитываются по формулам:

$$p(a_i) = Q_i / \left(1 + \sum_{k=1}^n Q_k \right), \quad i \neq 0; \quad p(a_0) = 1 / \left(1 + \sum_{k=1}^n Q_k \right),$$

где

$$Q_k = p(\bar{A}_k) / (1 - p(\bar{A}_k)) = q_k / (1 - q_k) = q_k / p_k, \\ q_k = P[\bar{A}_k = 1], \quad p_k = 1 - q_k.$$

При этом $\sum_{k=0}^n p(a_i) = 1$.

Для высоконадежных систем можно пользоваться приближенными выражениями:

$$p(a_i) \approx p(\bar{A}_k), \quad i \neq 0; \\ p(a_0) \approx 1 - \sum_{a_i \in I} p(a_i).$$

Отметим, что в последней модели неисправности являются зависимыми событиями в силу их несовместимости, поскольку появление какой-либо неисправности предполагает отсутствие другой, т.е. изменяет вероятность появления последней. Поэтому данная модель является гипотетической и в большинстве случаев нереальной. Однако ее широкое применение обусловлено простотой расчетов и для высоконадежных систем приводит к незначительным ошибкам.

8.4. Классы неисправностей цифровых устройств

Функциональные неисправности — искажение логики работы цифрового устройства, связанное с функциональным описанием ОД. Для обнаружения неисправностей используются функциональные тесты, рассматривающие ОД как «черный ящик» (т.е. структура объекта неизвестна или слишком сложна, чтобы ее рассматривать).

Такие тесты применяются при моделировании сложных цифровых устройств, описываемых на автоматном уровне с известной функциональной зависимостью выхода объекта от его входа и состояния.

Логические неисправности — искажение логики цифровых устройств, связанное со структурным описанием ОД. Для их обнаружения используются структурные тесты, рассматривающие ОД как «белый ящик» (т.е. известна функциональная схема объекта).

Такие тесты применяются при моделировании комбинационной логики с известными логическими зависимостями между внутренними элементами схемы устройства.

Структурные тесты намного более эффективны по времени диагностирования, но их применение далеко не всегда возможно. Кроме указанных классов неисправностей существуют и другие, связанные с технологическими особенностями схем, степенью интеграции и другими факторами. Но их мы рассматривать не будем. Например, некоторые неисправности могут превратить комбинационную схему в автомат. Но мы ограничимся только перечисленными классами неисправностей.

Классы логических неисправностей:

а) Константные логические неисправности — это неисправности, при которых один из входов или выходов логических элементов принимают логические значения «Тождественный 0» ($\equiv 0$) или «Тождественная 1» ($\equiv 1$). При этом физическая природа появления таких логических значений может быть различной (например, заземление какой-либо цепи или ее обрыв). Такие неисправности весьма распространены, особенно в схемах с TTL-технологией.

Приведем пример логических константных неисправностей. На рис. 8.2 приведена схема ЗИ-НЕ. В исправной схеме (рис. 8.2, а) при указанных значениях входных переменных ток I имеет большую величину, а I_0 близок к 0, транзистор T_1 открыт, T_2 закрыт и выход $F = 1$. Если же все входные переменные равны 1, то ток $I = 0$, а ток I_0 открывает транзистор T_2 , при этом $F = 0$. Исправная схема реализует функцию $F = x_1 x_2 x_3$. На рис. 8.2, б и в крестик обозначает обрыв в цепи.

Обрыв в цепи эмиттера x_1 транзистора T_1 приводит к константной неисправности типа $x_1^* \equiv 1$ на входе устройства (звездочка при переменной служит указателем на неисправность в данной цепи). При обрыве в цепи коллектора T_1 второй транзистор T_2 будет постоянно закрыт, что эквивалентно неисправности $F^* \equiv 1$.

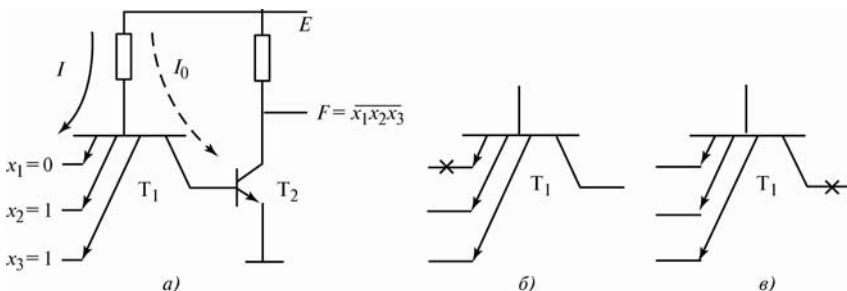


Рис. 8.2

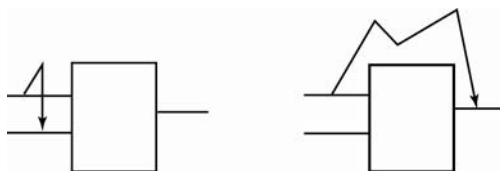


Рис. 8.3

б) Константные логические неисправности типа «Короткое замыкание» (к.з) — это неисправности, при которых некоторые цепи принимают одинаковые логические значения (рис. 8.3). Такие неисправности часто встречаются в схемах с МОП-технологией.

в) Логические неисправности типа «Перепутывание» — это неисправности, связанные с неверными соединениями цепей в схеме. Такие неисправности характерны для стадий проектирования и изготовления микросхем.

8.5. Задачи и классификация систем тестового диагностирования

Системы тестового диагностирования (СТД) — это совокупность ОД и средств диагностирования (СрД) (рис. 8.4, а).

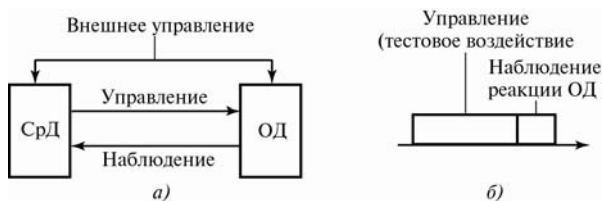


Рис. 8.4

В свою очередь, каждый тест имеет две фазы, представленные на рис. 8.4, б.

Задачи технической диагностики:

- исследование и препарирование ОД с целью повышения тестопригодности и контролепригодности;
- выбор метода тестирования и синтез тестов (если это необходимо);
- выбор метода сжатия тестовой информации (если это необходимо);
- выбор и построение алгоритмов диагностирования.

Классификация СТД:

- по уровню диагностирования:
 - а) диагностирование ВС на макроуровне,
 - б) диагностирование на микроуровне;
- по характеру тестирования:
 - а) СТД с применением регулярного тестирования:
 - на основе синтезированных тестов,
 - на основе исчерпывающего тестирования;
 - б) СТД с применением статистического тестирования:
 - на основе случайных тестов,
 - на основе псевдослучайных (ПС) тестов;
- по характеру анализа реакций:
 - а) СТД с непрерывным наблюдением реакций,
 - б) СТД с сигнатурным анализом;
- по виду цифрового ОД:
 - а) СТД с объектом комбинаторного типа,
 - б) СТД с объектом последовательностного типа.

8.6. Уровни диагностирования

Диагностика ВС осуществляется поэтапно на нескольких уровнях.

1) Диагностирование активных модулей на макроуровне. Активным называется устройство, обладающее диагностическими возможностями, т.е. имеющее в своем составе средства генерации тестов, анализа реакций и средства коммуникации тестовых сигналов.

Для диагностики используются как внешние активные средства, так и внутренние активные модули ВС, например процессоры.

2) Диагностирование пассивных модулей с помощью проверенных внутренних активных средств (ОЗУ, УВВ, модемы и т.д.). Точность — несколько функциональных блоков.

3) Диагностирование на микроуровне. Используются как внешние автономные средства диагностирования (автоматы проверки), так и внутренние ресурсы (микродиагностические программы). Точность — до отдельной микросхемы.

На 1-м и 2-м уровнях применяются функциональные тесты, на 3-м уровне — как функциональные, так и структурные тесты.

9. ОРГАНИЗАЦИЯ ДИАГНОСТИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА МАКРОУРОВНЕ

9.1. Метод раскрутки

Сначала дадим некоторые определения.

- *Диагностическое ядро* — работоспособная часть программно-аппаратных средств, обладающая диагностическими возможностями.
- *Начальное ядро* — работоспособная часть системы, начинающая тестирование.
- *Расширенное ядро* — работоспособная часть системы, содержащая начальное ядро и уже проверенные блоки.
- *Раскрутка* — способ диагностирования системы, состоящий в расширении ядра до полной проверки всех блоков (рис. 9.1).

Виды диагностического ядра:

- центральное начальное ядро (априорно известно). Обычно это внешние ресурсы, используемые для проверки некоторых функциональных блоков;
- распределенное начальное ядро (заранее неизвестное, но проявляющееся в процессе начального этапа диагностики). Этот вид ядра имеет место при самодиагностике многопроцессорных систем при взаимной проверке активных модулей системы. В процессе такой проверки выявляется начальное ядро, от которого начинается раскрутка.

Для раскрутки необходим диагностический интерфейс, содержащий магистраль управления (шина адреса и шина управления), магистраль доставки тестовых воздействий (шина данных) и магистраль транспортировки реакций (тоже шина данных).

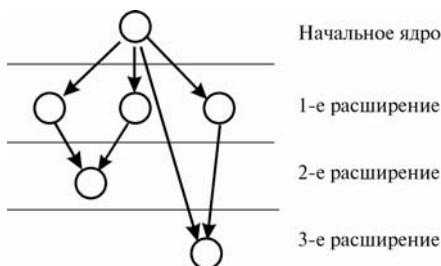


Рис. 9.1

9.2. Диагностическая модель многопроцессорной ВС с распределенным ядром

9.2.1. Графовая модель

Диагностическая модель представляется *помеченным ориентированным графом* $G(V, E, A)$, где $V = \{v_i\}$ — множество вершин графа, отображающих активные модули системы; $E = \{e_{ij}\}$ — множество дуг, отображающих тестовые связи; $A = \{a_{ij}\}$ — множество симптомов (синдром). Симптом a_{ij} может принимать одно из двух значений: $a_{ij} = 0$, если при проверке модулем v_i модуля v_j получен результат «Годеен», и $a_{ij} = 1$, если получен результат «Негодеен». Как уже указывалось выше, под *активным модулем* понимается совокупность программно-аппаратных средств, обладающих диагностирующими возможностями, т.е. совокупность, содержащая средства генерации тестов, средства анализа реакций проверяемого модуля и средства передачи тестовых сигналов и приема реакций (т.е. порты ввода/вывода). Такими средствами обладают процессоры, но не только они. Пассивные средства тестируются активными модулями в процессе раскрутки.

Модели симптомов. Пусть модуль v_i тестирует модуль v_j (рис. 9.2). Наличие тестовой связи e_{ij} означает, что v_i генерирует тесты, пересылает их в v_j , анализирует реакции и формирует симптом a_{ij} .

Введем обозначения: G — множество исправных модулей; F — множество неисправных модулей; $\{G, F\}$ носит название *отказовой ситуации*.

При анализе систем с распределенным или с центральным ядром применяют различные модели симптомов a_{ij} , три из которых представлены в таблицах на рис. 9.3. Нулем в них обозначен результат тестирования (симптом) «Годеен», единицей — результат «Негодеен», а звездочка означает, что результат может быть либо ноль, либо единица независимо от состояния тестируемого модуля. Множество симптомов в графовой модели называется синдромом.

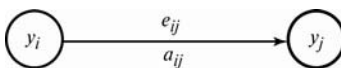


Рис. 9.2

v_i	v_j	
	G	F
G	0	1
F	1	1

а)

v_i	v_j	
	G	F
G	0	1
F	*	1

б)

v_i	v_j	
	G	F
G	0	1
F	*	*

в)

Рис. 9.3

Первая модель (рис. 9.3, а) используется в том случае, если при тестировании неисправным модулем v_i вероятность получения результата «Годеи» очень мала, и ею можно пренебречь. Такая модель применяется в системах с центральным ядром, например в однопроцессорных ПЭВМ.

Вторая модель (рис. 9.3, б) частично снимает это ограничение: при тестировании неисправным модулем v_i модуля v_j , который исправен: $a_{ij} = * \in \{0,1\}$. Если же $v_i \in F$, то $a_{ij} = 1$. Такая модель носит название БГМ-модели — по имени авторов (Barsi, Grandoni, Maestrini).

Третья модель (рис. 9.3, в) полностью снимает указанное ограничение. Она называется ПМЧ-моделью (Preparata, Metze, Chien). В дальнейшем рассматривается только ПМЧ-модель как наиболее универсальная. Вторая и третья модели применяются в многопроцессорных системах с распределенным ядром.

Ограничения ПМЧ-модели:

- Особенностью ПМЧ-модели является требование полноты проверок: при $v_i \in G$ обязательно должен получиться правильный результат. Но при $v_i \in F$ результат ненадежен.
- Каналы передачи тестов и реакций, предполагаются безотказными.

Система называется *t-диагностируемой*, если при наличии $|F| \leq t$ неисправных модулей любой допустимый синдром может быть однозначно и правильно расшифрован, т.е. правильно определена отказовая ситуация.

Показателем диагностируемости системы называется величина $\tau = \max t$.

Таким образом, при $|F| \leq \tau$ ВС является *t-диагностируемой*, а при $|F| > \tau$ не является, т.е. расшифровка синдрома не дает однозначного результата при $t = |F|$.

Задачей анализа ПМЧ-модели является определение τ для известной графовой модели (знание синдрома при этом не требуется). После выполнения анализа решается вторая задача — расшифровка синдрома

ма, полученного в результате взаимных проверок активных модулей ВС, т.е. осуществляется диагноз системы. Третьей задачей является синтез t -диагностируемой системы, под которым будем понимать реконструкцию модели, которая приводит к увеличению τ путем введения минимального числа дополнительных тестовых связей без увеличения числа активных модулей, и устранение избыточных связей.

Частные случаи использования ПМЧ-модели. Рассмотрим три частных случая графовой модели системы.

- 1) При наличии одной вершины в графе (одного процессора) (рис. 9.4, а) имеем единственную тестовую связь и единственный симптом $A = \{a_{11}\}$:

- а) положим $t = 0$. Тогда симптом $a_{11} = 1$ невозможен и $a_{11} = 0$ расшифровывается однозначно;
- б) положим $t = 1$. Тогда $a_{11} = 1$ расшифровывается однозначно, но при $a_{11} = 0$ расшифровка неоднозначна: возможны ситуации $v_1 \in G$ и $v_1 \in F$. Поэтому $t \neq 1$ и $\tau = 0$.

- 2) Рассмотрим случай двух процессоров, проверяющих друг друга (рис. 9.4, б):

- а) при $t = 0$ возможен только синдром $A = \{a_{12} = 0, a_{21} = 0\}$, при этом расшифровка выполняется однозначно;
- б) при $t = 1$ существует синдром $A = \{11\}$, который возможен при двух отказовых ситуациях $\{v_1 \in G, v_2 \in F\}$ и $\{v_1 \in F, v_2 \in G\}$. Поэтому $\tau = 0$.

- 3) Граф с тремя вершинами и тремя тестовыми связями, изображен на рис. 9.4, в:

- а) при $t = 0$ возможен только синдром $A = \{0\ 0\ 0\}$, который расшифровывается однозначно;
- б) положим $t = 1$. Тогда возможны четыре синдрома:
 $A = \{0\ 0\ 0\}$, $A = \{0\ 1\ * \}$, $A = \{* \ 0\ 1\}$ и $A = \{1\ * \ 0\}$,

где звездочка означает любое значение: 0 или 1. Первый синдром расшифровывается однозначно, а остальные синдромы, представляющие собой циклические перестановки символов, также однозначно указывают на отказавшую вершину;

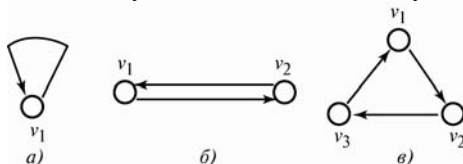


Рис. 9.4

в) при $t = 2$ возможны синдромы, которые нельзя однозначно расшифровать. Например, при синдроме $A = \{ 1 \ 1 \ 1 \}$ неизвестно, какая пара вершин неисправна. Поэтому

$$\tau = \max (t = 0, t = 1) = 1.$$

Вывод. Взаимная диагностика на основе ПМЧ-модели возможна только при наличии не менее трех процессоров.

9.2.2. Отображение физической структуры объекта в диагностическую модель

Физические связи в системе не всегда совпадают со структурой тестовых связей модели. Например, вычислительная сеть с 5 компьютерами, соединенными по структуре «общая шина», отображается полным графом $G(V, E)$, содержащим 5 вершин и 10 дуг (5 пар двусторонних дуг). Процесс отображения физической структуры объекта в диагностическую модель представляет собой самостоятельную задачу.

Анализ ПМЧ-модели. Цель анализа состоит в определении значения τ для заданной графовой модели. Для этого вводится понятие p -разбиения графа; p -разбиением называется разбиение графовой модели на три подмножества вершин X , Y и Z , удовлетворяющих следующим требованиям:

- 1) X, Y, Z не пересекаются, т.е. не имеют общих вершин;
- 2) $Z \neq \emptyset$;
- 3) все дуги, исходящие из X , входят только в Y : $\forall v_i \in X; \Gamma^+(v_i) \in Y$, если только $\Gamma^+(v_i) \notin X$. Здесь $\Gamma^+(v_i)$ — множество непосредственных последователей вершины v_i .

Вид p -разбиения приведен на рис. 9.5 (каждая стрелка отображает некоторое число дуг).

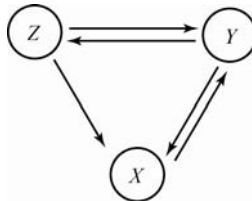


Рис. 9.5

Показателем p -разбиения (где p — номер разбиения) называется коэффициент

$$k(p) = |Y| + \lceil |Z| / 2 \rceil,$$

где скобки $\lceil \rceil$ означают целую верхнюю грань заключенного в них выражения.

Обозначим через P — общее число p -разбиений. Можно показать, что

$$|P| \leq 2^n - 1,$$

где n — число вершин графа.

Теорема 9.1. Если граф $G(V, E)$ не содержит попарно тестирующих модулей, то необходимым и достаточным условием t -диагностируемости системы является выполнение неравенств:

$$\bullet n \geq 2t + 1; \tag{9.1}$$

$$\bullet \forall v_i: |\Gamma^-(v_i)| \geq t, \tag{9.2}$$

где n — число вершин графа; $\Gamma^-(v_i)$ — множество непосредственных предшественников вершины v_i , т.е. множество дуг, входящих в v_i ; $|\Gamma^-(v_i)|$ — мощность этого множества.

Обозначим через τ_1 максимальное значение t , при котором выполняется условие (9.1), и через τ_2 — максимальное значение t , удовлетворяющее всем условиям (9.2). Тогда

$$\tau = \min(\tau_1, \tau_2).$$

Теорема 9.2. Для любого графа (содержащего или не содержащего попарно взаимно тестирующие модули, сильносвязного или несильносвязного) необходимым и достаточным условием t -диагностируемости системы является одновременное выполнение следующих трех условий:

$$\begin{aligned} &\bullet \text{условие (9.1),} \\ &\bullet \text{условие (9.2),} \\ &\bullet \text{условие: } \forall p : k(p) \geq t + 1, \end{aligned} \tag{9.3}$$

где число p -разбиений равно $|P|$.

Условие (9.3) должно выполняться для всех p -разбиений, т.е. для любого $k(p)$, включая разбиение с $\min k(p)$. Обозначим $\tau_3 = \max t$, при котором удовлетворяется это требование. Тогда для каждого отдельного p -разбиения

$$t \leq \min k(p) - 1,$$

а для всех разбиений в совокупности

$$\tau_3 = \max_t \min_{p \in P} k(p) - 1 = \min_{p \in P} k(p) - 1.$$

Показатель диагностируемости системы определяется по формуле

$$\tau = \min (\tau_1, \tau_2, \tau_3).$$

Примеры p -разбиений.

Для графа, изображенного на рис. 9.6, имеем

- 1) $X = \{1\}, Y = \{2\}, Z = \{3, 4, 5\}$ — является p -разбиением;
- 2) $X = \{5\}, Y = \{\emptyset\}, Z = \{1, 2, 3, 4\}$ — является;
- 3) $X = \{\emptyset\}, Y = \{\emptyset\}, Z = \{1, 2, 3, 4, 5\}$ — является,
- 4) $X = \{2\}, Y = \{1, 3, 4, 5\}, Z = \{\emptyset\}$ не является, так как $Z = \{\emptyset\}$.

Общее число p -разбиений с n вершинами $|P| \leq 2^n - 1$.

Случай сильносвязного графа. Граф называется сильносвязным, если любая его вершина достижима из любой другой вершины хотя бы по одному пути. В этом случае поиск всех p -разбиений существенно сокращается: вместо полного перебора всех допустимых подмножеств вершин графа перебирается только $|P_0| = m$ разбиений. Для этого используется следующий алгоритм.

- Шаг 1. Определяется, все ли вершины графа просмотрены. Если да — то переход к шагу 4, если нет — то к шагу 2.

- Шаг 2. Выбирается любая непросмотренная вершина v_i . Составляется множество X , состоящее из вершин графа, откуда не заходят дуги в вершину v_i . Составляется множество Y , состоящее из тех вершин, в которые заходят дуги из множества X (за исключением вершин, уже входящих в X). Формируется множество Z из оставшихся вершин графа, включая v_i .

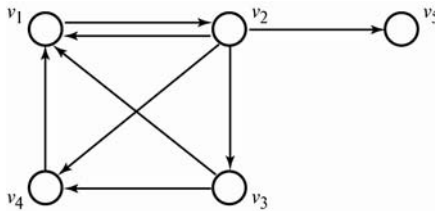


Рис. 9.6

В результате получаем p -разбиение для $p = i$, т.е. ассоциированное с вершиной v_i .

• Шаг 3. Рассчитывается показатель разбиения $k(p)$ для $p = i$, полученного на шаге 2 для вершины v_i :

$$k(i) = |Y| + \lceil |Z|/2 \rceil,$$

где скобки $\lceil \rceil$ означают целую верхнюю грань заключенного в них выражения. Переход к шагу 1.

В результате каждой вершине v_i будет соответствовать свое разбиение $p = i$. Обозначим через P_0 множество всех p -разбиений, полученных на шагах 1 – 3, их число равно n .

• Шаг 4. Рассчитывается τ_3 по формуле

$$\tau_3 = \min_{i \in P_0} k(i) - 1,$$

где минимум берется по всем p -разбиениям графа, принадлежащим множеству P_0 . Отсюда видно, что $|P_0| = n$, т.е. много меньше величины $|P|$.

Диагноз. Диагноз возможен только после выполнения анализа и определения τ . Для расшифровки синдрома используются два способа — универсальный, основанный на проверке гипотез, и упрощенный — намного менее трудоемкий, но не всегда возможный. Предварительно необходимо рассчитать τ .

1) Рассмотрим сначала упрощенный способ.

а) Если существует путь τ с одним или более нулями, закачивающийся единицей, то последний нулевой симптом указывает на исправную вершину $v_i \in G$ (рис. 9.7). Состояния остальных вершин графа находятся в процессе раскрутки.

б) Еще более эффективный путь применения упрощенного способа состоит в поиске нулевого контура с $m \geq \tau + 1$ вершинами. Нетрудно убедиться, что в этом случае все вершины, принадлежащие такому контуру, исправны и могут использоваться для раскрутки.

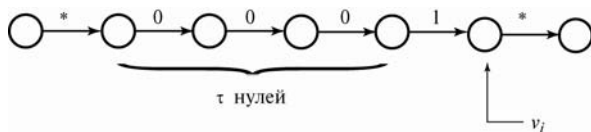


Рис. 9.7

2) При отсутствии указанных контуров и путей следует использовать универсальный способ диагноза. Порядок расшифровки синдрома состоит в следующем. Выбирается какая-нибудь непросмотренная дуга с единичным симптомом $a_{ij} = 1$. Выдвигается гипотеза $v_i \in G$, тогда $v_j \in F$. Выполняется раскрутка, в результате которой находятся другие исправные и неисправные модули. Если в процессе раскрутки возникает противоречие, то гипотеза отвергается. Выдвигается другая гипотеза $v_j \in G$ либо выбирается другая непросмотренная дуга с единичным симптомом, и процедура повторяется до полной расшифровки синдрома. Если раскрутка не может быть доведена до конца, но число найденных неисправных модулей не превышает τ , то проверяется допустимость симптомов неиспользованных дуг с учетом состояния уже проверенных вершин (т.е. удовлетворение условия $|F| \leq \tau$). Если эти симптомы допустимы, то решение найдено, если же нет, то расшифровка невозможна вследствие того, что $|F| > \tau$. Процедура завершается, когда в процессе раскруток найдены все неисправные модули и их число $|F| \leq \tau$.

Пример диагноза. Для графовой модели, изображенной на рис. 9.8, $\tau = 2$. Замечаем наличие пути с $\tau = 2$ (число нулей) и последней единицей: $4 \rightarrow 1 \rightarrow 2 \rightarrow 3$. Значит, $v_2 \in G$. Далее выполняем раскрутку: $v_2 \in G \Rightarrow v_3, v_5 \in F$, остальные вершины исправны, так как при $\tau = 2$. остальные симптомы в графовой модели возможны.

Синтез. Рассмотрим задачу реконструкции (модификации) графовой модели самодиагностируемой ВС. Под реконструкцией будем понимать добавление минимального числа дуг (без увеличения числа вершин) с целью повышения показателя диагностируемости τ (если это допускает условие (9.1)), а также удаление избыточных тестовых связей при сохранении нового τ .

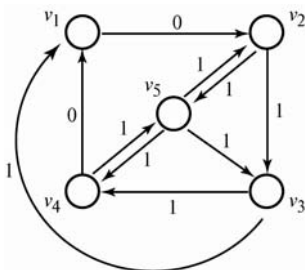


Рис. 9.8

Пусть в результате анализа получен показатель τ , а число вершин графа в соответствии с (9.1) позволяет достичь значения $\tau_0 > \tau$, и для увеличения τ требуется добавить новые тестовые связи. Для этого можно использовать следующую эвристическую процедуру. Среди полученных (в результате анализа) p -разбиений выбираем разбиение с $k(p) = \min$. В этом разбиении добавляется новая дуга из множества X и множество Z , что может увеличить $k(p)$ за счет уменьшения мощности множества X и увеличения мощности Y или Z . При этом в первую очередь добавляется дуга, обеспечивающая условие (9.2), если оно нарушено. Затем снова рассчитывается τ , и процедура при необходимости повторяется. Для устранения лишних связей выбирается p -разбиение с $k(p) = \max$, в котором удаляется дуга, ведущая из Y в Z или из X в Y , что в общем случае увеличивает мощность X (при этом нужно следить, чтобы не нарушилось (9.2)). Если при этом произошло уменьшение τ , то дуга восстанавливается, выбирается другая дуга и т.д. Отметим, что минимально возможное число тестовых связей при заданном показателе τ составляет (в соответствии с (9.2)) $N = \pi\tau$, и этого значения можно достичь при оптимальной конфигурации графа. Но оптимальная конфигурация возможна лишь при нечетном числе вершин и при наличии соответствующих физических каналов связи.

9.3. Диагностическая модель ВС с центральным ядром (на примере микропроцессорной системы)

Как было показано в предыдущем параграфе, организация диагностирования однопроцессорных (или двухпроцессорных) ВС на основе ПМЧ-модели невозможна. В этом случае обычно используется первая модель симптомов, рассмотренная ранее. Но при этом необходимо использовать внешние средства проверки, начинающие раскрутку. В дальнейшем подключаются внутренние средства системы. Внешние средства должны быть достоверно исправны. Это достигается тщательной многократной их проверкой перед началом процесса раскрутки. Кроме того, чтобы обеспечить их безотказность в течение сеанса диагностирования, они должны быть достаточно простыми. На рис. 9.9 приведена схема организации диагностирования микропроцессорной (МП) системы с шинной организацией внутренних магистралей.

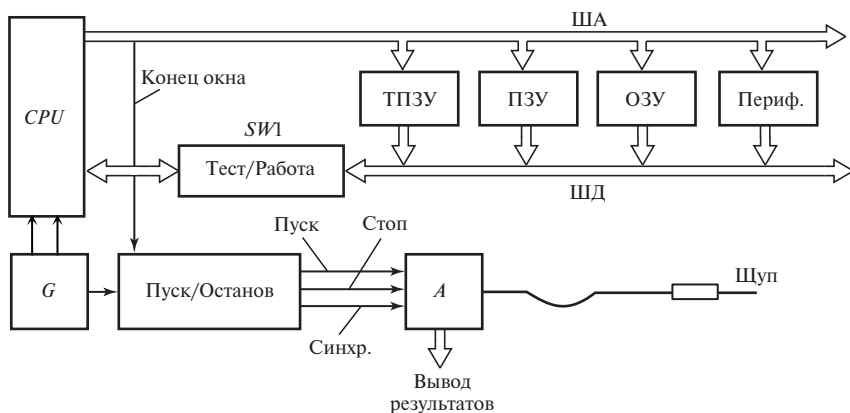


Рис. 9.9

На рисунке приняты следующие обозначения: *CPU* — центральный процессор; *SW1*, *SW2* — ключи режима работы (рабочий или тестовый) и пуска/останова процесса диагностирования ВС; ТПЗУ — тестовое ПЗУ, используемое в начальный период раскрутки; *G* — генератор тактовых импульсов центрального процессора *CPU*; *A* — анализатор реакций (как правило, сигнатурный анализатор); ШД и ША — шины данных и адреса.

Фазы раскрутки. Начальным ядром системы являются *A*, *G* и ключи *SW1* и *SW2*.

На 1 фазе раскрутки проверяется так называемое *ядро микропроцессора*, которым являются устройство управления (УУ) процессора, ША, дешифратор адреса, ШД, счетчик команд и порты ввода/вывода.

1-я фаза раскрутки — проверка ядра МП методом *свободного прогона* (свободного счета) (рис. 9.10). Для этого *SW1* переводится в тестовый режим, при котором ШД заземляется, что эквивалентно подаче на аккумулятор процессора пустой машинной команды «0000 0000» (на ассемблере — команда NOP). При подаче на каждом шаге этой команды никаких операций не совершается, но счетчик команд будет инкрементироваться, и на линиях ША возникают двоичные адресные сигналы, соответствующие числам $0, 1, 2, \dots, 2^l - 1$ (l — число разрядов ША). Ключ *SW2* запускает процесс тестирования, который завершается при появлении заднего фронта старшей линии ША. В результате образуется окно, определяющее стадию работы анализатора. Щуп анализатора поочередно подключается к разным линиям (разрядам) ША, и анализатор *A* подсчитывает число импульсов в окне (по-

ложительных и отрицательных), снимаемых с соответствующей линии. Это число заранее известно: на младшей линии 2^0 оно равно $2^l - 1$, на следующей линии 2^1 — в 2 раза меньше и т.д., на старшей линии 2^{l-1} будет два импульса (на рис. 9.10 разрядные линии ША обозначены 2^0 , 2^1 и т.д. — всего l разрядов). При подсчете только положительных импульсов их число сократится вдвое.

На этом завершается 1-я фаза, формируется 1-е расширение, и к раскрутке подключается начальное ядро и ядро МП.

2-я фаза — проверка ТПЗУ и некоторых операций АЛУ (прием и суммирование данных). Последний разряд ТПЗУ содержит сумму содержимого его ячеек по принятому модулю (эталонная сумма). Для проверки ТПЗУ ключ $SW1$ переводится в рабочий режим и процессор последовательно обращается к ячейкам ТПЗУ, выполняет суммирование их содержимого и сравнивает с эталоном из последней ячейки.

Результатом 2-й фазы является 2-е расширение.

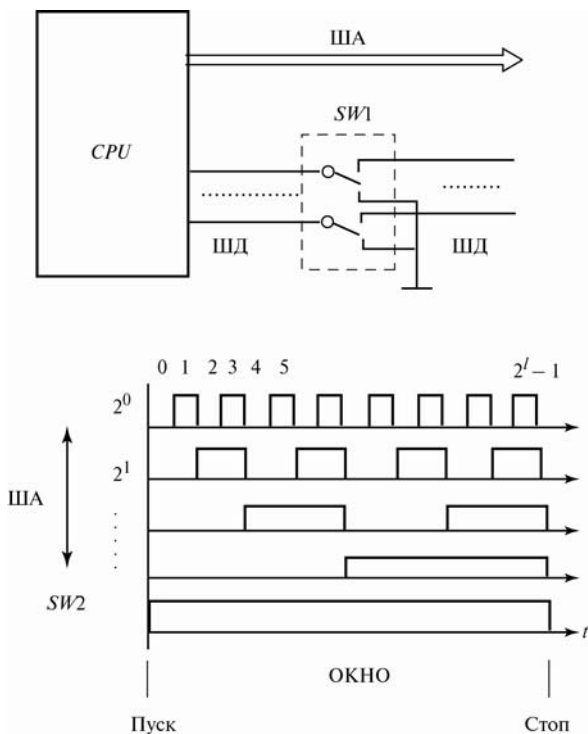


Рис. 9.10

3-я фаза — проверка работоспособности МП. ТПЗУ поочередно подает команды и тесты на МП и проверяет регистровую логику и систему команд. Результат — 3-е расширение.

4-я фаза — проверка ПЗУ. ПЗУ проверяется микропроцессором так же, как ТПЗУ. Результат — 4-е расширение.

5-я фаза — проверка ОЗУ с помощью микродиагностических программ. В проверке участвует МП+ПЗУ. Результат — 5-е расширение.

6-я фаза — МП и ОЗУ проверяют остальные блоки системы (периферию). На этом завершается процесс раскрутки.

10. ОРГАНИЗАЦИЯ ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ НА УРОВНЕ ФУНКЦИОНАЛЬНЫХ МОДУЛЕЙ

10.1. Микродиагностика

Микродиагностирование — это совокупность программно-аппаратных средств, проверяющих функциональные модули с применением *функциональных тестов*, использующих *тестовые микропрограммы*.

Тестовые микропрограммы в зависимости от их размещения делятся:

- 1) на встроенные микропрограммы (хранятся в тестовых ПЗУ),
- 2) загружаемые микропрограммы (загружаются в ОЗУ).

Микропрограммы ($\mu\text{П}$) состоят из микрокоманд ($\mu\text{к}$), которые имеют следующие разновидности:

- 1) $\mu\text{к}$ загрузки — доставка данных из накопителя (ПЗУ или ОЗУ) к входам функциональных блоков. Обозначим их ЧТ — чтение;
- 2) $\mu\text{к}$ проявления неисправностей (искажение функции блока) — $\text{М}_{\text{прояв}}$;
- 3) $\mu\text{к}$ транспортировки тестов и реакций (между блоками и к анализатору) — $\text{М}_{\text{транс}}$;
- 4) $\mu\text{к}$ анализа (сравнение с эталонами) — А .

Проверенные с помощью диагностических $\mu\text{П}$ блоки в дальнейшем участвуют в раскрутке.

С точки зрения диагностики все функциональные блоки можно разбить на два класса:

- 1) регистровая логика,
- 2) преобразовательная логика.

Функционально полной диагностической микропрограммой называется упорядоченная последовательность $\mu\text{к}$, начинающаяся с $\mu\text{к}$ загрузки, заканчивающаяся $\mu\text{к}$ анализа и содержащая хотя бы одну $\mu\text{к}$ транспортировки, а в случае проверки преобразовательной логики — хотя бы по одной $\mu\text{к}$ проявления неисправности. Микропрограмму будем обозначать буквой P . Тогда полная микропрограмма имеет вид:

$$P = \text{ЧТ} - \text{М}_{\text{транс}} - \text{М}_{\text{прояв}} - \text{М}_{\text{транс}} - \text{А}.$$

10.1.1. Микродиагностирование некоторых функциональных блоков ВС

1. Микродиагностика ПЗУ. Все ячейки ПЗУ, кроме последней, содержат некоторые команды и данные. В последней N -й ячейке записывается сумма кодов ячеек по некоторому принятому модулю. Микропрограмма имеет следующий вид.

- Процессор выполняет последовательное суммирование содержимого ячеек с 1-й до $(N - 1)$ -й и производит свертку их содержимого по принятому модулю m .
- Чтение последней — N -й ячейки ПЗУ, в которой записан правильный результат суммирования (эталон).
- Сравнение полученной суммы с эталоном.

2. Микродиагностика ОЗУ. Существует значительное количество тестов, ориентированных на различные режимы ОЗУ. Опишем некоторые из них.

а) Сканирующий тест:

- Во все ячейки записывается нулевой код (0000), выполняется последовательное считывание содержимого каждой ячейки и сравнение с эталоном (0000).
- Во все ячейки записывается единичный код (1111), выполняется считывание и сравнение с эталоном (1111).
- Снова записывается нулевой код, затем производится считывание и анализ.

Данный тест проверяет устойчивость памяти к взаимному влиянию ячеек от суммарного тока утечки.

б) Шахматный тест:

- В 1-ю ячейку записывается код 1010, во 2-ю ячейку — код 0101, в 3-ю — код 1010 и т.д. Затем — считывание и сравнение с эталоном.
- В ячейки записываются обратные коды, снова производится считывание и анализ.

Тест проверяет взаимное влияние ячеек и разрядов в статическом режиме.

в) Адресный тест:

- В каждую ячейку записывается ее адрес: 0001, 0010, 0011 и т.д., выполняется считывание и анализ.
- Тест проверяет взаимное влияние ячеек и разрядов, а также работоспособность ША и дешифратора адреса.

3. Микродиагностика счетчика:

- В счетчик записывается нулевой код 0000.
- Выполняется декрементирование на 1. В результате получаем 1111.
- Сравнение с эталоном.
- Запись 1111.
- Выполняется инкрементирование на 1. Результат 0000.
- Сравнение с эталоном.

4. Микродиагностика сумматора:

- Отключаются цепи переноса.
- В ячейки операндов заносится код 1111.
- Выполняется операция $1111 \oplus 1111 = 0000$.
- Сравнение с эталоном.
- Включаются цепи переноса.
- В регистр первого операнда заносится код 1111, в регистр второго операнда — 0001 и выполняется сложение. Результат 0000.
- Аналогично выполняется сложение без переполнения сумматора.

5. Микродиагностика регистра сдвига:

- Обнуление.
- Проверка.
- Запись кода 1111.
- Проверка.
- Запись 1010.
- Проверка.
- Сдвиг влево. Результат 0100.
- Проверка.
- Сдвиг вправо. Результат 0010.
- Проверка.

10.2. Проектирование тестовых микропрограмм

Сначала выполняется проектирование программ регистровой логики и лишь затем — преобразовательной, так как для этого необходимо исправность всех регистров. Отметим, что в микропрограммах операция пересылки (транспортировки) самая распространенная операция. Для регистровой логики используются μ к транспортировки, для преобразовательной — μ к проявления неисправности.

10.2.1. Составление микропрограмм проверки регистровой логики

При проектировании используется графовое представление программ, охватывающее возможно большую совокупность путей их выполнения. Задача выполняется в 6 этапов.

1. Построение орграфа пересылок. Вершины графа отображают регистры, дуги — логические пути (им ставятся в соответствие мк пересылки). Таких графов может быть несколько, их совокупность должна покрывать все вершины и дуги. Каждый граф содержит одну входную и одну выходную вершину (входной и выходной регистры). В графе выделяются пути (в общем случае не элементарные, т.е. проходящие через некоторые вершины по несколько раз). Каждый такой путь определяет соответствующую программу пересылки. Выделение таких путей осуществляется на последующих этапах.

В качестве примера рассмотрим процедуру построения микропрограмм для исходного графа, изображенного на рис. 10.1.

2. Нестрогое ранжирование вершин и построение псевдодерева маршрутов.

- Входной вершине графа присваивается ранг 0 (высший уровень). Инцидентным вершинам присваивается ранг 1 (более низкий уровень). Далее ищутся вершины, инцидентные вершинам ранга 1, они составляют еще более низкий уровень (ранг 2) и т.д.

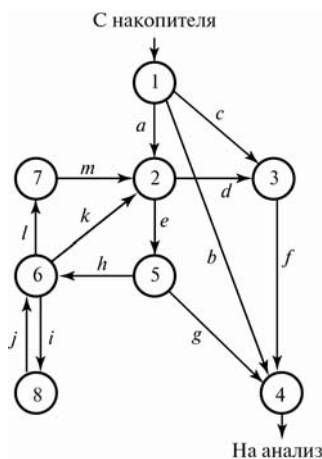


Рис. 10.1

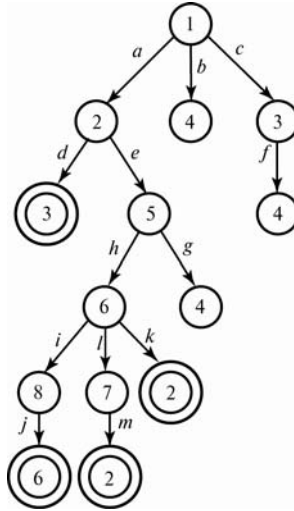


Рис. 10.2

- Вершины более высокого уровня (с меньшими номерами) соединяются имеющимися в графе дугами с вершинами меньшего уровня. В результате формируются *частичные маршруты*.
- Если в маршруте встречается вершина того же или более высокого уровня (уже встречавшаяся ранее), она объявляется *псевдовершиной* и дальнейшее построение маршрута прекращается (выходная вершина не является псевдовершиной). Результатом такого построения является *псевдодерево маршрутов* (рис. 10.2) (псевдовершины отмечены двойными кружками). Ранжирование названо нестрогим, поскольку оно отличается от строгого, при котором вершина, в которую заходят дуги от вершин i -го и $(i+1)$ -го рангов имеет ранг, больший, чем $(i+1)$. Строгое ранжирование вершин возможно только для графов без обратных связей.
- Найденные маршруты псевдодерева группируются по двум типам: маршруты типа 1 не содержат псевдовершин (1-2-5-4, 1-4, 1-3-4); маршруты типа 2 — содержат (оставшиеся 4 маршрута).

3. Построение полного дерева маршрутов. К каждой псевдовершине добавляется минимальная часть одного из маршрутов типа 1, начинающаяся с псевдовершины и заканчивающаяся конечной вершиной. Каждый маршрут нумеруется: P_1 , P_2 и т.д. В результате получаем полное дерево маршрутов (рис. 10.3).



Множество $\{P_i\}$ позволяет проверить работоспособность всех регистров (вершины дерева) и выявить ошибки микрокоманд (дуги дерева).

125

столбцы — вершины и дуги графа. В клетку (ik) таблицы покрытий заносится 1, если маршрут P_i покрывает элемент k -го столбца, в противном случае проставляется 0 (или ничего не проставляется).

Полнота проверки обеспечивается покрытием всех элементов (вершин и дуг) графа, а глубина диагностирования — различимостью всех пар столбцов. Одинаковые строки свидетельствуют об избыточности тестового набора (можно оставить только одну строку из одинаковых). Одинаковые столбцы образуют классы эквивалентных (неразличимых) состояний (их можно объединить).

Таблица 10.1

Таблица покрытий элементов графа

	1	2	3	4	5	6	7	8	a	b	c	d	e	f	g	h	i	j	k	l	m
P_1	1			1						1											
P_2	1		1	1							1			1							
P_3	1	1		1	1				1				1		1						
P_4	1	1	1	1					1			1		1							
P_5	1	1		1	1	1			1				1		1	1			1		
P_6	1	1		1	1	1	1		1				1		1	1				1	1
P_7	1	1		1	1	1		1	1				1		1	1	1	1	1		

В данной таблице одинаковых строк нет, а одинаковые столбцы есть: $\{1, 4\}$, $\{e, g\}$, $\{i, j\}$, $\{l, m\}$.

5. Минимизация таблицы покрытий. Метод минимизации зависит от цели — проверки исправности или локализации дефектов. Процедура минимизации рассматривается в гл. 12, посвященной построению безусловных и условных алгоритмов.

6. Составление микропрограмм проверки регистровой логики.

Каждой дуге ставится в соответствие микрокоманда M_k , а каждому пути графа — микропрограмма пересылки P_i , начинающаяся с мк чтения ЧТ и заканчивающаяся мк анализа А.

Приведем список микрокоманд: $a - M_1$, $b - M_2$, $c - M_3$, $d - M_4$, $e - M_5$, $f - M_6$, $g - M_7$, $h - M_8$, $i - M_9$, $j - M_{10}$, $k - M_{11}$, $l - M_{12}$, $m - M_{13}$.

Список микропрограмм:

P_1 : ЧТ — M_2 — А,

P_2 : ЧТ — M_3 — M_6 — А,

P_3 : ЧТ — M_1 — M_5 — M_7 — А,

P_4 : ЧТ — M_1 — M_4 — M_6 — А,

$$\begin{aligned}
P_5 : & \text{ЧТ} - M_1 - M_5 - M_8 - M_{11} - M_5 - M_7 - A, \\
P_6 : & \text{ЧТ} - M_5 - M_8 - M_{12} - M_{13} - M_5 - M_7 - A, \\
P_7 : & \text{ЧТ} - M_1 - M_5 - M_8 - M_{10} - M_{11} - M_5 - M_7 - A.
\end{aligned}$$

10.2.2. Составление микропрограмм проверки преобразовательной логики

Вводятся микрокоманды арифметических, логических операций, операций сдвига и др. Например:

Суммирование — M_{14} ,

Вычитание — M_{15} ,

Сдвиг — M_{16} и т.д.

Каждая $\mu\text{к}$ охватывает функциональный (преобразовательный) блок и некоторый набор регистров. В результате формируется тестовая микропрограмма проверки преобразовательной логики.

Пример проверки работы сумматора. Пусть операнд A помещается в регистр, отображаемый вершиной 2 исходного графа пересылок, операнд B — в регистр, отображаемый вершиной 3, и результат помещается в регистр, отображаемый вершиной 5. Соответственно вводится микрокоманда суммирования M_{14} . В результате формируется новая микропрограмма:

$$P_8: \text{ЧТ} - M_1 - \text{ЧТ} - M_3 - M_{14} - M_7 - A,$$

где M_1 , M_3 и M_7 — операции пересылки (дуги a , c и g соответственно).

Порядок тестирования.

Сначала проверяется вся регистровая логика путем использования микропрограмм пересылок ($P_1 - P_7$).

Затем проверяется преобразовательная логика путем использования микропрограммы P_8 и других спроектированных микропрограмм для функциональных блоков.

10.3. Декомпозиция последовательного устройства

Диагностирование объектов комбинационного типа намного проще, чем последовательного устройства (с обратными связями), так как требует меньшего объема тестов и во многих случаях разрешает использование структурных тестов. Поэтому в режиме тестирования

последовательностное устройство всегда стараются преобразовать к комбинационному виду (если это возможно).

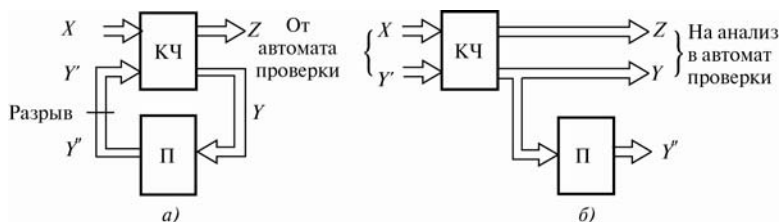


Рис. 10.4

Любое последовательностное устройство (автомат) состоит из двух частей — комбинационной части (КЧ) и элементов памяти (П), представляющих собой триггеры задержки (D-триггеры), включенные в цепи обратной связи (рис. 10.4).

Если разорвать обратную связь, то происходит *декомпозиция объекта* (рис. 10.4, б) на комбинационную часть и память. Обе эти части оказываются легкотестируемыми. Сначала тестируется КЧ — это намного проще, чем тестирование автоматов последовательностного типа. Если она исправна, тестируется память — это тоже достаточно просто. Возможность таких разрывов предусматривается на стадии проектирования устройства.

Если декомпозиция невозможна, то используются функциональные тесты.

10.4. Синтез тестов комбинационных устройств

Синтез тестов ориентирован на модель одиночных константных неисправностей комбинационных схем. Это обусловлено следующими причинами: 1) константные неисправности являются весьма распространенными; 2) тесты, ориентированные на этот вид неисправностей обычно покрывают подавляющее большинство неисправностей других видов; 3) фактически только для одиночных константных неисправностей разработаны простые методы синтеза тестов.

На практике используются несколько методов синтеза, реализованных в виде программных моделирующих систем, позволяющих найти тесты, покрывающие заданный набор возможных неисправностей. Известны следующие *методы синтеза*:

- 1) метод активизации одномерного пути,
- 2) метод булевой производной,

- 3) метод эквивалентных нормальных форм (метод Армстронга),
- 4) метод D -кубов (алгоритм Рота),
- 5) метод скобочных форм,
- 6) методы дедуктивного и конкурентного моделирования.

Наиболее простым является метод активизации одномерного пути. Но он обладает одним существенным недостатком — не всегда удается найти тесты, покрывающие некоторые неисправности, хотя такие тесты существуют. Другие методы лишены этого недостатка. Методы моделирования наиболее трудоемкие. Остальные методы синтеза тестов приблизительно равноценны по трудоемкости и предполагают возможность активизации многомерных путей. Для примера рассмотрим только метод булевой производной.

Метод булевой производной. При его применении, как и в других методах, используется понятие *активизации путей*. *Путем* в схеме называется маршрут прохождения сигнала от точки с неисправностью y_k^* до выхода F . Таких путей может быть несколько. Каждый из них называется *одномерным*, а совокупность одномерных путей, начинающихся от цепи y_k называется *многомерным путем*.

Активизацией цепи (переменной) анализируемой схемы на некотором наборе значений входных переменных x называется переключение логического состояния переменной в этой цепи при возникновении в ней константной неисправности. Если обозначить через y_k значение логической переменной в k -й цепи исправного объекта, через y_k^* — значение этой переменной при возникновении константной неисправности то активизация цепи означает $y_k = \bar{y}_k^*$. Это равенство называется *условием управляемости* цепи y_k .

Активизация пути — это переключение значений всех логических переменных вдоль одномерного или многомерного пути до выхода F схемы при возникновении неисправности y_k^* . Данное условие называется *условием наблюдаемости* неисправности y_k^* на выходе схемы.

Условием тестируемости, или *условием покрытия неисправности* y_k^* некоторым тестом является одновременное выполнение условий управляемости и наблюдаемости.

Переменная y_k есть некоторая функция n входных переменных: $y_k = f(x)$. Выходная переменная ОД представляется в виде функции $n + 1$ переменной: $F(x, y_k)$ или кратко $F(y_k)$.

Найдем условие тестируемости (покрытия) заданной неисправности.

1) Условие управляемости цепи y_k :

$$\text{если } y_k^* \equiv 0, \text{ то } f(x) = 1; \quad (10.1)$$

$$\text{если } y_k^* \equiv 1, \text{ то } f(x) = 0, \text{ или } \bar{f}(x) = 1. \quad (10.2)$$

2) Условие наблюдаемости неисправности на выходе ОД есть условие активизации выхода при активизации цепи с неисправностью:

$$\text{если } F(y_k) = 0, \text{ то } F(\bar{y}_k) = 1, \text{ или } \bar{F}(y_k) \& F(\bar{y}_k) = 1, \quad (10.3)$$

$$\text{если } F(y_k) = 1, \text{ то } F(\bar{y}_k) = 0, \text{ или } F(y_k) \& \bar{F}(\bar{y}_k) = 1. \quad (10.4)$$

Объединяя (10.3) и (10.4), условие наблюдаемости получаем в виде:

$$\bar{F}(y_k) F(\bar{y}_k) \vee F(y_k) \bar{F}(\bar{y}_k) = 1. \quad (10.5)$$

Левая часть этого равенства носит название *булевой производной* (или *булевой разности*) или *тестовой функции* и обозначается dF/dy_k :

$$dF/dy_k = F(y_k) \oplus F(\bar{y}_k).$$

В расчетах более удобно пользоваться формулой

$$dF/dy_k = \bar{F}(1) F(0) \vee F(1) \bar{F}(0),$$

в которой вместо y_k подставлена 1, а вместо \bar{y}_k — 0 (или наоборот, что следует из симметричности формулы). Из нее, в частности, видно, что булева производная не зависит от y_k .

Объединяя (10.1) и (10.5) или (10.2) и (10.5), получим условия тестируемости:

для неисправности $y_k^* \equiv 0$

$$f(x) \cdot dF/dy_k = 1, \quad (10.6)$$

для неисправности $y_k^* \equiv 1$

$$\bar{f}(x) dF/dy_k = 1. \quad (10.7)$$

Пусть задан набор возможных неисправностей. Порядок синтеза тестов выполняется в следующей последовательности.

1) Выбирается очередная цепь y_k с неисправностями $y_k^* \equiv 0$ или $y_k^* \equiv 1$.

- 2) Находится $y_k = f(x)$ и $\bar{y}_k = \bar{f}(x)$.
- 3) Находится $F(1) = F(y_k = 1)$ и $F(0) = F(y_k = 0)$, а также $\bar{F}(1)$ и $\bar{F}(0)$.
- 4) Находится $dF/dy_k = \bar{F}(1) F(0) \vee F(1) \bar{F}(0)$.
- 5) Решается уравнение (10.6) и находятся все тесты $T_1 = \{\pi_k, \pi_l, \dots\}$, покрывающие неисправность $y_k^* \equiv 0$, затем решается уравнение (10.7) и находятся тесты T_2 , покрывающие $y_k^* \equiv 1$.
- 6) Все ли цепи с неисправностями просмотрены? Если нет, то переход к шагу 1, если да — то конец.

Результаты синтеза тестов заносятся в таблицу неисправности (ТН).

Приведем некоторые свойства булевой производной, используемые для облегчения расчетов.

- 1) $dF(y_k)/dy_k$ не зависит от y_k , что уже указывалось раньше.
- 2) $dF(y_k)/dy_k = d\bar{F}(y_k)/dy_k = dF(y_k)/d\bar{y}_k = d\bar{F}(y_k)/d\bar{y}_k$ (следует из симметричности формулы булевой производной).
- 3) Если $F(y_k)$ не зависит от y_k (т.е. y_k не влияет на $F(x)$), то $dF(y_k)/dy_k \equiv 0$ при любых значениях входного тестового вектора x и при любом виде неисправности y_k^* . Обратное тоже справедливо: если выполняется данное тождество, то F не зависит от y_k .
- 4) Пусть цепь y_l находится на одном из путей между y_k и выходом F , и между y_k и F либо отсутствуют точки разветвления, либо, если они есть, то y_l не лежит внутри сходящегося разветвления (т.е. расположена до или после сходящегося разветвления). Тогда

$$dF(y_k)/dy_k = (dF(y_l)/dy_l) \cdot (dy_l(y_k)/dy_k),$$

где через $y_l(y_k)$ обозначена функция $n + 1$ переменной: $y_l = f(x, y_k)$.

- 5) Если $F(y_k) = F_1(y_k) \oplus F_2(y_k)$, то

$$dF(y_k)/dy_k = (dF_1(y_k)/dy_k) \oplus (dF_2(y_k)/dy_k).$$

Следствие: если F_2 не зависит от y_k , то $dF(y_k)/dy_k = dF_1(y_k)/dy_k$.

В расчетах удобно пользоваться булевыми производными элементарных логических функций:

- если $F = \bigwedge_{i=1}^n y_i$, то $dF/dy_k = \bigwedge_{i \neq k} y_i$ (т.е. все переменные без y_k);

- если $F = \bigvee_{i=1}^n y_i$, то $dF/dy_k = \bigwedge_{i \neq k} \bar{y}_i$ (т.е. без \bar{y}_k);
- если $F = y_1 \oplus y_2$, то $dF/dy_1 = dF/dy_2 \equiv 1$.

Пример 10.1. Найти все тесты, покрывающие неисправности $y_3^* \equiv 0$ и $y_3^* \equiv 1$ в схеме (рис. 10.5).

Решение:

- 1) Выбираем неисправности $y_3^* \equiv 0$ и $y_3^* \equiv 1$ (заданы в условии).
- 2) Находим

$$y_3 = f(x) = x_2 x_3; \quad \bar{y}_3 = \bar{f}(x) = \overline{x_2 x_3}.$$

- 3) Находим

$$F(y_3) = x_1 \vee x_2 x_3 \vee y_3 x_4,$$

затем находим

$$F(1) = x_1 \vee x_2 x_3 \vee x_4, \quad F(0) = x_1 \vee x_2 x_3,$$

$$\bar{F}(1) = \overline{x_1 \vee x_2 x_3 \vee x_4}, \quad \bar{F}(0) = \overline{x_1 \vee x_2 x_3}.$$

- 4) Определяем

$$\begin{aligned} dF(y_3)/dy_3 &= \overline{x_1 \vee x_2 x_3 \vee x_4} \cdot (x_1 \vee x_2 x_3) \vee (x_1 \vee x_2 x_3 \vee x_4) \cdot \overline{x_1 \vee x_2 x_3} = \\ &= 0 \vee 0 \vee x_4 \cdot \overline{x_1} \cdot \overline{x_2 x_3} = x_4 \cdot \bar{x}_1 \cdot \overline{x_2 x_3}. \end{aligned}$$

- 5) Для $y_3^* \equiv 0$ имеем уравнение для нахождения тестов:

$$x_2 x_3 \cdot x_4 \cdot \bar{x}_1 \cdot \overline{x_2 x_3} = 1,$$

которое не имеет решения. Поэтому

$$T_1 = \{\emptyset\}.$$

Для $y_3^* \equiv 1$: $\overline{x_2 x_3} \cdot x_4 \cdot \bar{x}_1 \cdot \overline{x_2 x_3} = 1$,

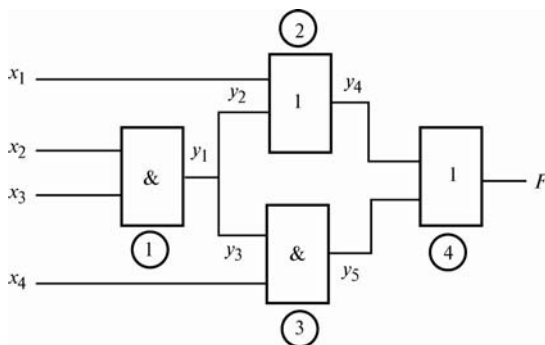


Рис. 10.5

или

$$\bar{x}_1 \bar{x}_2 x_4 \vee \bar{x}_1 \bar{x}_3 x_4 = 1,$$

откуда

$$T_2(x_1, x_2, x_3, x_4) = \{00^*1, 0^*01\},$$

или $T_2 = \{0001, 0011, 0101\}$.

Ответ: $T_1 = \{\emptyset\}$, $T_2 = \{0001, 0011, 0101\}$.

Решение задачи упрощается, если использовать свойства булевой производной и производные элементарных функций.

После нахождения функций $y_3 = f(x) = x_2 x_3$ и $\bar{y}_3 = \bar{f}(x) = \overline{x_2 x_3}$ переходим к определению производной $dF(y_3)/dy_3$. Для этого воспользуемся свойством 4 булевой производной. Поскольку от y_3 до F имеется единственный путь, то воспользуемся свойством 4 и производными для дизъюнкции и конъюнкции:

$$dF(y_3)/dy_3 = dF(y_5)/dy_5 \cdot dy_5(y_3)/dy_3.$$

Но блоки 4 и 3 (рис. 10.5) реализуют элементарные функции ИЛИ и И, для которых легко находится булева производная:

$$dF(y_5)/dy_5 = \bar{y}_4, \quad dy_5(y_3)/dy_3 = x_4.$$

Поэтому

$$dF(y_3)/dy_3 = \bar{y}_4 x_4 = \overline{x_1 \vee x_2 x_3} \cdot x_4 = \bar{x}_1 \cdot \overline{x_2 x_3} \cdot x_4.$$

Затем для $y_3^* \equiv 0$ составляется уравнение

$$f(x) dF(y_3)/dy_3 = x_2 x_3 \cdot \bar{x}_1 \bar{x}_2 \bar{x}_3 \cdot x_4 = 1,$$

которое не имеет решения ($T_1 = \{\emptyset\}$). Для $y_3^* \equiv 1$ решается уравнение

$$\bar{f}(x) dF(y_3)/dy_3 = \overline{x_2 x_3} \cdot \overline{x_1} \cdot \overline{x_2 x_3} \cdot x_4 = \overline{x_1} \cdot \overline{x_2 x_4} \vee \overline{x_1} \cdot \overline{x_3 x_4} = 1,$$

откуда $T_2 = \{00^*1, 0^*01\} = \{0001, 0011, 0101\}$, где «*» означает произвольное значение.

Ответ: $T_1 = \{\emptyset\}$, $T_2 = \{0001, 0011, 0101\}$.

Пример 10.2. Для предыдущей схемы найти тесты, покрывающие неисправности $x_4^* \equiv 0$ и $x_4^* \equiv 1$.

Решение. Поскольку $F(x) = x_1 \vee x_2 x_3$ не зависит от x_4 , то согласно свойству 2 булевой производной

$$dF(x_4)/dx_4 \equiv 0,$$

следовательно, тестов, покрывающих данную неисправность, не существует.

Ответ: $T = \{\emptyset\}$.

Если других неисправностей нет, то для найденных в предыдущих задачах тестов строится *таблица неисправностей* (ТН)

π	$y_3^* \equiv 0$	$y_3^* \equiv 1$	$x_4^* \equiv 0$	$x_4^* \equiv 1$	Исправное состояние
	a_1	a_2	a_3	a_4	
0001	0	1	0	0	0
0011	0	1	0	0	0
0101	0	1	0	0	0

Поскольку все тесты покрывают только одну неисправность $y_3^* \equiv 1$, то можно выбрать один из наборов, например 0001, и использовать его для проверки исправности схемы с локализацией неисправности $y_3^* \equiv 1$. Остальные дефекты не локализуются вследствие избыточности схемы. Отметим, что глубину диагностирования можно повысить, если добавить дополнительный выход F_1 от цепи y_1 .

10.5. Исчерпывающее и псевдослучайное тестирование

Проверка объектов на синтезированных тестах наиболее экономична по времени, так как требует минимального числа тестов. Альтернативой ей являются исчерпывающее и псевдослучайное (ПС) те-

стирования. Их существенным достоинством является отсутствие процедуры синтеза тестов на стадии проектирования.

При *исчерпывающем тестировании* последовательно перебираются все возможные векторы тестовых воздействий, число которых равно 2^n , где n — число входных переменных. Поэтому время тестирования ограничено значением n . Для сокращения этого времени используется *ПС-тестирование*.

ПС-последовательностью называется последовательность некоррелированных между собой двоичных символов в блоке длины N , который затем повторяется, образуя совокупность циклов. Чем больше N , тем более случайной является эта последовательность. Например, последовательность

001101011101 001101011101 001101011101 001101011101

Усеченное исчерпывающее тестирование

ПС-тестирование

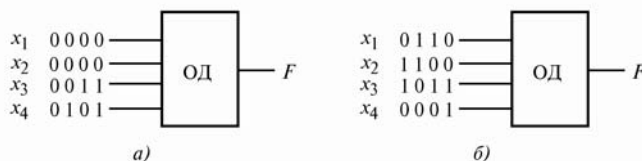


Рис. 10.6

является псевдослучайной. Цикл содержит $N = 12$ символов. Отметим, что эта последовательность заранее известна пользователю и определяется настройкой генератора. Для проверки объекта нужно, чтобы каждый его вход в процессе тестирования активизировался, так как иначе не будет покрыта возможная неисправность на этом входе. Кроме того, желательно, чтобы активизация разных входов имела одинаковую частоту. Именно таким образом строятся генераторы ПС-последовательностей. Сравним с этих позиций исчерпывающее и ПС-тестирование.

Пусть для сокращения времени при исчерпывающем тестировании решено урезать тестовую последовательность, ограничившись четырьмя тестами (рис. 10.6, а) вместо шестнадцати (на рисунке двоичные векторы расположены вертикально в порядке возрастания). Тогда вход x_1 вообще не будет активизирован и неисправность $x_1^* \equiv 0$ не будет обнаружена.

При ПС-тестировании (рис. 10.6, б) все входы активизируются. Вероятность покрытия набора заданных неисправностей устройства повышается при увеличении длины циклов ПС-последовательности,

так как при этом уменьшается корреляция символов и более полно удовлетворяются указанные требования.

Достоинства ПС-тестирования:

- 1) не требуется синтез тестов;
- 2) по сравнению с исчерпывающим тестированием объем тестов N существенно сокращается;
- 3) все входы устройства активизируются;
- 4) вероятность появления 0 и 1 на каждом входе устройства одинаковы, что приближает ПС-последовательность по своим свойствам к чисто случайной последовательности — равномерному распределению вероятностей значений входных переменных и их малой корреляции;
- 5) все символы ПС-последовательности заранее известны, так как циклы повторяются в отличие от чисто случайной последовательности.

11. СИГНАТУРНЫЙ АНАЛИЗ

11.1. Анализ реакций

Существуют два метода наблюдения реакций.

Непрерывное наблюдение реакции (рис. 11.1).

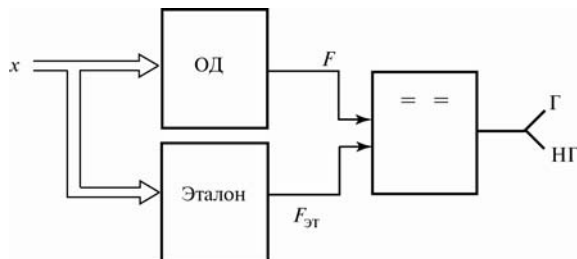


Рис. 11.1

Этот метод по затратам времени на тестирование наиболее эффективен, так как диагностирование прекращается при обнаружении расхождения между $F_{\text{Эт}}$ и F . Однако его недостатком является необходимость иметь столько эталонных блоков (или их моделей), сколько их имеется в ВС.

Сигнатурный анализ (СА) (рис. 11.2).

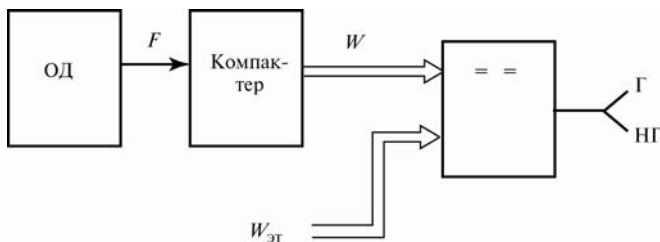


Рис. 11.2

Данный метод лишен указанного недостатка. При СА используется *компактер*, представляющей собой схему сжатия выходной последовательности (ВП) объекта. *Сигнатурой* W называется сжатая информативная характеристика ВП в виде некоторого числа. Таким образом, вместо эталонов исправных объектов в памяти автомата проверки хранятся только эталонные числа. Недостаток СА состоит в том, что тестирование нельзя прерывать: тестирование прекращается

только после подачи на вход всей тестовой последовательности, после чего формируется сигнатура.

Виды сигнатур:

$W_{\text{чет}}$ — четность числа единиц в ВП. Компактером служит триггер со счетным входом;

$W_{\text{СЕ}}$ — число единиц в ВП (СЕ — счет единиц). Компактер — счетчик числа единиц;

$W_{\text{СП}}$ — число перепадов (переходов 0 в 1 или наоборот в ВП). Компактер — счетчик числа перепадов;

$W_{\text{ост}}$ — остаток от деления ВП F на некоторый модуль M . Компактер строится на сдвиговом регистре с обратными связями, осуществляющем деление и выделение остатка.

Известны также и другие виды сигнатур.

11.2. Сигнатурный анализ методом контроля остатков

В качестве компактера для получения сигнатуры $W_{\text{ост}}$ используется сдвиговый регистр с обратными связями.

При рассмотрении контроля по числовым вычетам (гл. 7) число A представлялось в форме

$$A = qm + r,$$

где A, q, m, r — целые числа.

Аналогичное представление имеет место при СА по остаткам:

$$F = QM \oplus R,$$

где F — код ВП; M — делитель в виде некоторого задаваемого нами двоичного кода; Q — код частного и R — код остатка.

Однако операции выполняются по правилам алгебры полиномов, особенности которой состоят в следующем:

- F, M, Q и R представляются в полиномиальной форме;
- вместо арифметического сложения и вычитания используется операция сложения по модулю 2;
- арифметическое умножение и деление выполняются по правилам операций с полиномами.

Отметим, что операции можно выполнять не только в полиномиальном представлении, но и в кодовой форме.

ВП в полиномиальной форме имеет вид

$$F(x) = \sum_{i=0}^n \beta_i x^i,$$

а модуль (делитель) — в виде характеристического полинома $M(x)$, который определяет структуру компактера,

$$M(x) = \sum_{i=0}^m \alpha_i x^i.$$

Коэффициенты β_i при фиктивной переменной x определяются разрядами вектора F , n — степень полинома $F(x)$ (на единицу меньше числа разрядов), разрядные коэффициенты α_i определяют наличие или отсутствие обратных связей в компактере, а степень m полинома $M(x)$ равна числу ячеек сдвигового регистра. Разрядность остатка R на единицу меньше разрядности делителя M .

Пример операции деления полиномов. Пусть $F = 11110101$ (выходная последовательность ОД), $M = 1101$, или $F(x) = x^7 \oplus x^6 \oplus x^5 \oplus x^4 \oplus x^2 \oplus 1$, $M(x) = x^3 \oplus x^2 \oplus 1$ (примитивный полином).

$$\begin{array}{r}
 F(x) = x^7 \oplus x^6 \oplus x^5 \oplus x^4 \oplus 0 \oplus x^2 \oplus 0 \oplus 1 \quad | \quad \underline{x^3 \oplus x^2 \oplus 1} \\
 \underline{x^7 \oplus x^6 \oplus 0 \oplus x^4} \qquad \qquad \qquad x^4 \oplus 0 \oplus x^2 \oplus \dots = Q(x) \\
 0 \oplus x^5 \oplus 0 \oplus 0 \\
 \underline{0 \oplus 0 \oplus 0 \oplus 0} \\
 x^5 \oplus 0 \oplus 0 \oplus x^2 \\
 \underline{x^5 \oplus x^4 \oplus 0 \oplus x^2} \\
 x^4 \oplus 0 \oplus 0 \text{ (промежуточный остаток)} \\
 \dots\dots\dots \text{ и т.д.}
 \end{array}$$

Продолжая деление, получим окончательный остаток $R(x) = x^2 \oplus x \oplus 0$, который и является сигнатурой $W_{\text{ост}}(x)$ в полиномиальной форме.

Выполним деление в кодовой форме:

$$\begin{array}{r}
 11110101 \quad | \quad \underline{1101} \\
 \oplus \underline{1101} \qquad \qquad 10111 \\
 \hline
 0100 \\
 \oplus \underline{0000} \\
 \hline
 1001 \\
 \oplus \underline{1101} \\
 \hline
 1000 \\
 \oplus \underline{1101} \\
 \hline
 1011 \\
 \oplus \underline{1101} \\
 \hline
 110 = R
 \end{array}$$

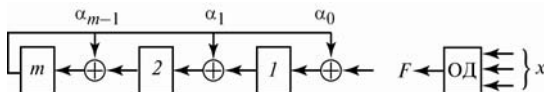


Рис. 11.3

Затемнением выделены векторы обратной связи сдвигового регистра, равные, в данном случае, либо 101, либо 000. Последний остаток (сигнатура) в кодовом представлении имеет вид $R = 110$.

На рис. 11.3 представлена схема компактера, выполняющего деление (по правилам деления полиномов). Он представляет собой сдвиговый регистр с обратными связями и содержит m разрядов, где m — степень характеристического полинома. Старший разряд анализатора помещен слева, чтобы код остатка, снимаемого с регистра, имел естественный вид (слева направо). Выходная последовательность ОД F подается в регистр с правой стороны. После подачи на вход ОД N тестов на его выходе образуется N -разрядная ВП, поступающая на вход компактера. После выполнения деления в ячейках регистра образуется m -разрядный остаток $R = W_{\text{ост}}$.

Пример компактера для СА по остаткам. На рис. 11.4 приведена схема 3-разрядного ($m = 3$) компактера, построенная на полиноме

$$M(x) = x^3 \oplus x^2 \oplus 1$$

(или в кодовой форме $M = 1101$).

Перед началом регистр обнуляется. Затем на тактах от 1 до m происходит его заполнение, при этом обратная связь еще не работает и деления не производится. На остальных тактах — от $m + 1$ до $N = n + 1$ — работает обратная связь и выполняется деление, при этом на каждом такте образуются промежуточные остатки, а на последнем такте — конечный остаток.

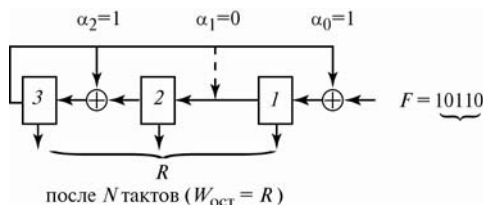


Рис. 11.4

11.3. Маскировка ошибок при сигнатурном анализе по остаткам (на сдвиговом регистре)

Неисправность ОД приводит к искажению выходной последовательности (ВП) при его тестировании:

$$F(x) = F_{\text{эт}}(x) \oplus E(x),$$

где $E(x)$ — полином вектора ошибки; $F_{\text{эт}}(x)$ — полином эталонной ВП (исправного ОД).

Ошибка E возникает в результате неисправности в схеме и может быть одиночной или кратной. Следует помнить, что кратность ошибки в ВП и кратность неисправности в схеме — это разные понятия. Например, одиночная неисправность может вызвать кратную ошибку в ВП. Отметим, что если степень полиномов $F(x)$ и $F_{\text{эт}}(x)$ равна n , то ВП F и $F_{\text{эт}}$ объекта содержит $N = n + 1$ бит. При этом степень i полинома $E(x)$ равна или меньше n (соответственно длина кода ошибки равна $i + 1$ бит).

Обозначим через e_i вектор одиночной ошибки. Он имеет следующий вид (сверху проставлены степени фиктивной переменной для каждого разряда):

$$E = e_i = \begin{matrix} x^n & x^i & x^0 & x^i & x^0 \\ 00 & \dots & 1 & \dots & 00 = 10 & \dots & 0, \end{matrix}$$

а соответствующий полином —

$$E(x) = e_i(x) = x^i.$$

В случае кратной ошибки

$$E(x) = \sum_{i=0}^n \omega_i e_i(x) = \sum_{i=0}^n \omega_i x^i,$$

где $\omega_i \in \{0,1\}$ (суммирование выполняется по mod 2). Но

$$e_i = Q_{ei} M \oplus R_{ei} \equiv R_{ei},$$

где Q_{ei} и R_{ei} — частное и остаток от деления при одиночной ошибке, а знак « \equiv » означает «сравнимость по модулю M ».

Аналогично, для кратной ошибки в ВП:

$$E \equiv \sum_{i=0}^n \omega_i R_{ei} = R_E,$$

R_E — остаток от деления кратной ошибки E на M . Тогда

$$F = (Q_{\text{ЭТ}} M \oplus R_{\text{ЭТ}}) \oplus (Q_E M \oplus R_E) = \\ = (Q_{\text{ЭТ}} \oplus Q_E) M \oplus (R_{\text{ЭТ}} \oplus R_E) \equiv R_{\text{ЭТ}} \oplus R_E = R.$$

Здесь $R = R_{\text{ЭТ}} \oplus R_E$ — реальная сигнатура проверяемого ОД.

Отсюда видно, что деление F на M можно разбить на две части: деление $F_{\text{ЭТ}}$ на M с выделением $R_{\text{ЭТ}}$ и деление E на M (на том же регистре) с выделением остатка $R_E = \text{res}(E/M)$. При $R_E = (00 \dots 0)$ ошибка E (если она есть) будет замаскирована. Незамаскированная ошибка позволяет выявить неисправность ОД, маскировка ошибки снижает достоверность тестирования.

На достоверность влияет выбираемый полином M . Особое значение при сигнатурном анализе по остаткам имеют так называемые примитивные полиномы.

Рассмотрим три случая: одиночных ошибок, двойных ошибок и ошибок произвольной кратности.

1. Случай одиночных ошибок: $E = e_i, R_E = R_{ei}$.

Для того чтобы ошибка не была замаскирована, нужно, чтобы при делении $E = e_i$ на M остаток не был нулевым:

$$R_{ei} = \text{res}(e_i/M) \neq (00 \dots 0).$$

- Если $i \leq m$ (m — число разрядов сдвигового регистра), то деление $100 \dots 0$ на M соответствует фазе заполнения регистра. При этом ошибка отсутствует: $R_{ei} = e_i \neq (00 \dots 0)$.
- Рассмотрим случай произвольного i , в частности $i > m$ (фаза собственно деления). При делении 1 на M (т.е. x^0 на $M(x)$) получим остаток $R_{e0} = \text{res}(1/M)$, при делении 10 на M (т.е. x^1 на $M(x)$) получим $R_{e1} = \text{res}(10/M)$, при делении 100 на M получим $R_{e2} = \text{res}(100/M)$ и т.д.

Следовательно, множество $\{R_{ei}\}$ может быть получено как последовательность промежуточных остатков при делении на M , при этом последовательность остатков будет периодически повторяться. Таким образом, для исключения маскировки ошибки нужно, чтобы при делении ни один промежуточный остаток R_{ei} не был нулевым.

Пусть $(m + 1)$ — разрядный код делителя имеет вид $M = 100 \dots 0$. Тогда деление бесконечной последовательности $100 \dots$ на M дает только нулевые промежуточные остатки. При любом другом делителе все промежуточные остатки в периоде отличны от нуля. Таким образом, имеем следующую теорему.

Теорема 11.1. Если $M(x)$ содержит не менее двух отличных от нуля коэффициентов, то любая одиночная ошибка в произвольной выходной последовательности F любой длительности искажает сигнатуру и проявляется.

2. Случай двойных ошибок: $E = e_i \oplus e_j$, $R_E = R_{ei} \oplus R_{ej}$.

Чтобы двойная ошибка не была замаскирована, нужно, чтобы

$$R_E = R_{ei} \oplus R_{ej} \neq (00 \dots 0) \dots \quad (11.1)$$

Как было замечено раньше, при делении $100 \dots$ на любой полином M промежуточные остатки периодически повторяются. Если суммируются остатки разных циклов, то при $R_{ei} = R_{ej}$ их сумма равна 0 и условие (11.1) не будет выполнено. Поэтому для выполнения (11.1) длина N тестовой последовательности (ВП) не должна превышать одного цикла. Рассмотрим один цикл. Для выполнения (11.1) в пределах одного цикла нужно, чтобы удовлетворялись условия:

а) остатки от одиночных ошибок были ненулевыми и не повторялись (в пределах цикла);

б) число остатков в цикле было бы не меньше числа возможных ошибок e_i . Тогда любая сумма в пределах цикла

$$R_{ei} \oplus R_{ej} \neq (00 \dots 0),$$

и каждой одиночной ошибке e_i будет соответствовать свой остаток R_{ei} . Выполнение условий а) и б) зависит от выбранного полинома $M(x)$.

Рассмотрим некоторые свойства характеристических полиномов $M(x)$, влияющие на выполнение условий а) и б).

$M(x)$ степени t называется *неприводимым* или *простым*, если он не делится без остатка ни на какой другой полином меньшей степени.

При делении $100 \dots$ на неприводимый полином $M(x)$ (в кодовой форме) число промежуточных остатков в цикле всегда больше, чем при делении на составной полином той же степени. Поэтому для выполнения условия б) $M(x)$ должен быть простым.

Из теории циклических кодов (полей Галуа) известно, что любой неприводимый полином $M(x)$ степени t является делителем некоторого другого полинома степени $k > t$, имеющего вид:

$$\Phi(x) = x^k \oplus 1,$$

или

$$\Phi = 100 \dots 01.$$

Таких полиномов бесконечно много. Например, если $x^k \oplus 1$ делится на M , то на M делятся все полиномы $(x^k \oplus 1)^l$ ($l = 2, 4, 8, \dots$), которые тоже имеют такой же вид: $(x^{2k} \oplus 1, x^{4k} \oplus 1$ и т.д.). Наименьшую степень k , при которой $\Phi(x)$ делится на $M(x)$ обозначим через T : $T = k_{\min}$; T называется *показателем, которому принадлежит $M(x)$* . Каждому значению показателя T обычно принадлежат несколько полиномов $M(x)$ определенной степени m , а различные $M(x)$ одинаковой степени m могут принадлежать различным показателям T .

Пусть имеем множество полиномов $\{M_1, \dots, M_s\}$ степени m , причем

$$M_1 \in T_1, \dots, M_s \in T_s.$$

Среди этих полиномов выберем $M(x)$, принадлежащий максимальной степени $T_{\max} = \max \{T_1, \dots, T_s\}$. Такой полином $M(x) \in T_{\max}$ называется *примитивным*. Отметим, что среди полиномов степени m обычно имеется несколько примитивных.

Существуют таблицы примитивных полиномов. Например, примитивными являются следующие полиномы (в кодовом представлении): для $m = 1$ — $M = 11$ (т.е. $M(x) = x \oplus 1$), для $m = 2$ — $M = 111$ ($M(x) = x^2 \oplus x \oplus 1$), для $m = 3$ — $M = 1101$ и 1011 , для $m = 4$ — $M = 10011$, для $m = 5$ — $M = 100101$, для $m = 6$ — $M = 1000011$ и т.д.

Отметим особенности примитивных полиномов степени m .

- Последний коэффициент в $M(x)$ равен 1, так как если бы он был нулевым, то M делился бы на x и не был бы простым.
- Промежуточные остатки при делении $\Phi(x) = x^{T_{\max}} \oplus 1$ (т.е. $100 \dots 1$) на $M(x)$ отличны от нуля и различны, а последний остаток равен $(0 \dots 0)$. Поэтому деление бесконечной последовательности $100 \dots$ на M дает только различные ненулевые и периодически повторяющиеся промежуточные остатки. Этим обеспечивается условие а).
- Показатель, которому принадлежит M , всегда равен $T_{\max} = 2^m - 1$.
- Число остатков (включая последний нулевой остаток) при делении $\Phi(x)$ на $M(x)$ максимально по сравнению со случаем использования непримитивного $M(x)$ той же степени и равно $T_{\max} - 1 = 2^m - 2$. Поэтому деление $100 \dots$ на M обеспечивает максимальный цикл повторения различных ненулевых остатков: $L = T_{\max} = 2^m - 1$.

Пусть длина тестовой последовательности F равна

$$N = n + 1 \leq T_{\max} = 2^m - 1,$$

т.е. не превышает длины цикла. Тогда число возможных одиночных ошибок также равно $N \leq T_{\max}$, при этом число различных ненулевых остатков не меньше числа одиночных ошибок. Число двойных ошибок равно C_n^2 . Но в пределах цикла они будут давать различные ненулевые остатки, так как всевозможные суммы по модулю 2 различных ненулевых кодов не могут совпадать и не равны нулю: $R_E = R_{ei} \oplus R_{ej} \neq 0$. Можно проверить на примерах, что все такие суммы будут различными.

Этим обеспечиваются условия а) и б).

Если же $N > 2^m - 1$, то деление $100\dots$ на M (т.е. для одиночных ошибок) дает больше одного цикла повторяющихся остатков. Тогда остатки из разных циклов с одинаковыми номерами будут совпадать. При этом

$$R_E = R_{ei} \oplus R_{ej} = 0.$$

В этом случае двойная ошибка маскируется.

Из сказанного следует следующая теорема.

Теорема 11.2. Если $M(x)$ — примитивный полином степени m , а F — произвольная выходная последовательность длины $N \leq 2^m - 1$, то любая двойная ошибка искажает сигнатуру и проявляется.

Пример повторения цикла длины. $T_{\max} = 2^m - 1$. Пусть $m = 2$, $M(x) = x^2 + x + 1$, или $M = 111$ (примитивный полином). Находим $T_{\max} = 2^m - 1 = 2^2 - 1 = 3$. Отсюда $\Phi = 1001$, т.е. степень полинома $\Phi(x)$ равна 3. При делении Φ на M имеем:

$$\begin{array}{r} 1001 \mid 111 \\ \oplus \underline{111} \quad 11 \\ \quad 111 \\ \oplus \underline{111} \\ \quad 00 \end{array}$$

Число остатков при этом равно $T_{\max} - 1 = 2^2 - 2 = 2$.

При делении бесконечной последовательности $100\dots$ на M число остатков в цикле составляет $T_{\max} = 2^2 - 1 = 3$:

$$\begin{array}{r}
 100... \underline{111} \\
 \underline{111} \quad 11011 \dots \\
 110 \\
 \underline{111} \\
 010 \\
 \underline{000} \\
 100
 \end{array}$$

..... (повторение цикла).

В связи с последним свойством примитивные полиномы называют *полиномами максимального цикла*.

3. Случай ошибок произвольной кратности:

$$E = e_i \oplus e_j \oplus \dots \oplus e_k, R_E = R_{ei} \oplus R_{ej} \oplus \dots \oplus R_{ek}.$$

Такие ошибки могут быть замаскированы даже при длине ВП $N \leq 2^m - 1$. Однако при использовании (в качестве делителей) примитивных полиномов вероятность маскировки таких ошибок оказывается малой. Это следует из следующей теоремы (без доказательства).

Теорема 11.3. Пусть $M(x)$ — примитивный полином степени m и F — ВП произвольной длины $N = n + 1$. Тогда вероятность маскировки ошибки любой кратности при тестировании произвольно выбранной схемы и при любом значении n и m

$$p_{\text{ош}} < 1 / 2^m \dots \quad (11.2)$$

независимо от значения n .

Например, $m = 16$. Тогда $p_{\text{ош}} < 1/2^{16} = 0.00002$ (т.е. менее 0,002 %). Вывод: $p_{\text{ош}}$ быстро убывает при увеличении числа разрядов m регистра, поэтому редко используют регистры с $m > 16$.

Для проведения сигнатурного анализа по остаткам *всегда* выбирается примитивный полином $M(x)$.

11.4. Выбор метода сигнатурного анализа

Качество СА определяется числом незамаскированных анализатором неисправностей объекта, при которых W не совпадает с $W_{\text{эт}}$. Показателем качества служит коэффициент полноты проверки исправности объекта:

$$k_{\text{п.п}} = N/N_0, \quad 0 \leq k_{\text{п.п}} \leq 1,$$

где N — число незамаскированных неисправностей, при которых $W \neq W_{\text{эт}}$, а N_0 — общее число возможных неисправностей.

Для большинства схем лучшим качеством обладает СА по остаткам. Несколько уступает ему СА по счету перепадов. Затем идут СА по счету единиц и по четности. Однако для каждой конкретной схемы для выбора метода СА нужно определить $k_{п.п.}$. С этой целью сначала строится таблица функций неисправности (ТФН), затем находятся сигнатуры для различных методов сжатия, после чего определяются $k_{п.п.}$ для различных видов сигнатур и выбирается метод, обеспечивающий максимальное значение $k_{п.п.}$.

Процесс выбора метода СА рассмотрен в «Задачнике». Но в некоторых случаях можно сразу определить наилучший метод сжатия.

Теорема 11.3. При исчерпывающем тестировании комбинационных устройств с древовидной структурой (рис. 11.5) СА по методу четности числа единиц позволяет обнаружить любую кратную или одиночную неисправность. Другими словами, при любой неисправности устройства $W_{чет\ i} \neq W_{чет\ 0}$, где $W_{чет\ 0}$ — сигнатура ВП исправного устройства, $W_{чет\ i}$ — сигнатура ВП при наличии i -й неисправности.

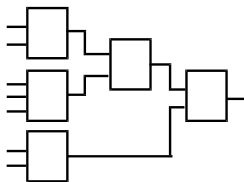


Рис. 11.5

Ясно, что для таких схем СА по четности числа единиц самый лучший, так как компактер в этом случае самый простой (триггер со счетным входом).

Теорема 11.4. При исчерпывающем тестировании комбинационных устройств, в которых пути в каждом сходящемся разветвлении имеют одинаковую четность числа инверсий, СА по методу счета единиц позволяет обнаружить любую одиночную неисправность (если она не маскируется самой схемой объекта).

На рис 11.6 схема содержит два разветвления.

В первом разветвлении верхний путь имеет одну инверсию, нижний путь — три инверсии, т.е. четность числа инверсий одинаковая.

Во втором разветвлении верхний путь имеет три инверсии, нижний путь — одну инверсию, т.е. и здесь четность числа инверсий одинаковая. Следовательно, любая одиночная неисправность схемы не будет замаскирована сигнатурным анализатором.

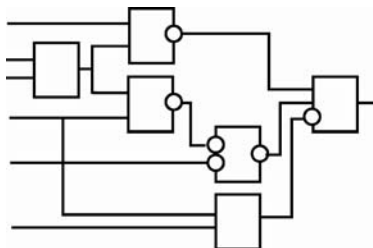


Рис. 11.6

Теорема 11.5. (обобщение результатов § 11.3).

СА на сдвиговом регистре, построенном на примитивном полиноме степени m , маскирует не больше ошибок, чем любой другой СА на сдвиговом регистре той же или меньшей степени. При этом $p_{\text{ош}} < 1/2^m$. Поэтому в СА по остаткам всегда используют m -разрядные регистры, построенные на примитивных полиномах.

Примечание. Теоремы 11.3 и 11.4 справедливы для исчерпывающего тестирования, а теорема 11.5 — для любого вида тестирования (исчерпывающего, псевдослучайного или на синтезированных тестах). Качество СА зависит: от схемы ОД, от метода тестирования, от порядка подачи тестов, от вида СА.

11.5. Эквивалентные схемы сигнатурного анализатора на сдвиговом регистре

Рассмотренная ранее схема СА по остаткам на основе полинома $M(x)$ называется схемой с внутренними сумматорами. Для нее

$$W = \text{res} (F/M).$$

Для любой схемы с внутренними сумматорами существует эквивалентная ей схема с внешними сумматорами, обладающая таким же качеством. Эта схема строится на основе полинома $M^*(x)$.

Пусть

$$M(x) = \sum_{i=0}^m \alpha_i x^i$$

и

$$M^*(x) = \sum_{i=0}^m \alpha_i^* x^i.$$

Полином $M^*(x)$ связан с $M(x)$ соотношением

$$M^*(x) = x^m \cdot M(1/x).$$

Пример 11.1.

Пусть

$$M(x) = x^3 \oplus x^2 \oplus 1.$$

Здесь $m = 3$, $\alpha_0 = \alpha_2 = \alpha_3 = 1$, $\alpha_1 = 0$. Тогда

$$M^*(x) = x^3 (1/x^3 \oplus 1/x^2 \oplus 1) = x^3 \oplus x \oplus 1.$$

Здесь $\alpha_0^* = \alpha_1^* = \alpha_3^* = 1$, $\alpha_2^* = 0$.

Схема регистра с внешними сумматорами имеет другой вид и приведена на рис. 11.7. Такая схема называется *СА на сдвиговом регистре*. Сигнатуру, снимаемую с нее, будем обозначать W^* . Для нее $W^* \neq \text{res}(F/M)$ и $W^* \neq \text{res}(F/M^*)$, но значения W^* и W связаны линейной зависимостью. Поэтому, зная W^* , можно однозначно определить W , и наоборот. Но на практике просто находят сигнатуру для исправного объекта путем прогона известных тестов (например, при исчерпывающем тестировании), затем записывают ее в память эталонных сигнатур.

Отметим, что число состояний в обеих схемах одинаково и равно $2^m - 1$ (все состояния, кроме 00...0). В этом смысле схемы эквивалентны.

Достоинство регистра с внешними сумматорами — регистр и блок сумматоров конструктивно независимы (две разные микросхемы); кроме того, такой анализатор более простой и быстродействующий. Поэтому на практике в большинстве случаев применяют именно его.

Значение сигнатуры W^* можно получить, проследив по тактам содержимое разрядов регистра. Введем обозначения:

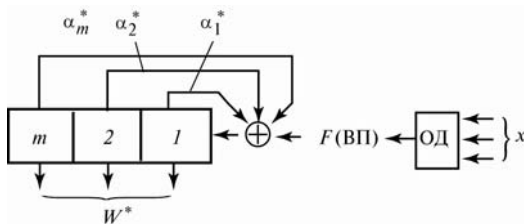


Рис. 11.7

$q_i^*(k)$ содержимое i -й ячейки регистра на k -м такте,

$F(k)$ — символ, поступающий на регистр с выхода ОД на k -м такте, т.е.

$$F = F(1) F(2) F(3) \dots$$

Тогда

$$q_1^*(k+1) = \alpha_1^* q_1^*(k) \oplus \alpha_2^* q_2^*(k) \oplus \dots \oplus \alpha_m^* q_m^*(k) \oplus F(k+1),$$

$$q_2^*(k+1) = q_1^*(k),$$

.....

$$q_m^*(k+1) = q_{m-1}^*(k).$$

В матричной форме эти соотношения имеют вид:

$$W^*(k+1) = B \cdot W^*(k) \oplus F(k+1),$$

где $W^*(k)$ — содержимое регистра на k -м такте, а B — матрица, определяемая приведенными соотношениями.

$$B = \begin{bmatrix} \alpha_1^* & \alpha_2^* & \dots & \alpha_{m-1}^* & \alpha_m^* \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \quad W^*(k) = \begin{bmatrix} q_1^*(k) \\ \dots \\ q_m^*(k) \end{bmatrix},$$

$$F(k+1) = \begin{bmatrix} F(k+1) \\ 0 \\ \dots \\ 0 \end{bmatrix},$$

векторы $W^*(k)$ и $F(k+1)$ имеют размерность $(m \times 1)$, а матрица B — размерность $(m \times m)$, при перемножении матриц сложение выполняется по модулю 2.

Пример 11.2. Определить сигнатуру для компактера, построенного на полиноме (в кодовом виде) $M^* = 1011$, если на его вход подается последовательность $F = 11110101$.

Анализатор для указанного полинома изображен на рис. 11.8.

11.6. Генерация ПС-последовательностей

Аппаратная реализация. Для генерации используется такой же регистр с ОС на примитивном полиноме, как и в СА. Если на его вход подать 100..., то на выходе каждого разряда регистра возникает ПС-последовательность с периодом $2^m - 1$, при этом вероятности появления 0 и 1 (в длинной последовательности) одинаковые. Регистр перебирает все свои состояния, кроме состояния 00...0. Поэтому можно занести в регистр любой ненулевой начальный код и осуществлять сдвиги, ничего не подавая на вход. ПС-последовательности снимаются с любых разрядов регистра и подаются на входы тестируемого объекта. Такие ПС-последовательности максимальной длины называются *М-последовательностями* (в них корреляция между соседними символами минимальна), а примитивные полиномы с ОС иногда называют *полиномами максимального цикла*.

Программная реализация. Прежде всего отметим, что все генераторы случайных чисел, реализованные в цифровых машинах, на самом деле порождают не случайные, а псевдослучайные последовательности. Наилучшим качеством обладают генераторы, имитирующие работу сдвигового регистра с ОС, чаще всего с внешними сумматорами. Работа такого генератора основана на формуле

$$W^*(k+1) = B \cdot W^*(k),$$

которая отличается от приведенной выше только отсутствием входного вектора $F(k+1)$, поскольку на вход генератора ничего не подается.

12. АЛГОРИТМЫ ДИАГНОСТИРОВАНИЯ

12.1. Классификация

Алгоритм — набор тестов и порядок их выполнения, обеспечивающие заданную полноту проверки и глубину диагностирования. Целью построения алгоритма является минимизация средней длины проверки или, в более общем случае, минимизация средней стоимости проверки. Под *длиной* l_i i -й проверки понимается число элементарных проверок, необходимое для определения i -го состояния объекта, а под *средней длиной* $l_{\text{ср}}$ — математическое ожидание числа элементарных проверок с учетом вероятностей различных состояний объекта $p(a_i)$.

Приведем классификацию по различным признакам.

Классификация по структуре алгоритма

1) *Безусловные*, в которых последовательность тестов предопределена:

а) с безусловной остановкой, в которых длина всех проверок одинакова, т.е. $l_{\text{ср}} = l = \text{const}$,

б) с условной остановкой, в которых длина каждой i -й проверки l_i зависит от состояния a_i , т.е. $l_i = \text{var}$ (i — номер состояния объекта, включая исправное a_0).

Пример алгоритма с безусловной остановкой. Результатом элементарной проверки π_i является заключение «Годен» (Г) или «Негоден» (НГ).

π_1	π_2	π_3	π_4	a_i	l_i
Г	Г	Г	Г	a_0	$l_0 = 4$
Г	НГ	Г	НГ	a_1	$l_1 = 4$
НГ	Г	Г	НГ	a_2	$l_2 = 4$

Пример алгоритма с условной остановкой. Такой алгоритм получается из предыдущего путем исключения лишних проверок.

π_1	π_2	π_3	π_4	a_i	l_i
Г	Г	Г	Г	a_0	$l_0 = 4$
Г	НГ			a_1	$l_1 = 2$
НГ				a_2	$l_2 = 1$

Здесь длина проверки, т.е. среднее число элементарных проверок с результатом «Годен» (Г) или «Негоден» (НГ), будет меньше:

$$l_{\text{ср}} = \sum p(a_i)l_i.$$

Результатом построения безусловных алгоритмов является *диагностический словарь* (ДС) — минимизированная таблица соответствия результатов проверок состоянию объекта. ДС для каждого ОД постоянно хранится в памяти автомата проверки. Примерами ДС являются приведенные выше таблицы.

2) *Условные*, в которых последовательность элементарных тестов зависит от промежуточных результатов. Для них также $l_i = \text{var}$ и

$$l_{\text{ср}} = \sum p(a_i)l_i.$$

Результатом построения условных алгоритмов является *дерево проверок*, являющееся графической моделью условного алгоритма. Построенное дерево также хранится в памяти автомата проверки.

Классификация по цели алгоритма:

- 1) для проверки исправности (работоспособности) ОД;
- 2) для локализации дефектов.

Классификация по модели неисправностей:

- 1) с ориентацией на одиночные неисправности;
- 2) с ориентацией на кратные неисправности.

В дальнейшем рассматриваются только алгоритмы, ориентированные на модель одиночных неисправностей.

Задача построения алгоритма — минимизация функции цели при выполнении ограничений.

Функция цели:

$$l_{\text{ср}} = \sum p(a_i)l_i \rightarrow \min.$$

Ограничения: для алгоритма проверки работоспособности

$$k_{\text{п.п}} \geq k_{\text{п.п. доп}},$$

для алгоритма локализации дефекта

$$k_{\text{г.д}} \geq k_{\text{г.д. доп}},$$

где $k_{\text{п.п}}$ — коэффициент полноты проверки — доля неисправностей, охватываемых проверкой, от общего числа неисправностей;

$k_{\text{г.д}}$ — коэффициент глубины диагностирования — доля неисправностей, различаемых проверкой, от общего числа неисправностей;

$k_{\text{п.п. доп}}$ и $k_{\text{г.д. доп}}$ — допустимые значения этих коэффициентов.

Примечание: Для безусловных алгоритмов с безусловной остановкой $l_{\text{оп}} = l = \text{const}$, и задача состоит в $l \rightarrow \min$ при указанных ограничениях.

Безусловные алгоритмы, в том числе и построение диагностического словаря, подробно описаны в «Задачнике».

12.2. Безусловные алгоритмы для модели одиночных неисправностей

Безусловные алгоритмы с безусловной остановкой.

Исходные данные: для алгоритмов проверки исправности — ТН (без столбца a_0), для алгоритмов локализации дефектов — таблица пар состояний (ТПС), получаемая из ТН (без удаления столбца a_0). Каждый столбец ТПС соответствует паре состояний (a_i, a_j) , а строки — номерам тестов. В клетке таблицы проставляется единица, если тест различает состояния (для одного состояния выдает результат «Годен», для другого — «Негоден»), и ноль, если не различает.

Нижняя граница функции цели для безусловных алгоритмов локализации дефектов с безусловной остановкой оценивается выражением

$$l_{\min} \geq \lceil \log_2 (n + 1) \rceil,$$

где n — число неисправностей, а $n + 1$ — общее число состояний, включая исправное.

Правая часть неравенства, являющаяся нижней границей для l_{\min} , не всегда достижима, так как зависит от заданного набора тестов. Однако она указывает на степень близости l_{\min} к наилучшему случаю. Алгоритм с безусловной остановкой изложен в «Задачнике» и здесь рассматриваться не будет, поскольку применяется довольно редко.

Безусловные алгоритмы с условной остановкой. Примером такого алгоритма может служить распространенная *процедура обратного просмотра* (рис. 12.1).

Функциональные блоки и все цепи ранжируются в обратном порядке и нумеруются справа налево в соответствии с рангами. Для каждой цепи определяется эталонная сигнатура при использовании 16-разрядного сдвигового регистра и исчерпывающего тестирования. На рисунке приведены номера тестов π_i (в соответствии с ранжировкой), в рамках указаны 4-разрядные 16-ричные эталонные сигнатуры.

Алгоритм состоит в проверке соответствия реальных и эталонных сигнала-

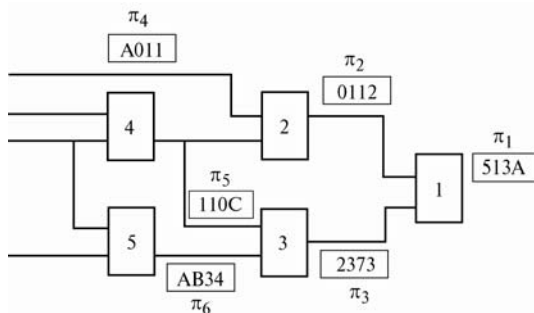


Рис. 12.1

тур, начиная с выхода схемы. Пусть получены следующие результаты проверок:

π_1	π_2	π_3	π_4	π_5
$531A_{16} \rightarrow 1234_{16} \rightarrow 2373_{16} \rightarrow A011_{16} \rightarrow 110C_{16} \rightarrow \text{Блок 2 неисправен.}$				
НГ	НГ	Г	Г	Г

Первая и вторая проверки дают результат НГ, так как сигнатуры не совпадают с эталонной; третья, четвертая и пятая проверки — Г. В результате локализуется неисправный блок 2. Дальше проверки можно не проводить. В результате получаем $l = 5$ вместо 9 (включая проверки сигнатур на входных полюсах).

12.3. Условные алгоритмы для модели одиночных неисправностей

Эти алгоритмы наиболее распространен в практике технической диагностики. Исходными данными в этом случае служит ТН (со столбцом a_0). Условный алгоритм позволяет как проверить исправность (работоспособность) ОД, так и локализовать дефекты и практически всегда более эффективен, чем безусловный алгоритм. Моделью условного алгоритма является *дерево проверок*. Вершины дерева характеризуют состояние процесса диагностирования и отображают степень нашего знания о состоянии ОД. Дуги определяют результаты выбранных проверок («Годен» или «Негоден»).

Процедура построения дерева состоит в выборе на каждом шаге нужной проверки π_k из ТН так, чтобы общая совокупность проверок $\{\pi_k\}$ имела минимальную среднюю длину теста $l_{cp} = \sum p(a_i)l_i$, где

l_i — число элементарных проверок $\{\pi_k\}$ в дереве от корневой вершины до вершины a_i .

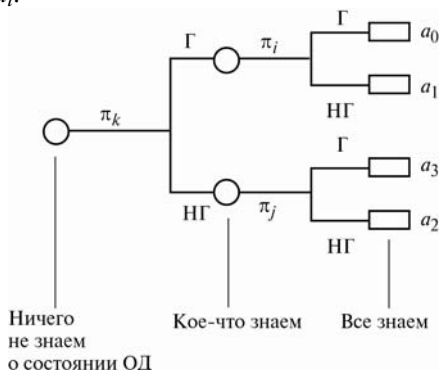


Рис. 12.2

На рис. 12.2 приведен пример дерева. До начала диагностирования (корень дерева), мы ничего не знаем о состоянии объекта диагностирования; по окончании диагностирования (висячие вершины) мы точно знаем состояние ОД (a_0 , a_1 , a_2 и т.д.); на промежуточных стадиях (промежуточные вершины) имеем некоторую информацию о состоянии, а именно: какие состояния невозможны и какие возможны. После каждой вершины проставляется номер выбранной на очередном шаге проверки π_i и два результата (дуги дерева) — результат «Годен» (Г) — вверх и результат «Негоден» (НГ) — вниз. Висячие вершины обозначены прямоугольниками.

С каждой вершиной ассоциируется ТН или ее сокращенный вариант. С корнем ассоциируется полная ТН. С вершиной по результату Г — таблица с удаленными столбцами (состояниями a_i), которые невозможны, т.е. столбцами, покрываемыми выбранной проверкой π_k . С вершиной по результату НГ ассоциируется таблица, в которой удалены оставшиеся столбцы, т.е. столбцы, не покрываемые выбранной проверкой π_k (соответствующие состояния невозможны); при этом невозможным оказывается и исправное состояние a_0 объекта, поэтому столбец a_0 удаляется. Таким образом, по мере продвижения от корня дерева к висячим вершинам происходит дихотомия ТН и ее сокращение. В результате с каждой висячей вершиной будет ассоциирована таблица, состоящая из одного столбца, определяющего состояние объекта диагностирования (a_0 , a_1 , a_2 , ...). При этом самая верхняя ви-

сячая вершина дерева соответствует состоянию a_0 , так как до нее были только результаты «Годен».

Перейдем к задаче построения дерева. Для минимизации $l_{\text{ср}}$ можно использовать разные подходы, например, применить метод динамического программирования или метод ветвей и границ. Но наиболее простой подход состоит в процедуре, подобной эффективному кодированию. Если на каждом шаге построения условного алгоритма выбирать наиболее информативную проверку, то $l_{\text{ср}} \rightarrow \min$ (как при эффективном кодировании). Такой подход назовем *информационным методом построения условного алгоритма*.

Рассмотрим строку ТН (рис. 12.3), в нижней строке записаны вероятности соответствующих состояний $p_i \equiv p_i(a_i)$ объекта. Значения p_i можно определить по надежностным характеристикам ОД: p_0 есть вероятность безотказной работы объекта в момент проведения сеанса диагностирования, а p_i ($i \neq 0$) — вероятности одиночных отказов.

Высоконадежные объекты, для которых

$$p_0 \gg \sum_{i=1}^n p_i$$

значение p_i практически совпадает с ненадежностью i -го элемента. Если эти вероятности неизвестны, то нужно ориентироваться на наихудший случай равновероятности отказов всех элементов: $p_i = p$ ($i \neq 0$). Тогда, если n — число элементов объекта, то

$$p_0 = 1 - \sum_{i=1}^n p_i = 1 - np.$$

При построении условного алгоритма для этого случая средняя длина проверки $l_{\text{ср}}$ окажется несколько больше, чем при ориентации на реальные вероятности состояний, но полнота проверок и глубина диагностирования остается прежней. Алгоритм для рассматриваемого случая строится таким образом, чтобы минимизировать $l_{\text{ср}}$.

a_i	a_1	a_2	a_3	a_4	a_0
π_k	1	1	0	0	0
p_i	p	p	p	p	p_0

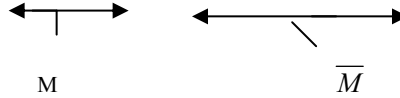


Рис. 12.3

Обозначим через N множество возможных неисправностей ($a_1— a_n$), $|N| = n$; через M — множество столбцов (состояний объекта) с единицами в рассматриваемой строке ТН, т.е. множество неисправностей, охватываемых данной проверкой, $|M| = m$; через \overline{M} — множество столбцов с нулями в строке ТН, $|\overline{M}| = n + 1 - m$.

Пусть A — случайное событие, состоящее в появлении результата проверки. Тогда неопределенность нашего знания о результате проверки π_k (Γ или $\overline{\Gamma}$) до ее выполнения равна априорной энтропии $H(A)$, а неопределенность после выполнения проверки (апостериорная энтропия) $H(A/\pi_k) = 0$, так как результат будет известен.

Количество получаемой нами информации, бит, от данной проверки равно

$$\begin{aligned} I &= H(A) - H(A/\pi_k) = H(A) = - [p(\Gamma) \log_2 p(\Gamma) + p(\overline{\Gamma}) \log_2 p(\overline{\Gamma})] = \\ &= - (P[a_i \in \overline{M}] \log_2 P[a_i \in \overline{M}] - P[a_i \in M] \log_2 P[a_i \in M]) \end{aligned}$$

и $\max I = \max H(A) = 1$ бит, что имеет место при $P[a_i \in M] = m p = 1/2$. Но $p = (1 - p_0)/n$. Отсюда находим

$$m = 1/2p = n / [2(1 - p_0)]. \quad (12.1)$$

График этой функции приведен на рис. 12.4. Но реально функция (12.1) определена на интервале $0 \leq p_0 \leq 1/2$, так как при $p_0 > 1/2$ число единиц в строке $m > n$, что не может быть. По этому графику или по формуле (12.1) можно определить наиболее информативную проверку π_k , т.е. строку ТН, имеющую m единиц в соответствии с графиком. При $p_0 > 1/2$ наиболее информативной является проверка, имеющая $m = n$ единиц.

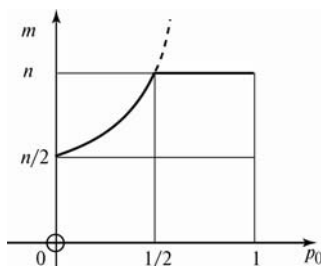


Рис. 12.4

В зависимости от значения надежности p_0 объекта можно выделить три зоны:

- 1) $p_0 \geq 1/2$. В этой зоне наиболее информативная проверка (строка ТН) содержит *максимальное число* единиц (в оптимальном случае $m = n$). Таких строк может оказаться несколько, тогда в качестве очередной проверки выбирается любая из них;
- 2) $p_0 = 0$. В этом случае выбирается строка, содержащая $m = n/2$ единиц или близкое к этому числу значение;
- 3) $0 \leq p_0 \leq 1/2$. В этом диапазоне выбирается строка, определяемая формулой (12.1) или графиком, которая соответствует наиболее информативной проверке.

Если объект высоконадежный (в этом случае $q_0 = 1 - p_0 \ll 1$), то третью зону можно не рассматривать. Однако вторую зону нельзя не рассматривать. Этот случай определяет ситуацию, когда пользователь априорно знает, что объект неисправен (при этом столбец a_0 в ТН отсутствует). На этом принципе базируется процедура построения дерева.

Алгоритм построения дерева проверок:

1. Выбирается очередная непросмотренная вершина дерева (вначале — корневая) и ассоциированная с ней ТН. Если таблица состоит только из одного столбца, то соответствующая вершина — терминальная, ей приписывается состояние a_i данного столбца и ветвление от этой вершины прекращается. В противном случае (в таблице — более одного столбца) выполняется переход к шагу 2.

2. Проверяется наличие столбца a_0 . Если он есть (это означает, что не исключена возможность работоспособного состояния ОД), то выбирается строка ТН (проверка π_k) с максимальным числом единиц $m = \max$. Если столбец a_0 отсутствует, то выбирается строка с числом единиц $m = n/2$ (или наиболее близким к этому значению), где n —

число столбцов в данной ТН (без столбца a_0). Такие проверки являются наиболее информативными.

3. Выполняется ветвление: рассматриваются два возможных результата выбранной проверки «Г» и «НГ» и строятся две новые вершины и соответствующие редуцированные ТН в соответствии со следующими правилами.

а) Для построения сокращенной таблицы по результату «Г» удаляются столбцы с единицами в выбранной строке и сама строка (так как по данному пути она не повторится). Получается сокращенная таблица с новыми значениями n и m .

б) Для построения таблицы по результату «НГ», удаляются столбцы с нулями, а также выбранная строка. В полученной ТН столбец a_0 (если он был) оказывается удаленным.

Переход к шагу 1.

Процедура построения завершается, когда построены все терминальные (висячие) вершины.

Для определения l_{cp} нужно знать значение p_0 , которое оценивается либо экспериментально, либо определяется расчетным путем по λ -характеристикам элементов. Для этого по дереву для каждого состояния a_i определяется совокупность элементарных проверок l_i , после чего рассчитывается

$$l_{cp} = \sum p(a_i)l_i,$$

которое будет минимальным.

Пример построения дерева проверок.

Пусть по результатам синтеза тестов построена таблица неисправности (табл. 12.1). В ней возможное число одиночных неисправностей $n = 4$. Замечаем отсутствие в ней столбца a_0 . Это означает, что объект изначально неисправен ($p_0 = 0$ и $p = 1/4$). Построим дерево проверок. Исходная таблица ассоциируется с корневой вершиной дерева (рис. 12.5). Выбираем строку с числом единиц $m = n/2 = 2$. Таких строк две, выбираем любую, например π_1 , покрывающую состояния a_2 и a_3 .

Таблица 12.1

π_k	a_1	a_2	a_3	a_4
π_1		1	1	
π_2	1	1	1	1
π_3	1		1	1

π_4		1		1
π_5		1		

Отметим, что если бы нулевой столбец существовал, то следовало бы выбрать строку с $m = \max$, т.е. π_2 . Далее выполняем разбиение ТН на две подтаблицы — по результату «Годен» и по результату «Негоден» (Г и НГ). Для построения первой подтаблицы из ТН удаляются столбцы с единицами, а для второй подтаблицы — столбцы с нулями в выбранной строке π_1 , которая тоже удаляется. В полученных таблицах число возможных неисправных элементов тоже изменяется, в

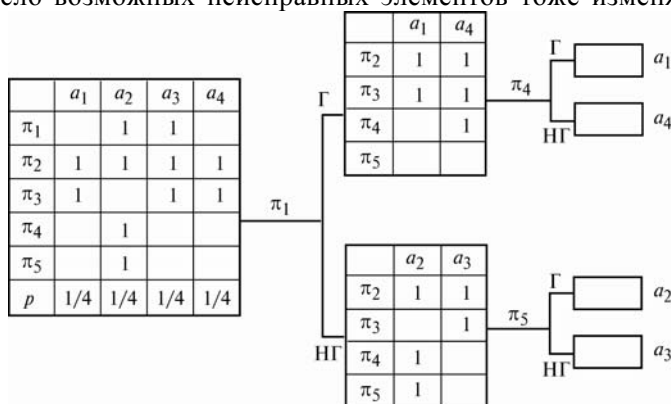


Рис. 12.5

данном случае оно равно $n = 2$. Подобная дихотомия производится при построении подтаблиц для последующих вершин. В результате получаем полное дерево проверок.

После построения дерева можно подсчитать среднюю длину проверки по формуле

$$l_{\text{cp}} = \sum_{i=1}^4 p(a_i) l_i$$

где l_i — число проверок до достижения i -й висячей вершины.

В данном примере все l_i одинаковы и равны 2, а $p(a_i) = 1/4$. Поэтому $l_{\text{cp}} = 2$.

13. ПОВЫШЕНИЕ ТЕСТОПРИГОДНОСТИ СВТ

13.1. Общие принципы повышения контролепригодности и тестопригодности СВТ

Для повышения ремонтпригодности в первую очередь нужно позаботиться о повышении контролепригодности и тестопригодности на стадии проектирования объекта, поскольку именно эти характеристики влияют на трудоемкость обнаружения и локализации дефектов и время проведения ремонта. Для этого существуют различные пути.

1) Декомпозиция последовательностных устройств путем разрыва глубоких обратных связей. Для этого имеются следующие возможности:

- механический разрыв;
- снятие питания;
- перевод части элементов в третье состояние;
- электрический разрыв путем использования элементов управляемого разрыва (ЭУР).

На рис. 13.1 приведена схема использования ЭУР, разрывающего две части рабочей схемы. Сигнал v — управляющий. При $v = 0$ цепи y и z соединяются (рабочий режим). При $v = 1$ (тестовый режим) вместо y на вторую часть схемы подается внешний тестовый сигнал x .

Схема ЭУР очень проста и приведена на рис. 13.2.

2) Разрыв цепей около точек разветвления.

Наихудшей называется неисправность с минимальной вероятностью обнаружения при тестировании (обнаруживается на редких тестах или не обнаруживается вообще). Такая неисправность располагается либо на входах схемы, либо в цепях вблизи точек разветвления. Повысить тестопригодность можно ликвидацией точек разветвления в тестовом режиме путем разрыва с добавлением дополнительного входного полюса (рис. 13.3). В тестовом режиме на дополнительный вход и на остальные входы схемы подаются тестовые сигналы, проявляющие данную неисправность.

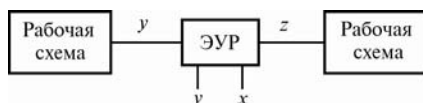


Рис. 13.1

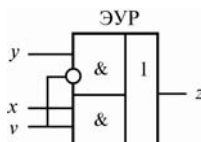


Рис. 13.2



Рис. 13.3

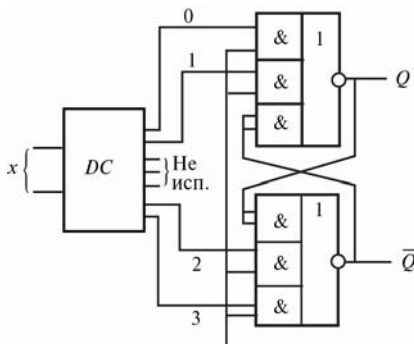


Рис. 13.4

- 3) Устранение состязаний путем использования синхронных элементов.
- 4) Запрет недопустимых тестовых векторов.

Примером может служить RS -триггер на элементах ИЛИ-НЕ, который не допускает одновременной подачи на входы R и S единичных сигналов. Запрет обеспечивается путем введения промежуточного декодера (рис. 13.4), с выхода которого единичные сигналы поступают либо на вход R , либо на вход S .

- 5) Обеспечение возможности обнуления всех элементов памяти, регистров и счетчиков и установки последовательностных устройств в начальное состояние. Этим достигается повторяемость реакций при псевдослучайном или других видах тестирования.
- 6) Добавление дополнительных входных и выходных полюсов в схемах или избыточных элементов с целью повышения тестопригодности.
- 7) Разнесение аналоговых и цифровых элементов по разным конструктивам.

13.2. Метод сканирования

Идея состоит в декомпозиция схемы на комбинационную часть (КЧ) и регистровую логику, в результате чего повышается ее контролепригодность и тестопригодность. Используется при проектировании заказных кристаллов, но может применяться и в любых других конструктивах. В схемах, использующих метод сканирования, каждый D -триггер регистровой логики работает вместе с ЭУР, имеющим два входа — рабочий (D_1) и тестовый (D_2), а также вход управления G (в прежних обозначениях — это входы y , x и v) (рис. 13.5, а). На рис. 13.5, б приведено условное изображение такого усложненного триггера. При $G = 0$ включается рабочий режим (при этом используется только вход D_1), а при $G = 1$ — тестовый режим (используется только вход D_2). Каждый триггер T — двухступенчатый, состоит из двух D -триггеров — ведущего и ведомого. При появлении импульса синхронизации в ведущий триггер записывается состояние переменной на его входе, а после окончания импульса это значение переписывается в ведомый триггер и появляется на выходе T . Такая организация нужна для устранения состязаний при одновременной записи различающихся данных в разные триггеры T .

В рабочем режиме ($G = 0$) триггеры выполняют рабочие функции, входы D_2 при этом не используются (рис. 13.6), x и z — внешние выходы объекта.

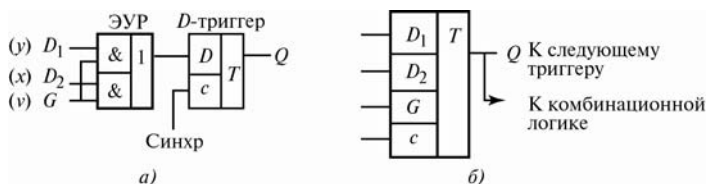


Рис. 13.5

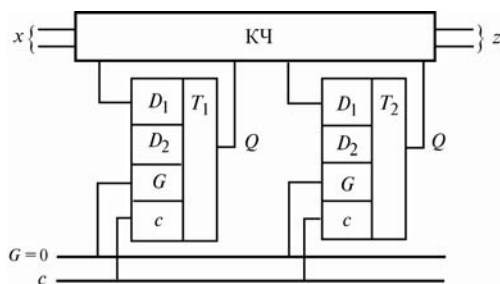


Рис. 13.6



Процедура тестирования.

- 1) Устанавливается тестовый режим: $G = 1$.
- 2) На вход D_2 сдвигового регистра подается тестовая последовательность (например, бегущая 1, затем — бегущий 0). Выходная последовательность регистра сравнивается с эталоном (при необходимости ее можно сжать в СА). Выходы КЧ при этом не анализируются.

- 1) Устанавливается тестовый режим: $G = 1$, при этом триггеры отключаются от КЧ. В регистр заносится тестовый набор для проверки КЧ (например, очередной вектор ПС-последовательности). На входах x КЧ устанавливается внешний тестовый набор, дополняющий тест со стороны регистра.
- 2) Устанавливается рабочий режим: $G = 0$, при этом триггеры подключаются к КЧ.
- 3) Включаются внешние выходы z КЧ и с них считываются реакции на тестовые наборы, установленные слева и снизу.
- 4) Подается один синхроимпульс, и реакции КЧ считываются вниз, заменяя содержимое D -триггеров.
- 5) Устанавливается тестовый режим: $G = 1$. Подаются синхроимпульсы, и обновленные данные регистра (т.е. реакции КЧ вниз) считываются и сравниваются с эталоном.

166

13.3. Разновидности метода сканирования

1. Метод сквозного сдвигового регистра *LSSD* (Level-sensitive Scan Design — сканирование, чувствительное к уровню синхронизирующих сигналов). Реализация отличается от описанной использованием триггеров-защелок, управляемых уровнем (а не фронтом) синхроимпульсов и обеспечивающих отсутствие состязаний.

2. Метод сканирования на основе многофункционального сдвигового регистра *BILBO* (Built-in Logic Block Observation — встроенные средства управления и наблюдения состояния объекта). Используется в СБИС со встроенными средствами диагностирования, которые строятся на основе базовых элементов *BILBO*, каждый из которых имеет 2 управляющих входа G_1 и G_2 , сигналы на которые подаются от внешнего управляющего устройства и могут принимать следующие значения:

$G_1 = 0, G_2 = 1$ — обнуление всех триггеров *BILBO*;

$G_1 = 1, G_2 = 1$ — установка рабочего режима кристалла;

$G_1 = 0, G_2 = 0$ — установка режима сканирования: все базовые элементы *BILBO* объединяются в единый сдвиговый регистр;

$G_1 = 1, G_2 = 0$ — в сдвиговом регистре включаются обратные связи и устанавливается: в верхнем регистре — режим генерации тестов (генератор псевдослучайной последовательности — ГПП), в нижнем регистре — режим СА.

Структурная схема кристалла приведена на рис. 13.8.

Базовые регистры B_1 и B_2 выполнены на базовых элементах *BILBO*.

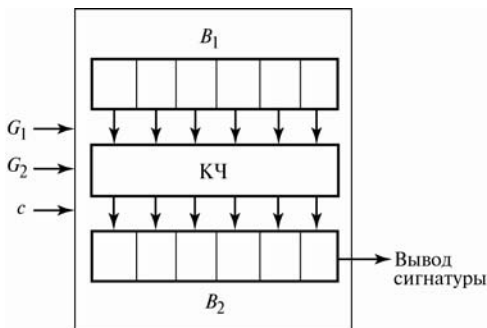


Рис. 13.8

На вход «с» подаются синхроимпульсы. Комбинационная часть (КЧ) объекта располагается между регистрами.

Порядок работы:

1) Рабочий режим. При $G_1 = 1$, $G_2 = 1$ кристалл функционирует в рабочем режиме.

2) Тестовый режим.

а) Обнуление регистров: $G_1 = 0$, $G_2 = 1$.

б) B_1 устанавливается в режим ГПП, B_2 — в режим СА ($G_1 = 1$, $G_2 = 0$).

в) В B_1 заносится ненулевой код и начинается тестирование КЧ, в B_2 сжимаются реакции при подаче синхроимпульсов «с».

г) Устанавливается режим сканирования ($G_1 = 0$, $G_2 = 0$). При этом оба регистра объединяются в сквозной сдвиговый регистр и отключаются от КЧ. В конце тестирования в регистрах B_1 и B_2 остается некоторый код, эталонное значение которого заранее известно. Эти коды сдвигаются и выводятся на анализ (сравнение с эталоном). Результат «Негоден» свидетельствует о неисправности КЧ или регистровой логики.

3. Метод граничного сканирования *BST* (Boundary Scan Testing) и JTAG-интерфейс. Встроенные в кристалл средства сканирования (*BILBO*) обеспечивают высокую эффективность проверки, но удорожают изделие. Далеко не все БИС и СБИС содержат такие средства. Обычно это заказные кристаллы. Для тестирования СБИС было предложено в корпус каждого кристалла (но не в сам кристалл) встраивать дополнительные ячейки — триггеры *BSC* (Boundary Scan Cells), связанные с каждым выводом БИС. Эти ячейки принимают тестовые сигналы от внешнего проверяющего устройства и реакции со стороны БИС и могут соединяться в единую последовательную цепочку (сдвиговый регистр). Такая архитектура отличается большой гибкостью и получила название *метода граничного сканирования* — *BST* (Boundary Scan Testing), поскольку дополнительные ячейки (*BSC*) располагаются в корпусе на границе кристаллов. Использование граничного сканирования позволяет проверять отдельные или все БИС, расположенные на плате, и их соединения. Однако для широкого использования *BST* потребовались организационные меры, направленные на унификацию и определяющие единый набор управляющих сигналов, и единая аппаратная организация. Поэтому был разработан международный стандарт *IEEE 11.1* (или *IEEE 11.1a*), получивший название

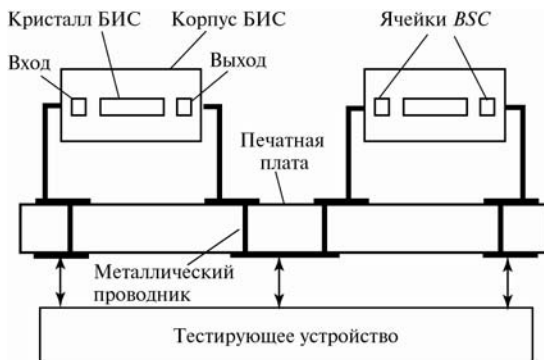


Рис. 13.9

JTAG-интерфейс (Joint Test Action Group — объединенная стандартизованная группа средств для тестирования). В настоящее время многие производители микросхем выпускают изделия, поддерживающие *JTAG-интерфейс*. Он используется при контроле и отладке программируемых логических интегральных схем (ПЛИС), процессоров, системных узлов плат и т.д. Интерфейс связывает внешнее тестирующее устройство (например, ПЭВМ) с тестируемым объектом (печатной платой) (рис. 13.9). В частности, сигналы *JTAG* могут передаваться по шине PCI вычислительных машин.

Тестирование выполняется программными средствами ПЭВМ, которая генерирует тесты и анализирует результаты проверки, и встроенным устройством управления *JTAG-интерфейса* (УУ).

УУ предназначено для интерпретации поступающих команд и управления процессом граничного сканирования. Оно содержит следующие аппаратные средства:

- 1) тестовый порт доступа (*TAP* – Test Access Port), соединяющий проверяемое устройство с тестирующим оборудованием;
- 2) контроллер порта (*TAP-контроллер*), управляющий процессом сканирования;
- 3) регистр команд;
- 4) регистры тестовых данных.

Порт (*TAP*) имеет четыре контакта, через которые передаются четыре вида интерфейсных сигналов:

- 1) сигналы синхронизации последовательных данных (*TCK* – Test Clock);
- 2) сигналы выбора тестового режима (*TMS* – Test Mode Select);

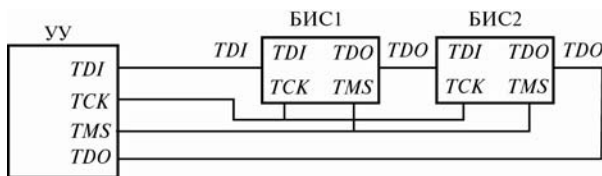


Рис. 13.10

- 3) входные данные в объект и команды в последовательном двоичном коде (*TDI* – Test Data Input);
- 4) выходные данные от объекта тестирования (БИС) в тестирующее устройство в последовательном двоичном коде (*TDO* – Test Data Output).

Основой УУ является *TAP*-контроллер, управляющий поведением *BSC*-ячеек в зависимости от значения поступающего из порта интерфейсного сигнала *TMS*.

Встроенные в УУ регистры команд и данных — независимо работающие сдвиговые регистры. Регистр данных передает данные на контакты *TDI* и принимает информацию с контакта *TDO* после выполнения команд в *JTAG*-цепочках.

При наличии нескольких БИС на плате они объединяются в последовательную *JTAG*-цепочку, управляемую общим УУ (рис. 13.10).

Стандартом *JTAG* предусматривается также использование (при необходимости) средств самотестирования БИС, встроенных в кристалл, по командам, инициированным внешним устройством (ПЭВМ и контроллером), поступающим через порт доступа к *BSC*.

Для реализации метода граничного сканирования предусмотрены несколько режимов и соответствующих команд *JTAG*-интерфейса:

- 1) Режим самотестирования СБИС внутренними средствами без ввода внешних данных. Результаты поступают в ячейки *BSC*. Для этого используется команда *RUNBIST*.
- 2) Ввод данных *TDI* через *JTAG*-цепочку *BSC* во внутренние ячейки кристалла для выполнения тестирования встроенными средствами БИС и вывод результатов в ячейки *TDO*. Для инициализации проверки используется команда *INTTEST*.
- 3) Режим проверки внешних соединений БИС на плате, минуя внутрисхемную логику (команда *EXTEST*).
- 4) Режим проверки штатной работы БИС путем сканирования и наблюдения данных, поступающих из БИС в ячейки *BSC* в заданные моменты времени (команда *SAMPLR / PRELOAD*).

В пределах каждого режима используются внутрирежимные команды.

Унификация аппаратно-программных средств, поддерживающих JTAG-интерфейс, и их незначительная избыточность, привела к широкому использованию метода сканирования в практике проверки и отладки плат, содержащих БИС и СБИС.

13.4. Повышение диагностируемости ВС на структурном уровне

Пусть два процессора A и B тестируют друг друга, а процессор C (априорно исправный) наблюдает реакции и сравнивает их с эталонами (рис. 13.11, a). Модуль C является начальным ядром и используется для начала раскрутки. Однако при получении результата «Негоден» он не сможет определить, какой из наблюдаемых процессоров неисправен. Для определения этого он, кроме наблюдения реакций, должен управлять их поведением. Тогда он будет знать рабочие состояния наблюдаемых объектов и их нормальное поведение при переходах в различные состояния. Таким образом, для правильного диагностирования необходимо и достаточно, чтобы каждый объект управлялся и наблюдался другими объектами, которые ранее были уже проверены и оказались исправными, т.е. объектами более высокого ранга. Для анализа удобно использовать *граф управляемости / наблюдаемости (у/н)*, в котором вершины отображают анализируемые объекты, а дуги — подачу управляющих сигналов (тестов) и наблюдения реакций. Если объект A управляет объектом B , то дуга направлена в сторону B и обозначается двойной стрелкой, а если A наблюдает реакции объекта B , то дуга обозначается одиночной стрелкой и также направлена в сторону B . Отметим, что поток информации при передаче наблюдаемых сигналов направлен в противоположную сторону — против стрелки.

Особенностью графа, изображенного на рис. 13.11, a , является наличие контуров. Это недопустимо, так как невозможно выделить вершину более высокого ранга среди модулей A и B . Поэтому необходимо устранить обратные связи, образованные как стрелками управляемости или наблюдаемости, так и смешанными стрелками.

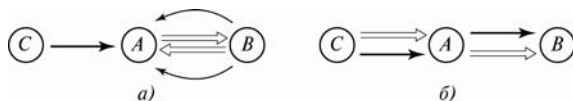


Рис. 13.11

На рис. 13.11, б изображен правильный граф, обеспечивающий условие тестируемости системы. В нем добавлена стрелка управляемости от C к A и устранены лишние стрелки управляемости и наблюдаемости между модулями. Здесь уже модуль C является ядром, способным начать проверку системы.

Отметим, что обратные связи имеют место только в сильносвязных компонентах графа G . Подграф графа G называется максимально сильносвязным (*МСС-подграфом*), если он содержит максимальное число сильносвязных вершин и не содержит других вершин. Для обеспечения условий у/н нужно сначала найти все МСС-подграфы и затем в некоторых из них (или во всех) разорвать некоторые (или все) контуры в зависимости от требуемой глубины диагностирования. После этого вершины могут быть правильно проранжированы и добавлены недостающие дуги у/н. Отметим также, что к отдельным вершинам могут идти несколько дуг управляемости или наблюдаемости, охватывающие различные части тестируемого блока.

Процедура обеспечения управляемости / наблюдаемости:

- 1) По структурной схеме ВС строится граф у/н G .
- 2) Выделяются все МСС-подграфы и представляются в виде псевдовершин W . Эта процедура называется *конденсацией графа*. Проверятся условия у/н, включая псевдовершины. Если условия у/н не выполнены, то добавляются недостающие дуги от вершин более высокого ранга.
- 3) Проверяется достаточность глубины диагностирования. Если глубина достаточна, то переходим к шагу 4, если нет, то выбирается псевдовершина самого высокого ранга и соответствующий МСС-подграф. Выбирается одна из входных вершин МСС-подграфа и для нее выполняется разрыв глобальных (наиболее глубоких) обратных связей. Таким же образом разрываются другие контуры внутри выбранного подграфа вплоть до принятого уровня детализации. После разрыва контуров соответствующие вершины подключаются дугами у/н к вершинам более высокого ранга. Переходим к шагу 2.
- 4) Вершины одинакового ранга группируются в одном конструктивном блоке. Если это невозможно, объединяются вершины i -го ранга с вершинами $(i - 1)$ -го ранга или с вершинами $(i + 1)$ -го ранга, но не с обеими группами одновременно (чтобы не нарушить ранжировку). Модули оставшихся МСС-подграфов размещаются в

одном конструктивном блоке. Это обеспечивает глубину диагностирования с точностью до одного конструктивного блока.

Пример 13.1. На рис. 13.12, *а* изображен исходный граф у/н (шаг 1) и выделен единственный ММС-подграф.

На рис. 13.12, *б* МСС-подграф заменен псевдовершиной *W* и к вершине *K* добавлена недостающая дуга управляемости от одной из вершин более высокого ранга (шаг 2). Вершины располагаются в соответствии с их рангами.

На рис. 13.12, *в* изображен полученный граф, в котором разорвана дуга управляемости, идущая от входной вершины *D* МСС-подграфа к вершине *G* для ликвидации контура, и добавлена дуга управляемости к вершине *G* от вершины более высокого ранга; после этого выделен новый МСС-подграф (шаг 3).

На рис. 13.12, *г* данный подграф заменен псевдовершиной (шаг 2). Вершины располагаются в соответствии с новой ранжировкой.

На рис. 13.12, *д* в новом подграфе разорван контур путем удаления дуги от вершины *H* к вершине *D*, добавлена дуга наблюдаемости от *C* к *D* и дуга управляемости к вершине *F* (шаг 3). В полученном графе обратные связи отсутствуют и проведена ранжировка (уровни $L_1 — L_7$).

Результаты ранжировки сводим в таблицу:

Ранги						
L_1	L_2	L_3	L_4	L_5	L_6	L_7
<i>A</i>	<i>C</i>	<i>D</i>	<i>F</i>	<i>E</i>	<i>H</i>	<i>J</i>
<i>B</i>	<i>G</i>			<i>I</i>		<i>K</i>
						<i>L</i>

Предположим, что *C* и *G* невозможно разместить в одном конструктиве. Тогда переносим блок *C* в конструктив соседнего уровня L_3 и получаем новую таблицу размещения блоков:

Ранги						
L_1	L_2	L_3	L_4	L_5	L_6	L_7
<i>A</i>		<i>C</i>	<i>F</i>	<i>E</i>	<i>H</i>	<i>J</i>
<i>B</i>	<i>G</i>	<i>D</i>		<i>I</i>		<i>K</i>
						<i>L</i>

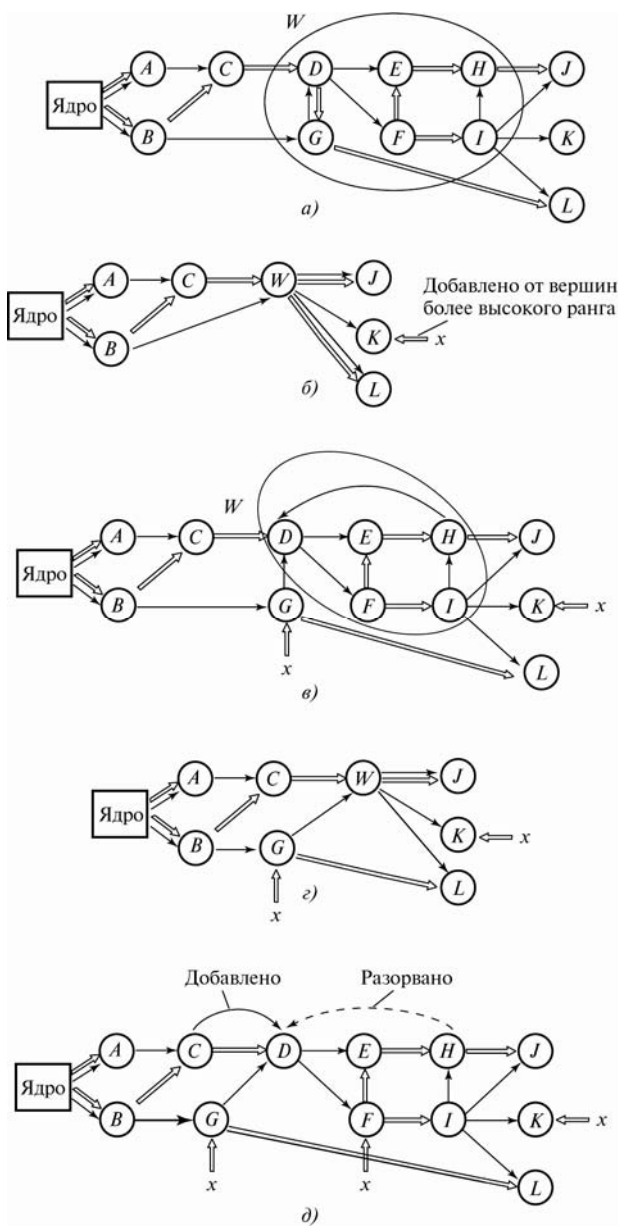


Рис. 13.12

13.5. Конденсация графа управляемости / наблюдаемости и выделение МСС-подграфов

Наиболее трудоемкой задачей на этапе структурного обеспечения условий у/н является выделение МСС-подграфов. Такая задача носит название *конденсации графа* и выполняется в несколько этапов. Рассмотрим пошагово процедуру конденсации на конкретном примере.

1. Исходный граф у/н представляется в виде структурного орграфа, т.е. дуги управляемости и дуги наблюдаемости представляются одинаково — в виде ориентированных дуг e_{ij} : символы стрелок $\{\Rightarrow, \rightarrow\}$ заменяются на символ $\{\rightarrow\}$.

Это связано с тем, что не имеет значения, какими дугами образуются контуры.

Пусть в результате получился граф, приведенный на рис. 13.13.

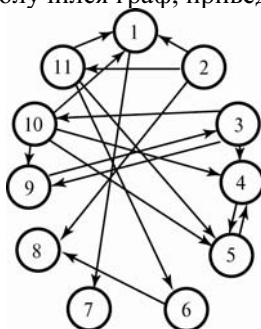


Рис. 13.13

2. Строится *матрица смежности*. Поскольку считается, что вершины связаны сами с собой, в диагонали матрицы добавляются единицы. Получаем *матрицу непосредственных связей* $\mathbf{M} = [e_{ij}]_{n \times n}$:

$$M = [e_{ij}] =$$

	1	2	3	4	5	6	7	8	9	10	11
1	1						1				
2	1	1						1			1
3			1	1					1		1
4				1	1						
5				1	1						
6						1		1			
7							1				1
8								1			
9			1						1		
10	1			1	1				1	1	
11	1				1	1					1

Рис. 13.14

3. Строится матрица достижимости $\mathbf{M}_D = \mathbf{M}^L = [e_{ij}^{(L)}]_{n \times n}$, где $L \geq n - 1 = 11 - 1 = 10$.

Матрица непосредственных связей \mathbf{M} определяет множество однозвенных путей: единица в ячейке (ij) свидетельствует о наличии дуги в графе между соответствующими вершинами. Возведя в квадрат матрицу однозвенных путей \mathbf{M}^2 , получим матрицу двухзвенных или однозвенных путей. Соответственно, \mathbf{M}^3 — матрица трехзвенных, двухзвенных или однозвенных путей и т. д. Таким образом, если в матрице \mathbf{M}^L элемент $[e_{ij}^{(L)}] = 1$, то существует хотя бы один путь от вершины i к вершине j длины L или менее. Но самый длинный путь в графе, содержащем n вершин, имеет длину, не превышающую $n - 1$. Дальнейшее возведение в степень не изменяет матрицу. Такая матрица $\mathbf{M}_D = [e_{ij}^{(n-1)}]$ называется *стабильной*. Но стабильная матрица может получиться и при меньшем значении степени L . Поэтому для получения стабильной матрицы обычно выполняется последовательное возведение \mathbf{M} в степень 2, затем 4, 8, 16 и т. д. Если в процессе возведения в степень соседние матрицы совпадают, то получена стабильная матрица.

Выполним данную процедуру для рассматриваемого примера (рис. 13.15).

$$\mathbf{M}_D = [e_{ij}^{(n-1)}] =$$

	1	2	3	4	5	6	7	8	9	10	11
1	1			1	1	1	1	1			1
2	1	1		1	1	1	1	1			1
3	1		1	1	1	1	1	1	1	1	1
4				1	1						
5				1	1						
6						1		1			
7	1			1	1	1	1	1			1
8								1			
9	1		1	1	1	1	1	1	1	1	1
10	1		1	1	1	1	1	1	1	1	1
11	1			1	1	1	1	1			1

Рис. 13.15

Примечание: матрицу \mathbf{M}_D можно получить и непосредственно по графу путем просмотра наличия путей от i к j .

4. Строится матрица контрдостижимости \mathbf{M}_D^T путем инвертирования матрицы \mathbf{M}_D :

$$\mathbf{M}_D^T =$$

	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1				1		1	1	1
2		1									
3			1						1	1	
4	1	1	1	1	1		1		1	1	1
5	1	1	1	1	1		1		1	1	1
6	1	1	1			1	1		1	1	1
7	1	1	1				1		1	1	1
8	1	1	1			1	1	1	1	1	1
9			1						1	1	
10			1						1	1	
11	1	1	1				1		1	1	1

Рис. 13.16

5. Строится матрица взаимной достижимости \mathbf{M}_B , получаемая логическим перемножением чисел одноименных клеток $\mathbf{M}_B = \mathbf{M}_D \times \mathbf{M}_D^T$:

$$\mathbf{M}_B =$$

	1	2	3	4	5	6	7	8	9	10	11
1	1						1				1
2		1									
3			1						1	1	
4				1	1						
5				1	1						
6						1					
7	1						1				1
8								1			
9			1						1	1	
10			1						1	1	
11	1						1				1

Рис. 13.17

6. Матрица \mathbf{M}_B позволяет построить все МСС-подграфы и не входящие в них отдельные вершины. Для этого используется следующий алгоритм.

- Выбирается любая непросмотренная k -я строка матрицы \mathbf{M}_B и перечисляются все покрываемые ею столбцы. Получаем множество $M(k)$.
- Выбирается любая i -я строка, $i \notin M(k)$, и создается множество $M(i)$.
- Выбирается любая j -я строка $j \notin \{M(k) \cup M(i)\}$ и строится множество $M(j)$ и т. д.

В результате получаем:

№ строки	Подмножество
1	$M(1) = \{1, 7, 11\}$ — МСС-подграф (псевдовершина A)
2	$M(2) = \{2\}$ — отдельная вершина (вершина B)
3	$M(3) = \{3, 9, 10\}$ — МСС-подграф (псевдовершина F)
4	$M(4) = \{4, 5\}$ — МСС-подграф (псевдовершина E)
6	$M(6) = \{6\}$ — отдельная вершина (вершина D)
8	$M(8) = \{8\}$ — отдельная вершина (вершина C)

7. По исходному графу устанавливаются связи между полученными подмножествами вершин (рис. 13.18):

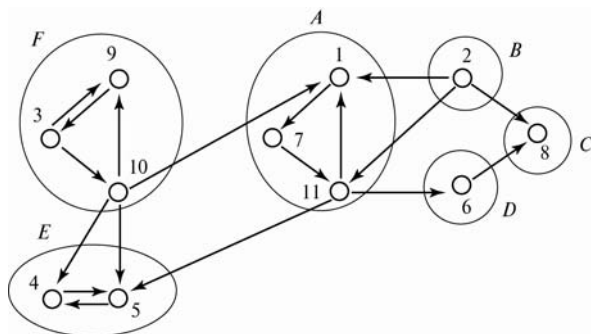


Рис. 13.18

8. Строится конденсированный граф (рис. 13.19) в виде комбинационной сети (без обратных связей). Выполняется ранжирование вершин (рис. 13.20).

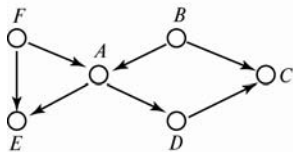


Рис. 13.19

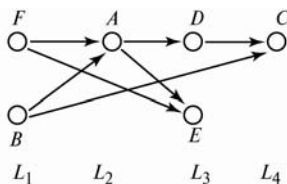


Рис. 13.20

9. Полученный конденсированный структурный граф вновь преобразуется в граф у/н путем замены его дуг на дуги исходного графа управляемости/наблюдаемости (символ $\{\rightarrow\}$ заменяем на $\{\Rightarrow, \rightarrow\}$).

После этого можно выполнять разрыв обратных связей.

14. ПОСТРОЕНИЕ САМОПРОВЕРЯЕМЫХ СВТ

14.1. Схемы самопроверяемого дублирования

Сначала рассмотрим схему встроенного контроля (СВК) без самопроверки. Рассмотрим схему дублирования, содержащую средства контроля. Она представляет собой схему нагруженного резервирования и схему сравнения СС (рис. 14.1). СВК включает в себя дубликат ОД и СС.

При отсутствии неисправности выход СС $f = 0$. При одиночной неисправности (либо в ОД, либо в дубликате) $f = 1$. После этого тестируются ОД и дубликат, и выполняется отключение неисправного канала. В дальнейшем будем считать, что возможны только одиночные неисправности.

Если имеется несколько выходов ОД ($\mathbf{F} = F_1, F_2, \dots, F_m$), все они сравниваются компаратором с выходами дубликата ($\mathbf{F}' = F'_1, F'_2, \dots, F'_m$) (рис. 14.2).

Недостатком такой схемы является невозможность обнаружения неисправности $f^* \equiv 0$ на выходе f самого компаратора, так как будет получен результат «Годен». Это связано с наличием в СС только одного выхода.

Рассмотрим теперь самопроверяемые СВК (ССВК) для объектов с несколькими выходами F_1, F_2, \dots . Дубликат при этом содержит соответствующее число выходов F'_1, F'_2, \dots .

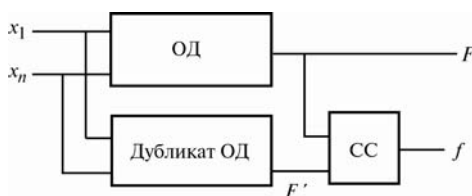


Рис. 14.1

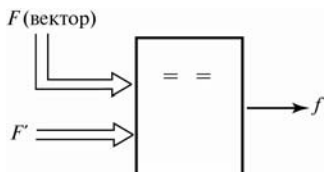


Рис. 14.2

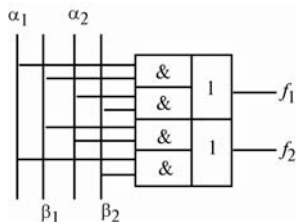


Рис. 14.3

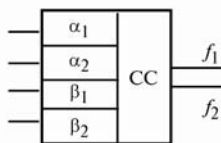


Рис. 14.4

СВК называется самопроверяемой, если она обнаруживает неисправность либо в ОД, либо в СВК (т.е. в дубликate и СС). Для этого СС должна содержать более одного выхода.

На рис. 14.3 приведена схема СС с 2 выходами f_1, f_2 и 4 входами $\alpha_1, \beta_1, \alpha_2, \beta_2$, а на рис. 14.4 — ее условное изображение. К схеме СС предъявляется требование $f_1 \neq f_2$ при отсутствии одиночной неисправности на входах или выходе и $f_1 = f_2$ при ее наличии. Покажем, что для нее это требование при определенных условиях выполняется.

Пусть на вход α_1 подается сигнал F_1 объекта, а на вход α_2 — сигнал дубликата F'_1 , соответственно на входы β_1 и β_2 подаются сигналы F_2 и F'_2 . Выходные сигналы СС равны

$$f_1 = \alpha_1 \beta_1 \vee \alpha_2 \beta_2,$$

$$f_2 = \alpha_2 \beta_1 \vee \alpha_1 \beta_2.$$

Если неисправность отсутствует, то должно быть

$$f_1 \neq f_2, \text{ или } f_1 \oplus f_2 = 1 (\text{истинно}).$$

Но

$$\begin{aligned} f_1 \oplus f_2 &= (\alpha_1 \beta_1 \vee \alpha_2 \beta_2) \oplus (\alpha_2 \beta_1 \vee \alpha_1 \beta_2) = \\ &= (\alpha_1 \beta_1 \oplus \alpha_2 \beta_2 \oplus \alpha_1 \beta_1 \alpha_2 \beta_2) \oplus (\alpha_2 \beta_1 \oplus \alpha_1 \beta_2 \oplus \alpha_1 \beta_1 \alpha_2 \beta_2) = \\ &= \alpha_1 \beta_1 \oplus \alpha_2 \beta_2 \oplus \alpha_2 \beta_1 \oplus \alpha_1 \beta_2 = \alpha_1 (\beta_1 \oplus \beta_2) \oplus \alpha_2 (\beta_1 \oplus \beta_2) = \\ &= (\alpha_1 \oplus \alpha_2) (\beta_1 \oplus \beta_2). \end{aligned}$$

Это выражение истинно (равно 1), если одновременно $\alpha_1 \oplus \alpha_2 = 1$ и $\beta_1 \oplus \beta_2 = 1$, т.е. $F_1 \neq F'_1$ и $F_2 \neq F'_2$. Отсюда следует, что выходы дубликата должны иметь инверторы.

На рис. 14.5 дана схема подключения каскадов СС к многовыходному объекту и дубликату. Выходы дубликата F' выполнены инверсными по отношению к выходам ОД F . Если неисправности нет, то $F' \neq F$, и на любом каскаде СС $\alpha_1 \neq \alpha_2, \beta_1 \neq \beta_2, f_1 \neq f_2$. При наличии неисправ-

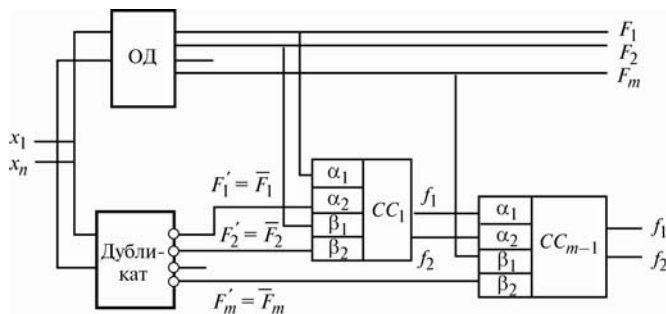


Рис. 14.5

ности в ОД или дубликате значения одной или нескольких пар выходов F и F' совпадут, при этом на каком-то каскаде СС будет $f_1 = f_2$, и это проявится на последнем каскаде.

Достоинства схемы дублирования:

- 1) Универсальность (дублирование применимо как для комбинационных, так и для последовательностных схем).
- 2) Выполняется резервирование, что повышает надежность.
- 3) Любая одиночная неисправность на входах или выходах СС приводит к $f_1 = f_2$ и обнаруживается, т.е. выполняется условие самопроверяемости.

Недостаток — необходимость иметь полный дубликат ОД.

14.2. Проектирование ССВК комбинационной логики

Здесь возможно упрощение схемы дубликата. Структурная схема ОД и ССВК приведена на рис. 14.6. В схеме звездочкой (*) обозначены входы схем $M2$, идущие от независимых выходов ОД. Выходы называются независимыми, если на любом входном наборе и при любой неисправности активизируется только один из этих выходов либо ни один выход не активизируется. Независимые выходы F_i объединяются схемами свертки $M2$, благодаря чему уменьшается количество СС (с $m - 1$ до $h - 1$). Соответственно уменьшается необходимое число выходов F'_k дубликата (с m до h , $h < m$). При этом вместо полного дубликата может быть использована его упрощенная модель — дополнительное дискретное устройство (ДДУ), выполняющее все рабочие функции ОД, но имеющее более простую схемную реализацию.

На рис. 14.6 использованы следующие обозначения: звездочками * помечены входы схем сверток $M2$, идущие от независимых выходов

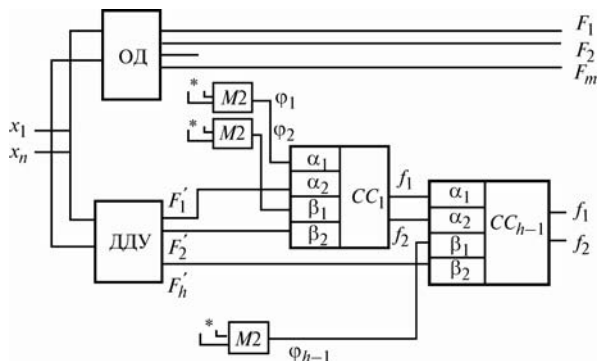


Рис. 14.6

F_i (свертки могут иметь несколько входов, т.е. быть составными); $\{M2\}$ и $\{CC\}$ — решающий орган (РО); ДДУ и РО образуют ССВК; ϕ_k — выходы $M2$; f_1, f_2 — выходы CC ; $F'_k = \phi_k$ (вместо $F'_k = \bar{F}_k$, как было в схеме дублирования).

Синтез ССВК предполагает решение двух задач:

- 1) определение независимых выходов ОД; это позволяет уменьшить число выходов h ДДУ и сократить число сумматоров $M2$ и CC ;
- 2) синтез экономичного ДДУ.

Для определенности рассмотрим модельный пример — комбинационную схему (объект диагностирования) (рис. 14.7), на примере которой проиллюстрируем процесс проектирования ССВК. Элементы схемы пронумерованы в соответствии с ранжировкой, выполненной независимо для каждого выхода F_i . Индексация внутренних переменных y_k (выходов элементов) определяется номером соответствующего блока.

Схема выполняет следующие логические функции:

$$F_1 = x_1 x_2 \vee x_1 x_3 \vee x_2 x_3, \quad F_2 = x_1 x_4 \vee x_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3, \quad F_3 = x_1 x_2 \wedge x_3 \bar{x}_4$$

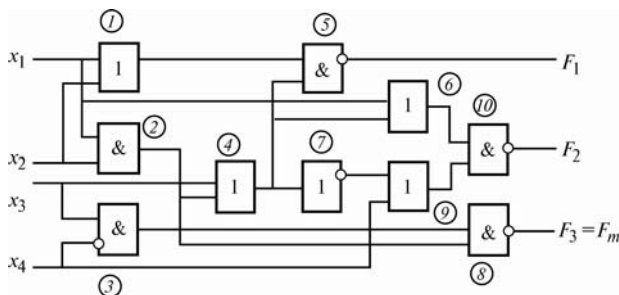


Рис. 14.7

14.2.1. Решение 1-й задачи — определение независимых выходов ОД (4 этапа)

1-й этап — построение обобщенного графа соединений ОД.

На этом этапе выполняется декомпозиция ОД с выделением явно независимых выходов. Кроме того, на этом этапе находится сокращенный набор внутренних переменных y_k , ответственных за активизацию определенных выходов ОД. Это позволяет избежать полного перебора неисправностей цепей, использовавшегося ранее при синтезе тестов.

Для построения обобщенного графа используется следующий алгоритм.

1) Строится *исходный граф соединений*. Вершины графа — логические элементы, дуги — электрические связи. При этом точки разветвления, а также входные и выходные полюса не учитываются.

2) На основе графа соединений строится *матрица соединений*. Строки и столбцы матрицы соответствуют номерам логических элементов. В матрице проставляется 1, если имеется связь между соответствующими элементами, или 0, если связи нет. В правом (дополнительном) столбце матрицы проставляется 1, если соответствующий элемент схемы (т.е. номер строки) является выходом ОД. Этот столбец называется *столбцом выходов* (СВ). Крайняя нижняя строка и крайний правый столбец использованы только для пометок.

Приведем для рассматриваемого примера матрицу соединений (табл. 14.1).

Таблица 14.1

	1	2	3	4	5	6	7	8	9	10	СВ	
1					1							
2				1				1				
3								1				
4					1	1	1					
5											1	
6										1		*
7									1			(7)
8											1	*
9										1		(5)
10											1	*
						*	*		*	*		(3)
						(8)	(6)		(4)	(2)		(1)

3) Помечается (знаком «*») одна из строк, имеющая 1 в столбце СВ (выходной элемент).

4) Помечаются столбцы, имеющие номера помеченных строк.

5) Помечаются строки, имеющие 1 только в помеченных столбцах и не имеющих 1 в непомеченных столбцах. Если такие строки есть, то — переход к шагу 4, если нет — то к шагу 6.

(В правом столбце и нижней строке пометки «*» пронумерованы в порядке их появления.)

6) Номера помеченных строк объединяются в одно подмножество. В этом подмножестве подчеркивается элемент, выбранный на шаге 3 (выходной элемент). Если подмножество состоит только из одного элемента, то он тоже подчеркивается.

В исходной таблице имеем подмножество {6, 7, 9, 10}.

7) Удаляются помеченные строки и столбцы. При этом, если в удаляемом столбце стоит 1 в непомеченной строке, то в СВ этой строки проставляется дополнительная 1. Если в таблице остались невычеркнутые строки и столбцы, то переход к шагу 3, иначе — к шагу 8.

В табл. 14.1 при удалении столбца 6 замечаем, что в нем стоит 1 в непомеченной строке 4. Поэтому ставим 1 в столбце СВ табл. 14.2.

Таблица 14.2

	1	2	3	4	5	8	СВ	
1					1			
2				1		1		
3						1		* (3)
4					1		1	(добавлена 1)
5							1	
8							1	* (1)
			* (4)			* (2)		

В результате получаем 2-е подмножество {3, 8}.

Поскольку остались невычеркнутые строки и столбцы, то процесс продолжаем (табл. 14.3).

Таблица 14.3

	1	2	4	5	СВ	
1				1		*(3)
2			1		1	(добавлена 1)
4				1	1	
5					1	*(1)
	*(4)			*(2)		

Получаем 3-е подмножество {1, 5}.

В таблице в столбце 5 стоит 1 в непомеченной строке 4, но там уже имеется 1. Поэтому процесс продолжаем дальше (табл. 14.4)

Таблица 14.4

	2	4	СВ	
2		1	1	
4			1	*(1)
		*(2)		

Получаем подмножество $\{4\}$.

Удаляем 4-ю строку и 4-й столбец (табл. 14.5).

Таблица 14.5

	2	СВ	
2		1	*(1)
	*(2)		

В результате получаем последнее подмножество: $\{2\}$.

8) Строится обобщенный граф соединений:

а) *Построение вершин обобщенного графа.* Каждому выделенному подмножеству строк (элементов схемы) сопоставляется вершина обобщенного графа, которая объединяет все элементы подсхемы (среди них находится и одна *подчеркнутая* — *выходная вершина группы*). Если подчеркнутая вершина является выходом ОД, то соответствующая вершина обобщенного графа обводится двойным кружком, если же не является выходной, то вершина изображается одиночным кружком.

б) *Построение дуг обобщенного графа.* Если подчеркнутый элемент вершины i обобщенного графа непосредственно соединен хотя бы с одним элементом другой вершины j , то в обобщенном графе проводится дуга между соответствующими вершинами.

Результатом построения обобщенного графа является разбиение исходной схемы на некоторое множество одновыходных подсхем. В данном случае имеем 5 подсхем (рис. 14.8) (номера вершин графа, отображающие выделенные подсхемы, и выходы F_i исходной схемы проставлены рядом с вершинами графа).

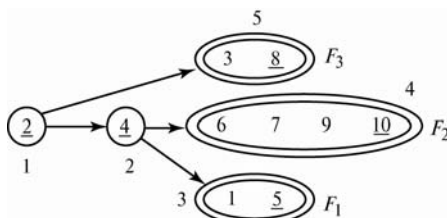


Рис. 14.8

Теорема 14.1. Если одиночная неисправность проявляется на выходе F_i , то она проявляется и на выходе одной из вершин обобщенного графа, связанных с F_i (т.е. на выходе подчеркнутого элемента соответствующей подсхемы).

Другими словами, выходы подсхем ответственны за неисправности внутри них. Поэтому вместо перебора всех возможных неисправностей исходной схемы можно рассматривать только неисправности на выходах подсхем, отображаемых вершинами обобщенного графа.

2-й этап — выделение графически независимых выходов устройства.

Введем следующие определения.

1) Вершина i обобщенного графа называется *графически существенной* для вершины j , если существует путь от i к j .

2) Выходы F_i и F_j называются *графически независимыми*, если они не имеют общих графически существенных вершин.

На рис. 14.8 вершина 1 графически существенна для вершин 2, 3, 4, 5, вершина 2 графически существенна для вершин 3 и 4. Выходы F_1 , F_2 , F_3 имеют общую графически существенную вершину 1. Поэтому все выходы графически зависимы.

Если F_i и F_j графически независимы, то они действительно независимы между собой; однако если F_i и F_j графически зависимы, то эти выходы могут быть как зависимыми, так и независимыми. Другими словами, графическая независимость является достаточным, но не является необходимым условием истинной независимости выходов. Однако выполнение 2-го этапа обычно существенно сокращает объем расчетов на 3-м (следующем) этапе при наличии графически независимых выходов.

3-й этап — определение пар независимых и зависимых выходов объекта в пределах группы графически зависимых выходов.

Выходы $\{F_i, F_j, F_k, \dots\}$ являются *независимыми*, если независима любая пара этих выходов. *Независимость пар* понимается как независимость активизации соответствующих выходов при появлении любой неисправности в любой внутренней точке y_k или на внешнем входе объекта. Другими словами, F_i и F_j независимы, если при появлении любой одиночной неисправности активизируются пути, ведущие только к одному из этих выходов. Но активизация выхода происходит, если и только если соответствующая булева производная равна 1. Это означает, что F_i и F_j независимы при неисправности y_k^* и любом наборе входных переменных объекта, если выполняется одно из следующих условий:

1) активизируется F_i и не активизируется F_j , т.е.

$$dF_i(y_k)/dy_k = 1 \text{ и } dF_j(y_k)/dy_k = 0,$$

2) активизируется F_j и не активизируется F_i , при этом

$$dF_i(y_k)/dy_k = 0 \text{ и } dF_j(y_k)/dy_k = 1,$$

3) не активизируется ни F_i , ни F_j , т.е.

$$dF_i(y_k)/dy_k = dF_j(y_k)/dy_k = 0.$$

Отсюда следует условие независимости F_i и F_j при неисправности y_k^* :

$$dF_i(y_k)/dy_k \cdot dF_j(y_k)/dy_k \equiv 0.$$

Условие же независимости F_i и F_j при любой одиночной неисправности (y_k^*, y_l^*, \dots) имеет вид:

$$dF_i(y_k)/dy_k \cdot dF_j(y_k)/dy_k \vee dF_i(y_l)/dy_l \cdot dF_j(y_l)/dy_l \vee dF_i(y_s)/dy_s \times \\ \times dF_j(y_s)/dy_s \vee \dots \equiv 0.$$

Проверим выполнение этого тождества для всех пар выходов в каждой графически зависимой группе (в рассматриваемом примере такая группа выходов единственная — $\{F_1, F_2, F_3\}$).

Рассматриваем пару цепей y_2 и y_4 , неисправности в которых отвечают за все неисправности в группах (вершинах обобщенного графа) 1 и 2. Цепи y_{10} , y_8 и y_5 не рассматриваются, поскольку это выходы объекта: F_3 , F_2 и F_1 .

Для получения булевых производных сначала составляем для исходной схемы (рис. 14.7) функции $F_i(y_k)$ от входных переменных и рассматриваемой внутренней переменной y_k (табл. 14.6):

Таблица 14.6

$F_1(y_2) = \overline{(x_1 \vee x_2)}(y_2 \vee x_3)$	$F_1(y_4) = \overline{(x_1 \vee x_2)}y_4$
$\overline{F_2(y_2)} = \overline{(y_2 \vee x_1 \vee x_3)(y_2 \vee x_3 \vee x_4)}$	$\overline{F_2(y_4)} = (y_4 \vee x_1)(\overline{y_4} \vee x_4)$
$F_3(y_2) = y_2 x_3 \overline{x_4}$	—

Функция $F_3(y_4)$ не рассматривается, поскольку F_3 графически не зависит от y_4 и $F_3(y_4)/dy_4 \equiv 0$ (в силу свойства булевой производной).

Вычислим производную

$$\begin{aligned} dF_1 / dy_2 &= \overline{(x_1 \vee x_2)(0 \vee x_3)} \cdot (x_1 \vee x_2) \vee \overline{(0 \vee x_3)} \cdot \overline{(x_1 \vee x_2)(1 \vee x_3)} = \\ &= \bar{x}_3(x_1 \vee x_2) \vee 0 = x_1 \bar{x}_3 \vee x_2 \bar{x}_3. \end{aligned}$$

Аналогично рассчитываются другие производные (табл. 14.7).

Таблица 14.7

$dF_1 / dy_2 = x_1 \bar{x}_3 \vee x_2 \bar{x}_3$	$dF_1 / dy_4 = x_1 \vee x_2$
$dF_2 / dy_2 = \bar{x}_1 \bar{x}_3 x_4 \vee x_1 \bar{x}_3 \bar{x}_4$	$dF_2 / dy_2 = \bar{x}_1 x_4 \vee x_1 \bar{x}_4$
$dF_3 / dy_2 = x_3 \bar{x}_4$	$dF_3 / dy_4 \equiv 0$

Теперь составим уравнения для проверки независимости пар выходов $\{F_1, F_2\}$, $\{F_1, F_3\}$, $\{F_2, F_3\}$:

$$\begin{aligned} 1) dF_1 / dy_2 \cdot dF_2 / dy_2 \vee dF_1 / dy_4 \cdot dF_2 / dy_4 &= \\ &= (x_1 \bar{x}_3 \vee x_2 \bar{x}_3)(\bar{x}_1 \bar{x}_3 x_4 \vee x_1 \bar{x}_3 \bar{x}_4) \vee (x_1 \vee x_2)(\bar{x}_1 x_4 \vee x_1 \bar{x}_4) = \\ &= \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 x_4 \vee x_1 \bar{x}_4 = \bar{x}_1 x_2 x_4 \vee x_1 \bar{x}_4. \end{aligned}$$

Эта функция не равна тождественно 0 (например, при $x_1 = 0$, $x_2 = 1$, $x_4 = 1$ она равна 1). Поэтому выходы F_1 и F_2 зависимы.

$$2) dF_1 / dy_2 \cdot dF_3 / dy_2 \vee dF_1 / dy_4 \cdot dF_3 / dy_4.$$

Второе слагаемое тождественно равно 0 в силу того, что $dF_3 / dy_4 \equiv 0$, первое также тождественно равно 0: $dF_1 / dy_2 \cdot dF_3 / dy_2 = (x_1 \bar{x}_3 \vee x_2 \bar{x}_3) x_3 \bar{x}_4 \equiv 0$.

$$3) dF_2 / dy_2 \cdot dF_3 / dy_2 \vee dF_2 / dy_4 \cdot dF_3 / dy_4.$$

Второе слагаемое — тождественный 0, а первое равно $(\bar{x}_1 \bar{x}_3 x_4 \vee x_1 \bar{x}_3 \bar{x}_4) x_3 \bar{x}_4 \equiv 0$, т.е. тоже тождественный 0.

Отсюда вывод: $\{F_1, F_2\}$ — зависимая пара, $\{F_1, F_3\}$ — независимая пара и $\{F_2, F_3\}$ — тоже независимая пара.

Далее на этапе 4 по найденным парам независимых и зависимых выходов находятся полные группы независимых выходов.

4-й этап — определение полных групп независимых выходов.

Если в некотором множестве $\{F_i, F_j, F_k, \dots\}$ все пары независимы, то это множество образует *группу независимых выходов*. Задача состоит в минимизации числа групп независимых выходов (так как в этом случае минимизируется число выходов ДДУ и число схем сравнения СС).

Пример 14.1. Пусть имеется 3 выхода $\{1, 2, 3\}$, при этом $\{1, 2\}$ — зависимые выходы, $\{1, 3\}$ и $\{2, 3\}$ — попарно независимы. Тогда оптимальное решение может быть получено полным перебором вариантов группирования элементов 1, 2 и 3. Таких вариантов пять:

1) $\{1\}, \{2\}, \{3\}$; 2) $\{1\}, \{2, 3\}$; 3) $\{1, 2\}, \{3\}$; 4) $\{2\}, \{1, 3\}$; 5) $\{1, 2, 3\}$.

Варианты 3 и 5 — недопустимые решения, так как содержат группу зависимых выходов, остальные варианты — допустимые. Наименьшее число групп $N = 2$ имеют варианты 2 и 4.

Чтобы исключить полный перебор, используются оптимизационные методы решения этой задачи. Один из них основан на сведении ее к задаче раскраски графа. Граф G называется k -хроматическим, если его вершины раскрашены в k цветов так, что смежные вершины имеют разные цвета. Наименьшее число цветов, в которые может быть раскрашен граф, называется *хроматическим числом* и обозначается $\gamma(G) = k_{\min}$. Задача определения $\gamma(G)$ называется *задачей раскраски графа*.

Существуют различные методы решения данной задачи. Ход решения поясним на примере, в котором используется следующий эвристический алгоритм, приводящий к тупиковому решению, обычно близкому к оптимальному.

1) Пары независимых и зависимых выходов представляются в виде неориентированного графа: вершины — выходы устройства, а дуги связывают зависимые пары вершин.

2) Выбирается любая пара несмежных (т.е. независимых) вершин и окрашивается одинаковым цветом. Если таких вершин нет — конец, иначе — переход к шагу 3.

3) Выбранная пара вершин склеивается. Переход к шагу 2.

В новом полученном графе все вершины будут окрашены в различные цвета. Каждая его вершина отображает группу независимых выходов.

Пример 14.2. Пусть имеется $m = 7$ выходов некоторого комбинационного устройства, среди которых определены 4 пары зависимых выходов $\{1, 2\}, \{4, 5\}, \{5, 6\}, \{4, 6\}$ и две графически независимых группы выходов $G_1 = \{1, 2, 3\}$ и $G_2 = \{4, 5, 6, 7\}$. Требуется минимизировать число групп, содержащих независимые выходы.

Графически независимые группы составляют два подграфа — G_1 и G_2 . В каждом подграфе дуги соединяют зависимые пары вершин (рис. 14.9).

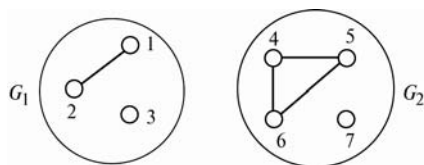


Рис. 14.9

Процесс склеивания вершин в подграфах G_1 и G_2 и затем во всем графе G приведен на рис. 14.10. Буквами обозначены цвета: к — красный, с — синий, ж — желтый.

В результате применения алгоритма раскраски получены $k = 3$ цвета и соответственно три группы: две группы независимых выходов $\{2, 4, 7\}$ и $\{1, 3, 6\}$ и группа, содержащая отдельный выход $\{5\}$. При этом дуги свидетельствуют о зависимости некоторых вершин, входящих в разные группы. Можно получить и другие тупиковые решения (возможно, даже более хорошие) в зависимости от того, какие вершины объединяются (склеиваются). Отметим, что цвет вершин не имеет значения, поэтому после каждого склеивания можно изменять цвет вершин.

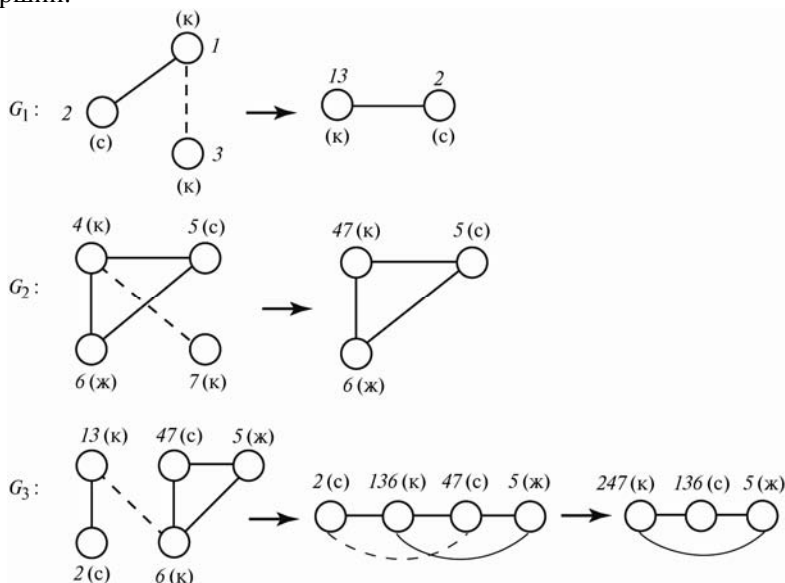


Рис. 14.10

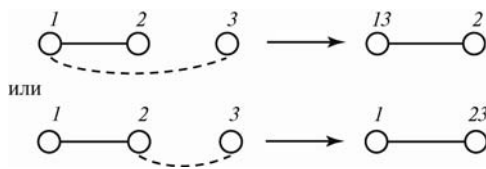


Рис. 14.11

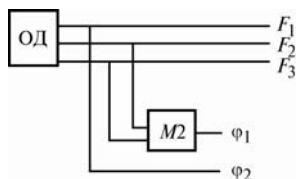


Рис. 14.12

Теперь вернемся к исходной задаче. Для нее рассматриваемый граф содержит три вершины — $1, 2, 3$ (соответствующие выходам F_1, F_2, F_3), из них две вершины — зависимые ($1, 2$), проводим между ними дугу, при этом возможны два варианта (рис. 14.11). Выберем второй вариант (группы $\{1\}$ и $\{2, 3\}$).

Независимые выходы в группе объединяются с помощью схем свертки $M2$ для последующего вывода на СС. Если же группа содержит один выход, то свертка не требуется и F_i сразу подается на СС (рис. 14.12).

На этом завершается решение первой задачи.

14.2.2. Решение 2-й задачи — синтез ДДУ

Сложность ДДУ зависит от выбранного множества групп независимых выходов и от рабочего набора значений входных переменных при функционировании объекта в штатном режиме. Число таких наборов обычно значительно меньше полного числа тестовых векторов.

Для выбранных групп $\{1\}$ и $\{2, 3\}$ имеем:

$$\begin{aligned} F'_1 &= \bar{\varphi}_1 = \overline{F_2 \oplus F_3}; \\ F'_2 &= \bar{\varphi}_2 = \bar{F}_1. \end{aligned} \quad (14.1)$$

Если бы в процессе штатного функционирования ОД на его вход подавались все возможные векторы x , то функция F'_i могла быть получена по таблице истинности (ТИ) в соответствии с (14.1) и синтези-

рована ДДУ. Однако если значения вектора x ограничены сокращенным рабочим набором, то ТИ сокращается и схема ДДУ упрощается.

Пусть рабочий набор содержит 6 из 16 возможных входных комбинаций (*ограниченная таблица истинности*) (табл. 14.8).

Таблица 14.8

x_1	x_2	x_3	x_4	F_1	F_2	F_3	φ_1	φ_2	F'_1	F'_2
0	0	1	1	1	0	1	1	1	0	0
0	1	0	1	1	1	1	0	1	1	0
0	1	1	0	0	1	1	0	0	1	1
1	0	0	1	1	0	1	1	1	0	0
1	1	0	1	0	0	1	1	0	0	1
1	1	1	0	0	1	0	1	0	0	1

При этом F'_i являются *неполностью определенными функциями*, т.е. определенными не на всех наборах.

Два входных набора называются *смежными* по переменной x_k , если они различаются только в одном разряде x_k . При этом значения выходных переменных не рассматриваются.

Переменная x_k называется *существенной для ДДУ*, если хотя бы на одном смежном наборе по переменной x_k изменяется значение F' :

$$x_k \rightarrow \bar{x}_k \Rightarrow F' \rightarrow \bar{F}'$$

(здесь стрелка « \rightarrow » означает изменение логического состояния переменной).

Несущественные для ДДУ переменные не рассматриваются, тем самым упрощается ее схема.

Алгоритм определения существенных переменных:

1) В ограниченной ТИ ищутся переменные x_k, x_l, \dots , для каждой из которых существует по крайней мере одна пара смежных наборов по соответствующей переменной, на которых переключается выход F'_i . Для этого просматривается очередной столбец x_k , находятся строки, где переменная x_k переключается. Далее определяется, являются ли эти строки смежными по x_k и переключается ли при этом выходная переменная. Найденные существенные переменные подчеркиваются. Если непомеченных переменных не осталось — то конец, иначе — переход к шагу 2.

2) Из ТИ вычеркивается один из столбцов (любой) с непомеченной (т.е. несущественной) переменной. Переход к шагу 1.

Пример 14.3. Проведем синтез ДДУ для ограниченной ТИ (табл. 14.8).

Все подчеркнутые (и выделенные полужирным шрифтом) переменные являются существенными.

Для выхода F'_1 составим табл. 14.9.

Таблица 14.9

	<u>x_1</u>	x_2	x_3	x_4	F'_1
1	0	0	1	1	0
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	1	0	1	0
6	1	1	1	0	0

Для x_1 имеем смежные наборы $\{2, 5\}$ и $\{3, 6\}$, на которых F'_1 переключается (меняет логическое значение). Значит, x_1 — существенная переменная. Переменная x_2 не является существенной, так как на смежных наборах $\{4, 5\}$ F'_1 не переключается. Переменные x_3 и x_4 также не являются существенными, так как для этих переменных нет смежных наборов.

Удалим, например, столбец x_4 . Получим сокращенную табл. 14.10.

Таблица 14.10

	<u>x_1</u>	<u>x_2</u>	x_3	F'_1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	1	0	0
6	1	1	1	0

В ней x_2 — существенная переменная (строки $\{1, 3\}$), x_3 — несущественная.

Удалим столбец x_3 , получим табл. 14.11.

Таблица 14.11

	<u>x_1</u>	<u>x_2</u>	F'_1
1	0	0	0
2	0	1	1
3	0	1	1
4	1	0	0
5	1	1	0
6	1	1	0

Остались только подчеркнутые переменные. В результате получаем тупиковое решение для выхода F'_1 :

$$F'_1(x_1, x_2) = \bar{x}_1 x_2. \quad (14.2)$$

Для выхода F'_2 имеем табл. 14.12.

	<u>x_1</u>	<u>x_2</u>	x_3	x_4	F'_2
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	1	0	1	1
6	1	1	1	0	1

Переменная x_1 существенная (см. наборы {2, 5} или {3, 6}), x_2 — тоже существенная (набор {4, 5}), а переменные x_3, x_4 несущественные, так как для них нет смежных наборов.

Удаляем x_4 , получаем табл. 14.13, в которой x_3 — существенная переменная (наборы {2, 3}), подчеркиваем ее.

Таблица 14.13

	<u>x_1</u>	<u>x_2</u>	<u>x_3</u>	F'_2
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	1	0	1
6	1	1	1	1

В результате получаем тупиковое решение для выхода F'_2 :

$$\begin{aligned} F'_2(x_1, x_2, x_3) &= \bar{x}_1 \underline{x_2 x_3} \vee \underline{x_1 x_2 x_3} = x_2 x_3 \vee x_1 x_2 \bar{x}_3 = \\ &= x_2 (x_3 \vee x_1 \bar{x}_3) = x_1 x_2 \vee x_2 x_3 \end{aligned} \quad (14.3)$$

(подчеркнуты склеиваемые термы).

Теперь на основании полученных функций (14.2) и (14.3) строится схема ДДУ. Этим завершается решение второй задачи. Напомним, что решение 1-й задачи дало группу независимых выходов $\{F_2, F_3\}$ и $\{F_1\}$. На рис. 14.13 приведена окончательная схема объекта с ССВК.

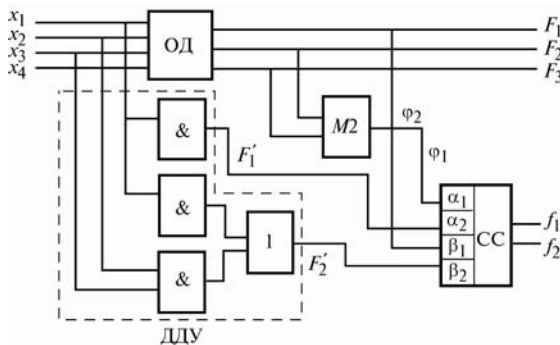


Рис. 14.13

Пусть теперь выполняется не функциональный, а тестовый контроль работоспособности ОД. Достоинством такого контроля является проявление неисправностей, которые не обнаруживались рабочими входными наборами.

Тогда ДДУ строится на основании выражений (14.1):

$$F'_1 = \overline{F_2 \oplus F_3} = \overline{x_1 x_4 \vee x_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 \oplus x_1 x_2 \vee x_3 \bar{x}_4} ;$$

$$F'_2 = \overline{F_1} = x_1 x_2 \vee x_1 x_3 \vee x_2 x_3 .$$

Естественно, ДДУ в этом случае будет сложнее.

15. ПОСТРОЕНИЕ ЛЕГКОТЕСТИРУЕМЫХ СХЕМ

15.1. Идея построения легкотестируемых схем

Используя методы повышения ремонтпригодности, можно существенно сократить необходимый объем тестов для проверки исправности объекта.

Произвольную комбинационную схему в базисе И-ИЛИ-НЕ всегда можно модифицировать таким образом, чтобы некоторый набор из трех тестовых векторов покрывал все возможные одиночные константные неисправности.

Однако это требует введения дополнительных логических элементов и дополнительных входных и выходных полюсов, используемых в режиме тестирования. Обычно получаемая избыточность достаточно высока, поэтому для использования такого подхода необходимо иметь веское основание.

Посмотрим как выполняется такая модификация, что и будет доказательством приведенного утверждения.

Сначала рассмотрим двухвходовую схему И с возможными одиночными неисправностями типа $\equiv 0$ и $\equiv 1$ на ее входах x и выходе F , (рис. 15.1).

Легко заметить, что эти неисправности можно покрыть следующими тремя входными векторами, приведенными на рисунке:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

При этом на первый вход подается последовательность 011, на второй — 101, а выходная последовательность для исправной схемы будет 001. Любая одиночная неисправность искажает ее и, следовательно, проявляется. Но эти векторы и последовательности не единственные. Данный элемент можно проверить любыми двумя входными последовательностями, выбранными из следующего множества:

$$S_1 = \{011\}, \{101\}, \{110\}$$

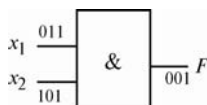


Рис. 15.1

(две единицы и ноль). Первая последовательность из двух выбранных подается на один из входов схемы И, вторая — на другой вход.

Для двухвходовой схемы ИЛИ выбираются любые две последовательности из множества

$$S_2 = \{001, 010, 100\}$$

(два нуля и единица).

Возможные выходные последовательности элемента И:

при $x_1 = 011, x_2 = 101$ — $F = 001$,

при $x_1 = 011, x_2 = 110$ — $F = 010$,

при $x_1 = 101, x_2 = 110$ — $F = 100$.

Аналогично, возможные выходные последовательности элемента ИЛИ:

при $x_1 = 001, x_2 = 010$ — $F = 011$,

при $x_1 = 001, x_2 = 100$ — $F = 101$,

при $x_1 = 010, x_2 = 100$ — $F = 110$.

Отсюда видно, что для схемы И $F \in S_2$, а для схемы ИЛИ $F \in S_1$. А это означает, что если на выходе схемы И стоит схема ИЛИ (или наоборот), то выходная последовательность первого каскада является тестом, подаваемым на один из входов второго каскада. При этом на второй вход второго каскада следует подать последовательность из множества S_1 или из S_2 в зависимости от типа соединения каскадов.

Если же после первого каскада стоит однотипный элемент с двумя входами, то между 1-м и 2-м каскадами следует ввести дополнительный элемент двойственного типа (т.е. И после ИЛИ или ИЛИ после И).

Если имеется инвертор, то он преобразует последовательность из множества S_1 в последовательность из множества S_2 или наоборот, при этом также может потребоваться введение дополнительного каскада (И или ИЛИ).

Например, на рис. 15.2 приведена исходная схема, а на рис. 15.3 изображена модифицированная схема (добавлена схема ИЛИ и дополнительный вход с) с тестовыми последовательностями, покрывающими все неисправности.

Пример схемы с инвертором приведен на рис. 15.4

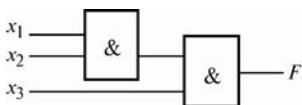


Рис. 15.2

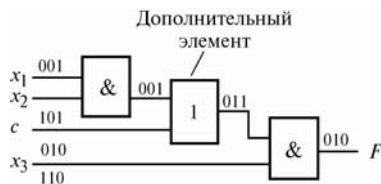


Рис. 15.3

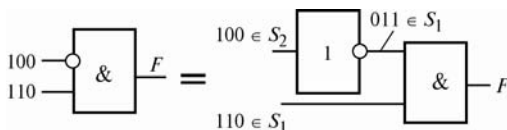


Рис. 15.4

15.2. Порядок построения легкотестируемых схем

Для получения легкотестируемой схемы применяется следующий алгоритм:

- 1) Комбинационная схема преобразуется в схему с базисом И, ИЛИ, НЕ.
- 2) Элементы И и ИЛИ преобразуются в двухвходовые элементы И и ИЛИ.
- 3) Выбираются тестовые наборы из множества S_1 или S_2 в зависимости от типа входных элементов.
- 4) Если выходная последовательность элемента не принадлежит требуемому тестовому множеству S_1 или S_2 (в зависимости от типа следующего элемента), то вводится дополнительная схема И или ИЛИ с дополнительным входом, на который подается последовательность из S_1 или из S_2 в зависимости от того, что нужно получить на выходе.
- 5) Если нужно не только проверить объект, но и локализовать неисправность на входе или выходе какого-либо элемента, то добавляются дополнительные выходы после каждого элемента (или в отдельных точках в зависимости от допустимой глубины диагностирования).

Пример 15.1. На рис. 15.5 изображена исходная схема ОД, выполняющая преобразование

$$F = x_1x_2x_3 \vee \bar{x}_1\bar{x}_2\bar{x}_3.$$

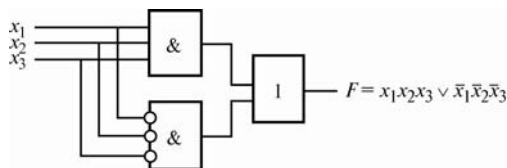


Рис. 15.5

Требуется построить легкотестируемую схему с тремя тестовыми наборами.

1) Исходная схема (рис. 15.5) уже содержит только элементы И, ИЛИ, НЕ.

2) Все элементы преобразуются в двухвходовые (рис. 15.6) (элементы пронумерованы).

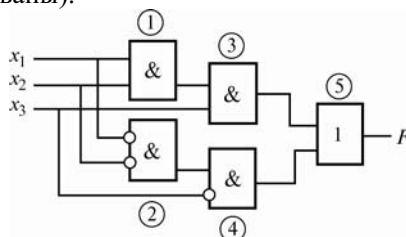


Рис. 15.6

3) Выполняются остальные этапы модификации. В результате получается схема, приведенная на рис. 15.7. Символами c_i обозначены дополнительные входы. Введены также дополнительные выходы.

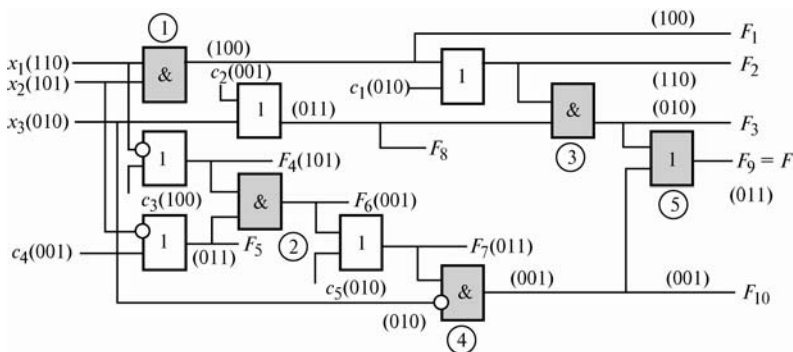


Рис. 15.7

Схема выполняет следующую функцию:

$$F = F_9 = (x_1 x_2 \vee c_1)(x_3 c_2) \vee [(\bar{x}_1 \vee c_3)(\bar{x}_2 \vee c_4) \vee c_5] \bar{x}_3 = \\ = x_1 x_2 x_3 \vee c_1 x_3 \vee x_1 x_2 c_2 \vee c_1 c_2 \vee \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 \bar{x}_3 c_4 \vee c_3 c_4 \bar{x}_3 \vee c_5 \bar{x}_3.$$

В рабочем режиме на все входы c_i подаются нули. При этом

$$F = x_1 x_2 x_3 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3.$$

В режиме тестирования на входы c_i подаются сигналы из набора S_2 (так как в данной схеме они поступают только на входы элементов ИЛИ) для того, чтобы на выходе получились сигналы из S_1 . В результате имеем 3 теста:

	x_1	x_2	x_3	c_1	c_2	c_3	c_4	c_5	F
π_1	1	1	0	0	0	1	0	0	0
π_2	1	0	1	1	0	0	0	1	1
π_3	0	1	0	0	1	0	1	0	1

Однако введение дополнительных элементов с входами c_i приводит к увеличению числа возможных неисправностей. Некоторые из них могут не проявиться на выходе F вследствие маскировки параллельными путями сходящихся разветвлений. Например, неисправность $c_1^* \equiv 1$ не изменяет логического состояния F ни на одном из трех тестов. Это видно из формулы для F , в которой c_1 содержится в двух термах: $c_1 x_3$ и $c_1 c_2$. При значениях c_2 , указанных в таблице, изменение c_1 не изменяет значения F . Для наблюдения этой неисправности вводится дополнительный выход F_2 .

Введение дополнительных выходов позволяет также локализовать неисправность.

Примечание. В рассмотренном примере построение легкотестируемой схемы привело к возрастанию количества модулей с 3 до 10 и числа входов с 3 до 8. Это плата за сокращение длины тестовой последовательности. Поэтому в сложных устройствах такую избыточность целесообразно использовать лишь в отдельных подсхемах для повышения их ремонтпригодности, особенно если неисправности не наблюдаются на выходе ОД, т.е. маскируются другими элементами.

КОНТРОЛЬНЫЕ ВОПРОСЫ

Глава 1

1. В чем различие понятий «Повреждение» и «Дефект»?
2. В чем различие показателей «Коэффициент готовности» и «Коэффициент оперативной готовности»?

Глава 2

1. Несовместимые события являются: а) всегда независимыми, б) всегда зависимыми, в) могут быть как независимыми, так и зависимыми?
2. Какой случайный процесс называется марковским?
3. Всегда ли однородный марковский процесс является стационарным?
4. Всегда ли стационарный марковский процесс является эргодическим?
5. Как выражается структурная функция системы в логической и арифметизированной формах?
6. Как определяются показатели надежности в расчетах методом структурных функций?
7. Назовите формы связности элементов системы, используемые в качестве показателей надежности.
8. Какое условие является необходимым и достаточным для связности вершин графовой модели системы?
9. Как выполняется генерация случайных величин с заданной функцией распределения методом обратных функций?

Глава 3

1. Как связана (в общем виде) вероятность безотказной работы объекта $P(t)$ с плотностью распределения ВБР $f_{\theta}(t)$ и интенсивностью отказов $\lambda(t)$?
2. Как вычисляются статистические оценки показателей надежности $P(t)$, $P(t+\Delta t)$, $f_{\theta}(t)$ и $\lambda(t)$?
3. Какие способы используются для статистической оценки средней наработки до отказа объекта?

Глава 4

1. Чему равна интенсивность восстановления μ в модели точечного потока отказов / восстановлений?

2. В каком случае параметр точечного потока ω_0 совпадает с интенсивностью отказов λ элемента?
3. Как рассчитать статистическую оценку параметра ω_0 точечного потока?
4. Какое свойство эргодического марковского процесса используется для статистической оценки значения коэффициента готовности по одной реализации?
5. Насколько быстро достигается установившийся режим коэффициента готовности $k_T(t)$?

Глава 5

1. Дайте определение понятия «Схема надежности».
2. Чему равна интенсивность отказов при последовательном соединении элементов в схеме надежности: а) $\lambda_c = \prod_j \lambda_j$ б) $\lambda_c = \sum_j \lambda_j$?
3. В каких случаях функция распределения ВБР элементов совпадает с функцией распределения ВБР системы?
4. Какой вид резервирования обеспечивает большую надежность системы: а) общее резервирование, б) раздельное резервирование?
5. Какой вид резервирования обеспечивает большую надежность системы: а) нагруженное резервирование с постоянно включенным резервом, б) нагруженное резервирование замещением (без учета надежности коммутаторов)?

Глава 6

1. Какие показатели надежности могут быть рассчитаны методом структурных функций?
2. При каких условиях нельзя применять метод структурных функций?
3. В каких случаях следует использовать ортогонализацию логической структурной функции вместо ее арифметизации?
4. Дайте определение понятий «минимальный путь» и «минимальный разрез».
5. В каких случаях более предпочтительно применение метода минимальных разрезов?
6. Всегда ли схема троирования обеспечивает более высокую надежность по различным показателям по сравнению с нерезервированной системой?

Глава 7

1. Каковы преимущества контроля вычислительных операций по вычитам по сравнению с контролем путем повторения вычислений?
2. Всегда ли при $A = a^m$ справедливо сравнение $A \equiv a \pmod{m}$?
3. О чем свидетельствует сравнимость $A \equiv 1 \pmod{m}$, если $A = a^{m-1}$?
4. В каких случаях цифровые вычеты по модулю m совпадают с числовыми?
5. Почему при контроле арифметических операций сложения и умножения по цифровым вычетам двоичный код числа A обычно переводится в код с основанием $h \in \{4, 8, 16, 32, \dots 2^k\}$?

Глава 8

1. Покрываются ли неисправные состояния при контроле работоспособности системы: а) всегда покрываются, б) никогда не покрываются, в) могут как покрываться, так и не покрываться?
2. В каких моделях события a_i являются несовместимыми: а) в модели кратных зависимых неисправностей; б) в модели кратных независимых неисправностей; в) в модели одиночных неисправностей; г) во всех моделях; д) ни в какой модели?
3. В чем различие функциональных и логических неисправностей?
4. Что такое «константная неисправность»?

Глава 9

1. В чем состоит метод раскрутки при диагностировании ВС?
2. В чем различие распределенного диагностического ядра от центрального (сосредоточенного)?
3. В чем особенности и какие показатели характеризуют ПМЧ-модель диагностирования?
4. При каком числе активных модулей возможно применение ПМЧ-модели?
5. Как выполняется раскрутка при использовании ПМЧ-модели?
6. Как выполняется определение показателя τ при взаимной проверке состояний модулей?
7. Как осуществляется расшифровка синдрома в ПМЧ-модели многопроцессорной системы?
8. Назовите этапы раскрутки в однопроцессорной ВС.

Глава 10

1. Назовите основные элементы функционально полной диагностической микропрограммы и этапы ее проектирования.

2. Укажите порядок выполнения команд в микропрограмме.
3. В чем состоит недостаток метода активизации одномерного пути при синтезе тестов?
4. Какие этапы содержит синтез тестов методом активизации одномерного пути?
5. Что такое «булева производная»?
6. В чем достоинство метода булевой производной? Перечислите этапы нахождения тестов этим методом.
7. Укажите пути упрощения расчетов при синтезе тестов методом булевой производной.
8. Как строится таблица неисправностей?

Глава 11

1. Что такое «сигнатура»?
2. Какие достоинства и недостатки имеет сигнатурный анализ по отношению к методу непрерывного сравнения реакций рабочего и эталонного объекта?
3. Какие компактеры используются при сигнатурном анализе?
4. Как выбирается полином для сигнатурного анализа по остаткам?
5. В чем преимущество использования сигнатурного анализа по остаткам по сравнению с другими методами анализа?
6. Как строятся компактеры эквивалентных анализаторов на сдвиговых регистрах?
7. В чем состоит преимущество компактеров на сдвиговых регистрах с внешними обратными связями по сравнению с компактерами с внутренними связями?

Глава 12

1. Каковы задачи построения алгоритмов диагностирования?
2. Чем различаются условные алгоритмы от безусловных?
3. Укажите нижнюю границу числа проверок при использовании безусловного алгоритма с безусловной остановкой?
4. Как выполняются проверки при использовании безусловных алгоритмов с условной остановкой по методу обратного просмотра?
5. Для чего служит диагностический словарь?
6. Как выполняется построение диагностического словаря?
7. Как изображается дерево проверок условного алгоритма?
8. Для чего служит дерево проверок?
9. Каковы особенности построения условного алгоритма с использованием информационного критерия?

Глава 13

1. Какие способы и средства используются для повышения тестопригодности ВС?
2. В чем состоит идея метода сканирования?
3. Какими достоинствами обладает метод граничного сканирования?
4. Для чего служит и как используется *JTAG*-интерфейс?
5. Как строится граф управляемости / наблюдаемости ВС?
6. Что такое максимально сильносвязный (МСС) подграф графа управляемости / наблюдаемости?
7. Для чего и как выполняется конденсация графа управляемости / наблюдаемости?

Глава 14

1. Какие схемы называются самопроверяемыми?
2. Почему в схемах встроенного контроля используются многovyходные компараторы?
3. Какими достоинствами и недостатками обладают схемы самопроверяемого дублирования?
4. Каким требованиям должны удовлетворять независимые выходы цифровых устройств?
5. Как строятся граф и матрица соединений?
6. Для чего строится обобщенный граф соединений?
7. Как определяются пары независимых выходов?
8. Как определяются полные группы независимых выходов?
9. Для чего используется дополнительное дискретное устройство (ДДУ)?
10. Как выполняется минимизация ДДУ?

Глава 15

1. Перечислите этапы построения легкотестируемых комбинационных устройств контроля исправности.
2. Как обеспечить локализацию всех одиночных неисправностей в таких устройствах?

СПИСОК ЛИТЕРАТУРЫ

1. **Половко А.М., Гуров С.В.** Основы теории надежности. — 2-е изд. — СПб.: БХВ-Петербург, 2006. — 704 с.
2. **Половко А.М., Гуров С.В.** Основы теории надежности. Практикум. — СПб.: БХВ-Петербург, 2006. — 560 с.
3. **Афонин В.А.** Обеспечение надежности СВТ: Сборник задач: учеб. пособ. — 2-е изд. — М.: Издательство МЭИ, 2004. — 84 с.
4. **Игуду К.А.** Надежность, контроль и диагностика вычислительных машин и систем. — М.: Высшая школа, 1989. — 216 с.
5. **Ушаков И.А.** Вероятностные модели надежности информационно-вычислительных систем. — М.: Радио и связь, 1991. — 132 с.
6. **Афонин В.А., Ладыгин И.И.** Построение отказоустойчивых вычислительных систем. — М.: МЭИ, 1987. — 68 с.
7. **Афонин В.А., Ладыгин И.И.** Руководство по расчету характеристик надежности вычислительных систем в дипломном и курсовом проектировании. — М.: Издательство МЭИ, 1994. — 39 с.
8. **Щербаков Н.С.** Надежность и достоверность работы цифровых устройств и ЭВМ. — М.: Издательство МЭИ, 1994. — 71 с.
9. **Щербаков Н.С.** Функциональное и тестовое диагностирование цифровых устройств и ЭВМ. — М.: Издательство МЭИ, 1994. — 124 с.
10. **Каган Б.М., Мкртумян И.Б.** Основы эксплуатации ЭВМ. — Изд. 2-е. — М.: Энергоатомиздат, 1988. — 432 с.
11. **Хетагуров Я.А.** Детерминированная теория надежности экземпляра вычислительной машины, системы. Вопросы проектирования, производства, эксплуатации. — М.: МИФИ, 1997. — 132 с.
12. **Петров Ю.П.** Обеспечение достоверности и надежности компьютерных расчетов. — СПб.: БХВ-Петербург, 2008.
13. **Основы технической диагностики: в 2 книгах. Кн. 1. Модели объектов, методы и алгоритмы диагноза** / В.В. Карибский, П.П. Пархоменко, Е.С. Согомонян, В.Ф. Халчев; под ред. П.П. Пархоменко. — М.: Энергия, 1976. — 404 с.
14. **Пархоменко П.П., Согомонян Е.С.** Основы технической диагностики (Оптимизация алгоритмов диагностирования, аппаратные средства) / под ред. П.П. Пархоменко. — М.: Энергия, 1981. — 320 с.
15. **Согомонян Е.С., Слабаков Е.В.** Самопроверяемые устройства и отказоустойчивые системы. — М.: Радио и связь, 1989. — 208 с.
16. **Андерсон Д.А.** Дискретная математика и комбинаторика: пер. с англ. — М.: Издательский дом «Вильямс», 2004. — 961 с.
17. **Байхельт Ф., Франкен П.** Надежность и техническое обслуживание. Математический подход: перевод с нем. — М.: Радио и связь, 1988. — 392 с.
18. **Райншке Н., Ушаков И.А.** Оценка надежности систем с использованием графов. — М.: Радио и связь, 1988. — 209 с.

19. **Вычислительные** сети (адаптивность, помехоустойчивость, надежность) / С.В. Самойленко, А.А. Давыдов, В.В. Золотарев, И.М. Третьякова. — М.: Наука, 1984. — 277 с.
20. **Журавлев Ю.П., Котелюк Л.А., Циклинский Н.И.** Надежность и контроль ЭВМ. — М.: Сов. Радио, 1978. — 416 с.
21. **Построение** сетей ЭВМ / М. Като, Д. Иимура, М. Токоро, Е. Тома. — М.: Мир, 1988. — 307 с.
22. **Киносита К., Асада К., Карацу О.** Логическое проектирование СБИС. — М.: Мир, 1988. — 309 с.

Учебное издание

Афонин Владимир Александрович

ОСНОВЫ ТЕОРИИ НАДЕЖНОСТИ

Учебное пособие

по курсу «Основы теории надежности»
для студентов, обучающихся по направлению
«Информатика и вычислительная техника»

Редактор *В.С. Егоров*

Компьютерная верстка *М.А. Маркина*

Подготовка электронного издания *Д.О. Лазарев*

АО «Издательский дом МЭИ»,

111250, Москва, Красноказарменная ул., 14А

тел/факс: (495) 640-83-27, адрес в Интернете: <http://www.idmei.ru>,

электронная почта: info@idmei.ru