Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

КРАТКИЙ КУРС ТЕХНИЧЕСКОЙ ДИАГНОСТИКИ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ

Методическое пособие по дисциплине
«Контроль и диагностика средств вычислительной техники»
для студентов специальности I-40 02 01
«Вычислительные машины, системы и сети»
вечерней и заочной форм обучения

Минск 2005

УДК 004 (075.8)
ББК 32.973 я 73
К 78

Авторы-составители:
М.М. Лукашевич, М.М. Татур

Краткий курс технической диагностики вычислительных устройств: Метод. пособие по дисциплине «Контроль и диагностика средств вычислительной техники» для студ. спец. I-40 02 01 «Вычислительные машины, системы и сети» вечерней и заочной форм обуч. / Сост. М.М. Лукашевич, М.М. Татур. – Мн.: БГУИР, 2005. – 36 с.: ил.
ISBN 985-444-844-4

К 78

В методическом пособии рассмотрены основные вопросы дисциплины «Контроль и диагностика средств вычислительной техники», читаемой студентам специальности I-40 02 01 «Вычислительные машины, системы и сети».

УДК 004 (075.8)
ББК 32.973 я 73

ISBN 985-444-844-4

1. ОБЩИЕ ПОНЯТИЯ ТЕХНИЧЕСКОЙ ДИАГНОСТИКИ

К таким понятиям относятся:

1. Техническое диагностирование, виды технического состояния объекта: исправное, работоспособное, правильного функционирования.
2. Дефект, сбой, отказ, неисправность, ошибка. Причинно-следственные связи.
3. Виды технического диагностирования: функциональное и тестовое диагностирование.
4. Средства диагностирования.

Техническая диагностика – это отрасль научно-технических знаний, основу которой составляют теория, методы и средства поиска и обнаружения дефектов объектов вычислительной техники (объекта диагностирования). Любой технический объект проходит следующие основные стадии «жизни»: проектирование, изготовление, эксплуатацию и ремонт. Основные задачи технической диагностики: предотвращение производственного брака при изготовлении объектов; повышение надежности и достоверности правильного функционирования при их эксплуатации; обеспечение быстрого и качественного ремонта.

Повышение надежности проявляется как улучшение показателей коэффициента готовности, коэффициента технического использования, времени восстановления работоспособного состояния, наработки до отказа. Предотвращение производственного брака достигается правильной организацией диагностирования на операциях входного контроля комплектующих изделий и материалов, контроля технологических процессов изготовления объектов, включая выходной контроль последних.

Для обеспечения качества любой технической системы (объекта) возникает необходимость на всех стадиях жизненного цикла определять ее техническое состояние. Процесс определения технического состояния (диагноза) объекта диагностирования называется ***техническим диагностированием***.

Процесс проектирования объекта сопровождается нормативно-технической документацией, в которой определен перечень требований к параметрам и характеристикам этого объекта. Если объект удовлетворяет всем требованиям, то он признается *исправным*, в противном случае – *неисправным*. Например, в нормативных требованиях указаны допуски температурных режимов, целостность корпуса объекта и т.п. Тогда выход за допуск, трещина на корпусе являются причинами, по которым техническое состояние объекта оценивается как неисправное.

С точки зрения обработки информации в вычислительных машинах особый интерес приобретают другие виды технических состояний, а именно, работоспособное/неработоспособное, правильно функционирующее/неправильно функционирующее.

Работоспособным называется техническое состояние, при котором объект может выполнять все предписанные функции с сохранением значений параметров в заданных пределах. Например, комбинационная схема способна реализовывать таблицу истинности на всех, без исключения, наборах, последовательностная схема способна реализовывать всю, без исключения, таблицу переходов и выходов.

На этапе эксплуатации важное значение приобретает техническое состояние *правильного функционирования*, когда объект правильно, без ошибок выполняет свою функцию (предписанный алгоритм) в текущий момент времени. Например, при вычислении арифметического сложения 2+3 получили 5, что соответствует действительности.

Классификатор технических состояний приведен на рис. 1.1. Переход из одного технического состояния в его антипод с приставкой НЕ происходит по причинам, связанным с воздействием внешней среды (электромагнитное поле, температура, неправильные действия оператора, механические воздействия) либо вследствие деградации объекта.

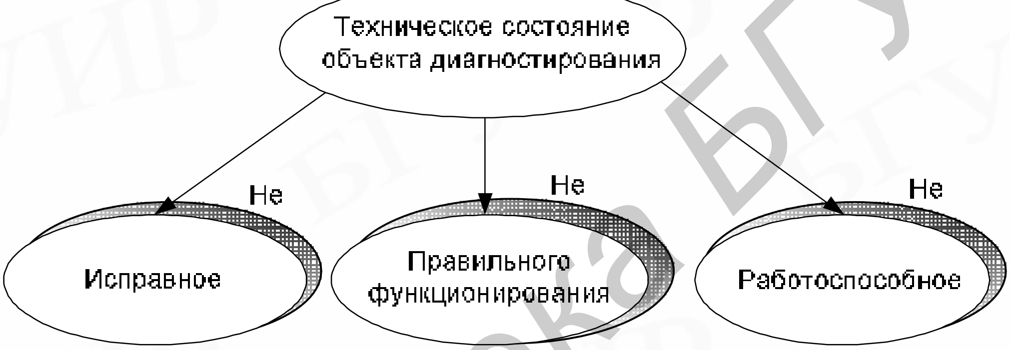


Рис. 1.1. Виды технического состояния

Первичным в этой цепи событий является ***физический дефект***, который может приводить к отказам и сбоям аппаратуры (объекта).

Отказ – это переход объекта диагностирования из работоспособного состояния в неработоспособное. Отказы могут быть полные либо частичные, постепенные либо катастрофические (внезапные), восстанавливаемые либо не восстанавливаемые.

Сбой – это самоустраняющийся отказ, временная утрата работоспособности объектом. Как сбой, так и отказ могут приводить к *ошибкам* в производимых вычислениях, реализуемых алгоритмах и т.п. Причинно-следственные связи подобных цепей событий отражены схемой на рис. 1.2.

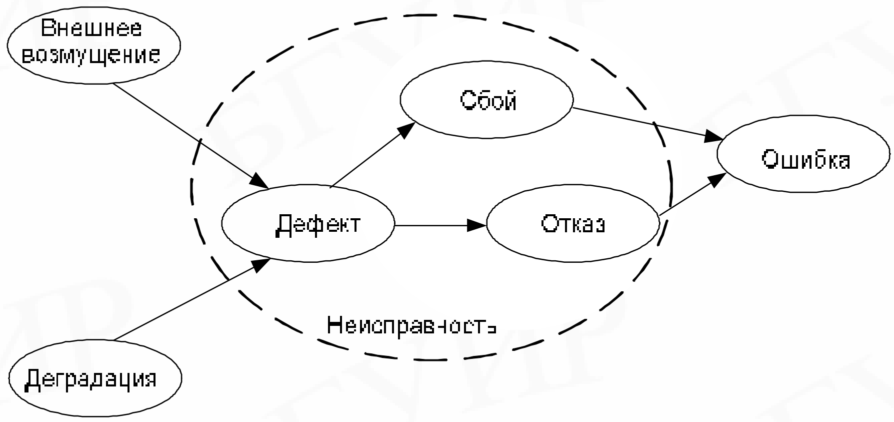


Рис. 1.2. Причинно-следственные связи

Особое место в технической диагностике занимает понятие неисправности.

Логическая неисправность (неисправность) – это модель, характеризующая проявление дефекта как изменение функции хотя бы одного элемента устройства (не смешивать с понятием неисправного технического состояния объекта, рис. 1.1).

В зависимости от вида технического состояния, получаемого в ходе диагностирования, различают ***функциональное и тестовое диагностирование***.

При *функциональном диагностировании* объект непосредственно используется по прямому назначению, на вход подаются «рабочие воздействия». Диагнозом является заключение о техническом состоянии: объект правильно (неправильно) функционирует в данный момент времени. Схема функционального дигностирования представлена на рис. 1.3.

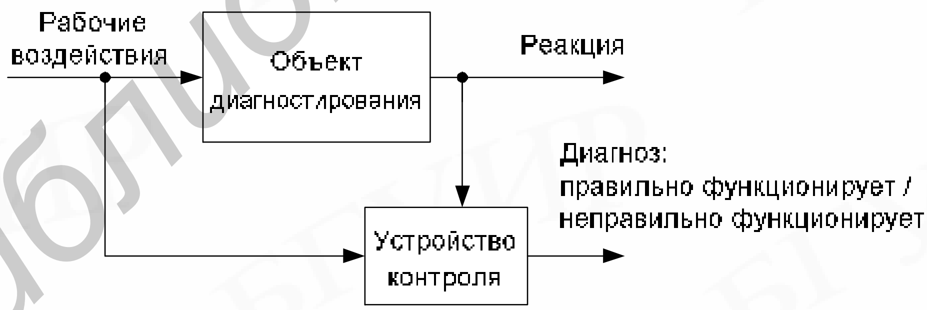


Рис. 1.3. Схема функционального диагностирования

Синонимами понятия «функциональное диагностирование» является «оперативный, аппаратный контроль» (“on line testing”). Обычно функциональное диагностирование применяется в системах управления, где риск принятия неверного решения в реальном времени должен сводиться к минимуму (управление пилотированием, наведение ракет, управление ядерным реактором).

В случае *тестового диагностирования* на вход объекта подаются специальные воздействия (тестовые наборы) и по реакции на них ставится диагноз: работоспособен или неработоспособен объект. При этом объект не используется по функциональному назначению. Синонимом тестового дигностирования является «тестирование» (“off line testing”). Общая схема тестового диагностирования приведена на рис. 1.4.

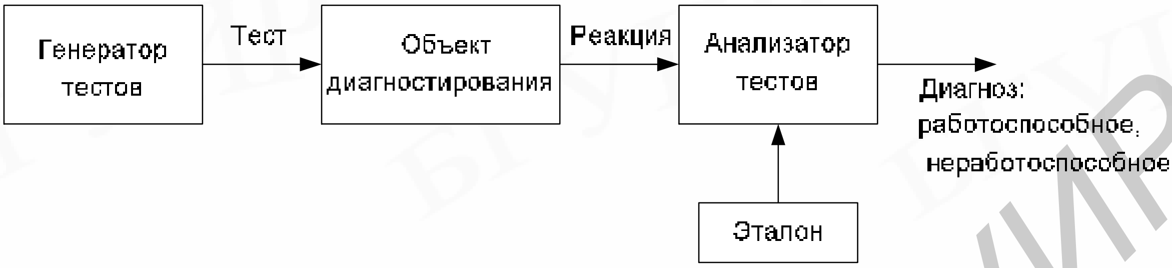


Рис. 1.4. Схема тестового диагностирования

Из рис. 1.3 и 1.4 видно, что наряду с объектом диагностирования присутствуют дополнительные аппаратные и программные ***средства диагностирования***, цель которых – автоматизировать процесс получения диагноза. Предметом технической диагностики является исследование как самого объекта диагностирования, так и методов построения средств диагностирования.

Конструктивно средства диагностирования могут быть как *внешними* по отношению к объекту диагностирования, так и *встроенными* в объект диагностирования, т.е. представлять единый вычислительный модуль, устройство.

Общая схема организации процедуры диагностирования выглядит следующим образом (рис. 1.5).

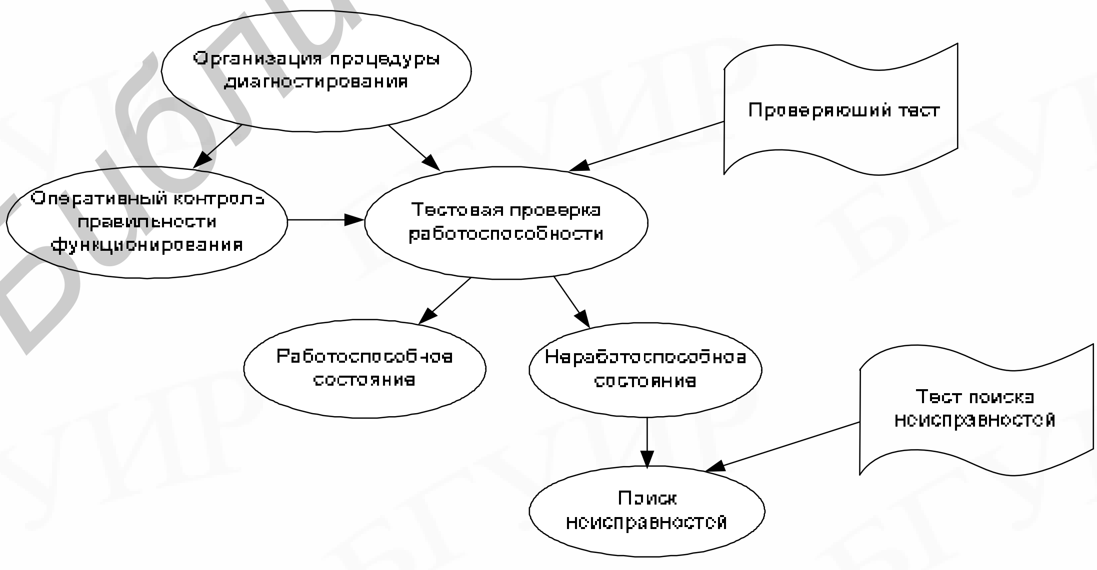


Рис. 1.5. Общая схема организации процедуры диагностирования

2. ДЕТЕРМИНИРОВАННЫЙ СИНТЕЗ ТЕСТОВ

В данном разделе рассматриваются следующие вопросы:

1. Моделирование структур и неисправностей.
2. Типовые задачи тестового диагностирования.
3. Таблица функций неисправностей (ТФН).
4. Метод конкурентного моделирования.
5. Метод булевой производной.
6. Построение проверяющих тестов методом активизации путей.

Задачи детерминированного поиска, обнаружения и идентификации неисправностей могут быть формально решены только в том случае, когда определено, каким образом неисправность изменяет функцию объекта. Это – принципиальная позиция, с которой начинается любой формальный анализ структур. Процесс анализа называют ***моделированием***, которое может быть выполнено аналитически либо имитационно (на основе анализа структуры либо рекуррентных описаний структуры).

Цель моделирования состоит в том, чтобы проследить без физической реализации, как поведет себя исследуемая схема при тех или иных изменениях в ее структуре. Моделирование можно осуществлять с различной степенью детализации структуры объекта, а главное, происходящих в ней процессов. Поэтому различают два способа моделирования структур: *компилятивный и событийный* (рис. 2.1).

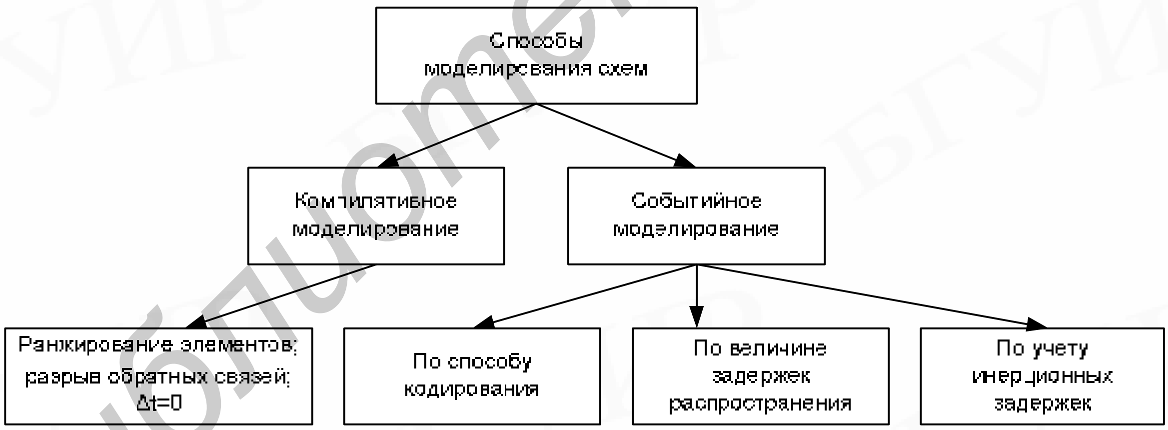


Рис. 2.1. Классификация способов моделирования структур

Наиболее простым является *компилятивный способ*. Он заключается в том, что каждому элементу структуры ставится в соответствие код команды с операцией, соответствующей моделируемому элементу. Машина последовательно выполняет команды и тем самым шаг за шагом продвигается к вычислению конечной функции моделируемой схемы. При этом необходимо следить за тем, чтобы соблюдалась корректность в очередности выполнения команд, поскольку результаты выполнения одних команд являются входными данными для других команд. Это достигается процедурой *ранжирования* схемы, т.е. изображения и описания элементов в возрастающей последовательности выполняемых операций, чтобы не было противоречий. При компилятивном моделировании не учитывают задержки прохождения сигнала через вентиль, т.е. Δt принимается равным 0. Такой способ применяется только при моделировании комбинационных схем и синхронных последовательностных схем.

В противоположность компилятивному способу при *событийном моделировании* логическая схема представляется достаточно подробно. Изменение значения сигнала называют событием – отсюда и название данного способа моделирования. При событийном моделировании основной упор делается на отслеживание факта изменения сигнала в анализируемых точках. При этом способе моделирования могут оцениваться реальные временные характеристики исследуемого объекта (время распространения сигнала по цепям) и, как следствие, могут выявляться логические состязания. Учет задержек при моделировании дает возможность обрабатывать последовательностные асинхронные схемы.

Моделирование неисправностей в логических схемах – это исследование влияния неисправностей на объект диагностирования при воздействии на него рабочих сигналов либо тестов. Введение понятия *модели неисправности* позволяет абстрагироваться от конкретных физических процессов (дефектов), произошедших в объекте либо в его отдельных элементах, и сосредоточиться на том, как эти процессы (дефекты) изменяют реальную функцию устройства по сравнению с эталонной.

Согласно базовым понятиям теории автоматов логические схемы можно разделить на комбинационные и последовательностные. Функция комбинационной схемы в любой момент времени определяется только логическим состоянием ее входов, и поэтому в отношении неисправностей комбинационные схемы сравнительно легко поддаются анализу. Функция последовательностной схемы определяется как состоянием входов, так и внутренними состояниями, поэтому формально описать поведение такой схемы в условиях различных неисправностей чрезвычайно сложно. В учебных целях ограничимся рассмотрением комбинационных схем.

В ходе логического моделирования неисправностей в схему вводят (симулируют) неисправность и анализируют влияние данной неисправности на функцию устройства. В зависимости от иерархического уровня представления структуры неисправности можно разделить на *логические* (на уровне логических элементов) и *функциональные* (на уровне функциональных узлов и устройств). В логических неисправностях выделяют следующие классы (модели): инверсные, константные, мостиковые, перепутывания. В свою очередь функциональные неисправности подразделяют на неисправности узлов, ОЗУ, ПЗУ, микропроцессоров, мониторов и т.п. Классификатор моделей приведен на рис. 2.2.

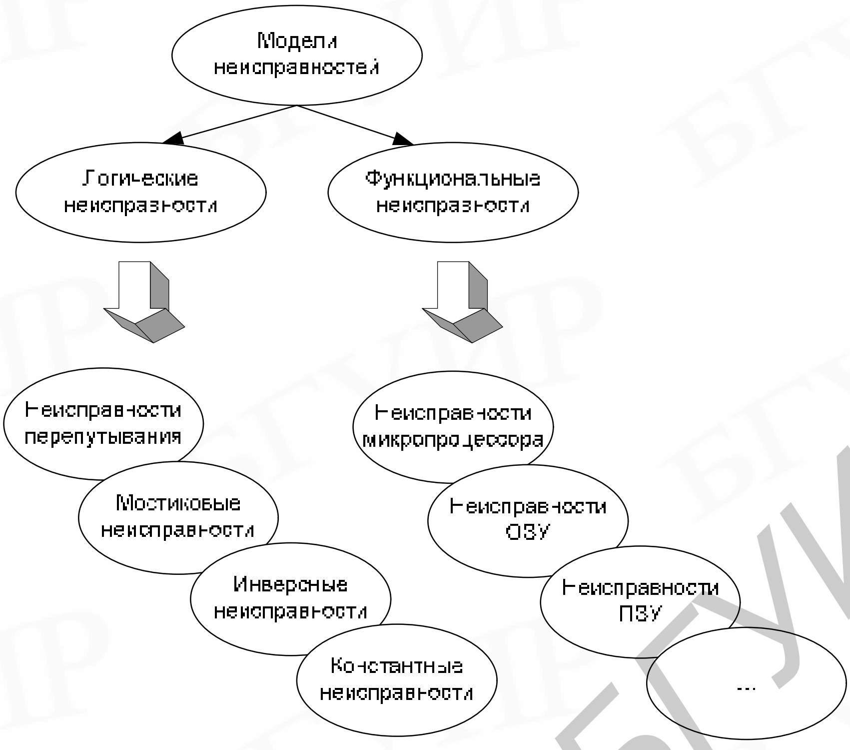


Рис. 2.2. Классификатор моделей неисправностей

Принципиальное различие логических и функциональных неисправностей состоит в том, что для моделирования логических неисправностей разработан и используется формальный аппарат, поэтому на логическом уровне широко применяют аналитические методы моделирования. На логическом уровне имеется возможность количественно оценить результаты моделирования, посчитать число выявляемых неисправностей. Для неисправностей функционального уровня моделирование осуществляется эвристическими процедурами и алгоритмами и, как следствие, применяется имитационное моделирование. В большинстве случаев для функциональных неисправностей и соответствующих тестов проблематично получить количественные оценки эффективности тестирования.

Основное предположение для всех моделей *логических неисправностей* состоит в том, что можно точно указать место такой неисправности в схеме и при ее возникновении схема будет оставаться логической и выполнять некоторую функцию. Наибольшее распространение нашли модели константных, инверсных, мостиковых, неисправностей типа «перепутывание». Логические неисправности моделируются в точках, расположенных на линиях связи отдельных логических элементов. Эти *контрольные точки* постоянно будут в поле нашего внимания при решении задач анализа на логическом уровне.

Наибольшее распространение при решении формальных задач анализа получила модель *константных неисправностей*. Суть данной модели заключается в фиксации логического 0 либо 1 в линиях распространения сигнала. В учебных целях обычно моделируются *одиночные* константные неисправности.

Для каждой схемы можно моделировать как *одиночные*, так и *кратные* неисправности. Одиночная неисправность – это неисправность, задаваемая в одной контрольной точке. Кратная неисправность – сочетание одиночных неисправностей в различных контрольных точках.

Особое внимание необходимо обратить на моделирование неисправностей в *точках разветвления*. Напомним, что константная неисправность – это всего лишь модель, т.е. функциональная реакция схемы на какой-то дефект, и не следует ее отождествлять с электрическим сигналом в проводниках. Поэтому в точках разветвления необходимо моделировать неисправности всех возможных линий (*путей*), т.е. неисправности в точках разветвления моделируются независимо друг от друга.

Очевидной особенностью одиночных неисправностей является их относительно небольшое число, что позволяет осуществить перебор при решении задач анализа. Выполнение полного перебора всех возможных сочетаний кратных неисправностей даже для несложной схемы потребует неоправданных вычислительных затрат.

Класс константных неисправностей может быть дополнен инверсными неисправностями с целью повышения полноты моделируемых «неисправных» ситуаций. Модель *инверсной* неисправности описывается как появление фиктивного инвертора в заданной контрольной точке. Число возможных инверсных неисправностей в два раза меньше по сравнению с одиночными константными неисправностями.

Наряду с константными и инверсными неисправностями при моделировании иногда используют модель *мостиковых* неисправностей (*короткое замыкание, перемычка)*. Модель мостиковой неисправности описывается как замыкание между отдельными сигнальными линиями. В большинстве случаев неисправности подобного типа воспринимаются схемой как объединение по И либо по ИЛИ нескольких сигнальных линий. Не исключается возможность «короткого замыкания» одного из выходов логических элементов с одним или несколькими входами. В этом случае неисправность может проявлять себя как возникновение автогенерации либо преобразование комбинационной схемы в последовательностную.

Выделяют также класс неисправностей *перепутывания* связей, которые обусловлены ошибками, возникающими при проектировании и производстве цифровых схем. Число неисправностей данного класса, так же как и мостиковых, достаточно велико.

Типовые задачи тестового диагностирования. Центральными (общими) задачами тестового диагностирования являются:

– построение проверяющего (обнаруживающего) теста, оценка его эффективности;

– построение теста поиска неисправностей (локализующего теста), оценка его эффективности.

Частными задачами являются так называемые прямая и обратная задачи моделирования.

Поясним постановку названных задач с использованием исчерпывающей модельной информации об одиночных и константных неисправностях конкретной схемы (рис. 2.3), взятой для проведения исследований.

На схеме точками 1–9 обозначены контрольные точки, в которых будут моделироваться неисправности.

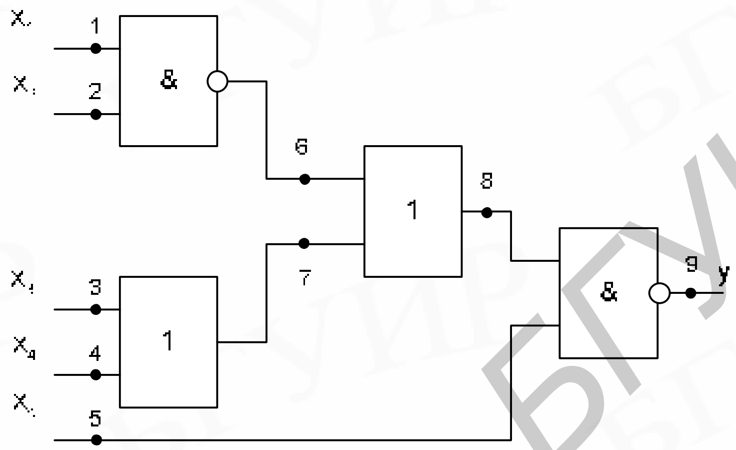


Рис. 2.3. Схема для моделирования информации об одиночных и константных неисправностях

Эти данные сведены в ***таблицу функций неисправностей*** (ТФН) (табл. 2.1).

Таким образом, для класса одиночных константных неисправностей в исследуемой схеме можно промоделировать всего 18 неисправностей: по две «константа 0» и «константа 1» в девяти полюсах схемы (контрольных точках). На рисунках и в таблицах функций неисправности будем обозначать как $N_{КТ}/_{тип}$.

ТФН получается в результате перебора всех возможных неисправностей заданного класса и вычисления реакций схемы при любых входных воздействиях (под это определение не попадают последовательностные схемы).

Например, $2/_0$ – это неисправность «константа 0» в контрольной точке 2. Столбец Y – это обычная таблица истинности комбинационной схемы, которая соответствует исправному состоянию.

Все или по крайней мере большинство последующих методов синтеза тестов и анализа схем основаны на следующей аксиоме: некоторый набор t_i обнаруживает некоторую неисправность F_j в тех случаях, когда значения функции в исправном $y_i(X)$ и неисправном $f_{ij}(X)$ состояниях не совпадают, т.е.

$$t_i \to y_i(X) \oplus f_{ij}(X) = 1. \qquad (1)$$

Таблица 2.1

Таблица функций неисправностей

	X_1	X_2	X_3	X_4	X_5		Y		$1_{/0}$	$1_{/1}$	$2_{/0}$	$2_{/1}$	$3_{/0}$	$3_{/1}$	$4_{/0}$	$4_{/1}$	$5_{/0}$	$5_{/1}$	$6_{/0}$	$6_{/1}$	$7_{/0}$	$7_{/1}$	$8_{/0}$	$8_{/1}$	$9_{/0}$	$9_{/1}$
0	0	0	0	0	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
1	1	0	0	0	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
2	0	1	0	0	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
3	1	1	0	0	0		1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
4	0	0	1	0	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
5	1	0	1	0	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
6	0	1	1	0	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
7	1	1	1	0	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
8	0	0	0	1	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
9	1	0	0	1	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
10	0	1	0	1	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
11	1	1	0	1	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
12	0	0	1	1	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
13	1	0	1	1	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
14	0	1	1	1	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
15	1	1	1	1	0		1		1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1
16	0	0	0	0	1		0		0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1
17	1	0	0	0	1		0		0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	1
18	0	1	0	0	1		0		0	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1
19	1	1	0	0	1		1		0	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	0	1
20	0	0	1	0	1		0		0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
21	1	0	1	0	1		0		0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
22	0	1	1	0	1		0		0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
23	1	1	1	0	1		0		0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	1
24	0	0	0	1	1		0		0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
25	1	0	0	1	1		0		0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
26	0	1	0	1	1		0		0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
27	1	1	0	1	1		0		0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	1
28	0	0	1	1	1		0		0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
29	1	0	1	1	1		0		0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
30	0	1	1	1	1		0		0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
31	1	1	1	1	1		0		0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1

Вначале сформулируем прямую и обратную задачи моделирования и покажем, как они решаются по ТФН.

Прямая задача – определить множество неисправностей, обнаруживаемых на заданном наборе.

Обратная задача – определить множество наборов, выявляющих заданную неисправность.

Например, в ТФН (см. табл. 2.1), применив правило (1) к набору 00001, получим неисправности $\{5/_0,\ 6/_0,\ 8/_0,\ 9/_1\}$. И для обратной задачи: неисправность $6/_0$ на основании правила (1) выявляется {00001, 10001, 01001} наборами. Отметим, что рассмотренные задачи моделирования имеют важное, но лишь теоретическое значение.

Сформулируем задачу построения проверяющего теста и оценки его эффективности.

Проверяющим тестом называется совокупность элементарных проверок (для комбинационной схемы – входных наборов), которая обеспечивает выявление большинства (в предельном случае – всех) неисправностей заданного класса.

Синтез проверяющих тестов в учебных целях осуществляется для одиночных константных неисправностей, поскольку этот класс является наиболее ограниченным, а следовательно, простым для применения формальных алгоритмов синтеза. При этом существует эмпирически подтвержденное утверждение, что проверяющий тест, построенный для одиночных неисправностей, выявляет кратные неисправности данного класса.

Выполнив ряд прямых задач моделирования для различных наборов, легко убедиться, что каждый из наборов t_i (элементарная проверка) обнаруживает некоторое подмножество неисправностей $\left\{F_j\right\}$. Тогда проверяющий тест, как совокупность наборов, объединяет подмножества неисправностей, обнаруживаемых каждым из них:

$$\{t_i\} \rightarrow \{\cup\ F_j\},\ i = \overline{1{,}2^n}\,,$$

где n – число входов комбинационной схемы.

Очевидно, что для комбинационных схем проверка таблицы истинности на всех входных наборах является проверяющим *исчерпывающим* тестом. Тогда длина теста (число тестовых наборов):

$$L = 2^n.$$

Исчерпывающие тесты не требуют введения каких-либо моделей неисправностей. Также очевидно, что с ростом числа n длина теста неоправданно растет и при $n > 20$ становится практически нереализуемой. Тогда задача синтеза тестов состоит в выборе из всей совокупности элементарных проверок тех, которые гарантируют проверку работоспособности всех элементов или, что то же самое, отсутствие неисправностей заданного класса. Другими словами, классическая задача построения теста сводится к минимизации числа тестовых наборов и максимизации числа обнаруживаемых неисправностей. В предельном случае, когда тест выявляет все неисправности, проверяющий тест называется *полным*.

Качество проверяющего теста характеризуется *коэффициентом полноты проверки работоспособности* $K_{ПП}$:

$$K_{ПП} = \frac{N_{обн}}{N_{общ}},$$

где $N_{обн}$ – число неисправностей, обнаруживаемых тестом; $N_{общ}$ – общее число неисправностей заданного класса.

Для полного теста $K_{ПП}$=1, или 100 %.

Решение задачи синтеза полного теста для исследуемой схемы (см. рис. 2.1) по ТФН (см. табл. 2.1) сводится к решению классической задачи покрытия, которая применяется в задачах синтеза и минимизации дискретных схем. Поэтому иногда коэффициент $K_{ПП}$ называют коэффициентом полноты покрытия неисправностей. Трудоемкость построения тестов $Q = km^3$ связана кубической зависимостью с числом элементов в схеме m, а задача построения оптимальных тестов относится к классу NP – полных задач. Формальное решение данной задачи не входит в содержание данного курса, поэтому приведем возможные решения, полученные покрытием ТФН с использованием эвристического подхода:

1) выбрать тестовый набор с максимальным числом выявляемых неисправностей;

2) отметить эти неисправности и исключить их из рассмотрения;

3) для оставшихся неисправностей повторить пп. 1 и 2.

Таким образом, для исследуемой схемы наборы {11110, 10001, 01001, 11001, 11101,11011}, полученные в результате покрытия ТФН, выявляют все неисправности.

Представляет интерес эмпирическая зависимость коэффициента полноты проверки от длины теста (рис. 2.4).

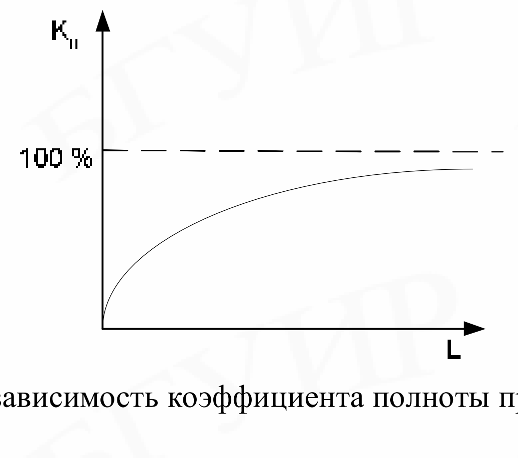


Рис. 2.4. Эмпирическая зависимость коэффициента полноты проверки от длины теста

На практике для сложных вычислительных устройств $K_{ПП}$ не достигает 100 %. Длина теста L и $K_{ПП}$ определяется в результате оптимизации технических решений.

Задача синтеза тестов поиска неисправностей сводится к нахождению совокупности наборов, максимально различающих неисправность, т.е.

$$\left\{t_i\right\} \to \left\{\cap F_j\right\}, \quad i = \overline{1{,}2^n}\,.$$

Эта задача может быть решена теоретически на уровне ТФН, однако ее практическая значимость невысока, поэтому не рассматривается в рамках учебного курса.

Из приведенных примеров видно, что решение прямой и обратной задач моделирования, а также задачи синтеза тестов по ТФН связано с большим объемом вычислений, причем трудоемкость этих задач растет по экспоненте, в зависимости от сложности объекта диагностирования. Альтернативным является подход, основанный на анализе структуры объекта диагностирования и реализуемых функций. В настоящем учебном курсе этот подход представлен методом конкурентного моделирования, методом булевой производной и методом активизации одномерных путей.

Заметим, что круг методов детерминированного синтеза тестов гораздо шире, назовем лишь метод *D*-кубов и метод эквивалентной нормальной формы. Но в рамках курса мы ограничимся лишь ранее названными методами. Несмотря на внешнее различие, их объединяет общая идея, которая может быть сформулирована в следующем виде.

Для обнаружения некоторой неисправности в схеме должны одновременно выполняться два условия:

1) условие проявления неисправности;

2) условие транспортировки неисправности на выход (полюс наблюдения).

Условие проявления неисправности (константной) в заданной контрольной точке состоит в формировании логического сигнала, противоположного данной неисправности.

Например, рассмотрим фрагмент логической схемы (рис. 2.5) в виде цепочки последовательно соединенных элементов.

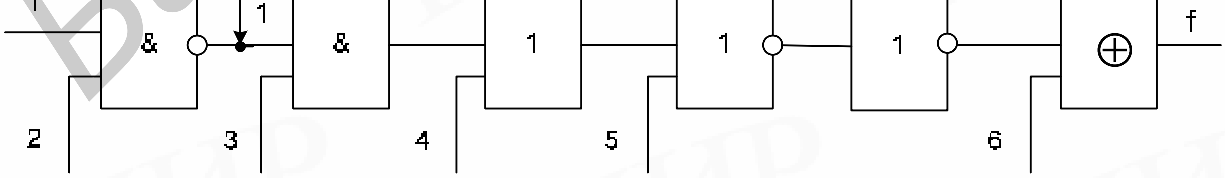


Рис. 2.5. Фрагмент логической схемы

Пусть нас интересует условие проявления неисправности «константа 1» на выходе элемента И–НЕ. Тогда на выходах 1, 2 должна быть такая комбинация, чтобы в контрольной точке был «логический 0», т.е. $\{1{,}1\} = \{1{,}1\}$.

Теперь поясним условие транспортировки неисправности на том же рис. 2.5. Цель транспортировки – наблюдение изменения сигнала на выходе f при изменении сигнала на входе 1. Это будет возможно, если «все вентили будут открыты», т.е. на входы 2, 3 элементов И, И–НЕ поданы $\{1{,}1\}$, а на входы 4, 5 элементов ИЛИ, ИЛИ–НЕ поданы $\{0{,}0\}$. Для элемента MOD2 условие транспортировки выполняется как при логическом 0, так и при логической 1.

Метод конкурентного моделирования. С помощью метода конкурентного моделирования можно решить прямую задачу моделирования, т.е. для заданного набора обнаружить множество выявляемых неисправностей.

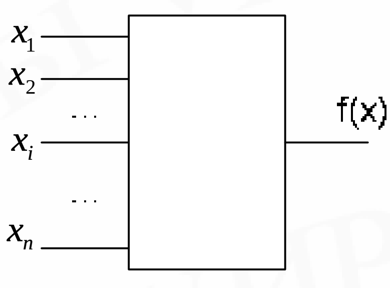
Алгоритм реализации состоит из следующих этапов:

1. На всех полюсах схемы помечается исправное состояние.
2. Инверсия исправного состояния является неисправностью, которая рассматривается как подозреваемая на обнаружение (условие проявления неисправности).
3. Для каждой неисправностей рассматривается возможность транспортировки к выходу. В списке обнаруженных неисправностей оставляют те, для которых выполнено условие транспортировки на выход.

<u>Пример</u>. Для исследуемой схемы (см. рис. 2.3) возьмем набор 00000 и определим, какие из 18 константных неисправностей он выявляет. Данный набор 00000 выявляет неисправности 5/1 и 9/0, при них изменится значение на выходе схемы. Остальные неисправности данным набором не выявляются. Сверим результаты, полученные методом конкурентного моделирования, с таблицей функций неисправностей. Результаты должны совпадать.

Метод булевой производной. С помощью метода булевой производной (БП) можно решить обратную задачу моделирования, т.е. для заданной неисправности вычислить наборы, которые будут выявлять эту неисправность.

Булева производная определяет значения логических переменных $x_1, x_2 \dots x_n$, кроме x_i, при которых изменение состояния x_i приводит к изменению значения функции $f(x)$:



Математически БП $f(x)$ по переменной x_i записывается в следующем виде:

$$\frac{df(x)}{dx_i} = f(x_1, x_2, ..., 1, ..., x_n) \oplus f(x_1, x_2, ..., 0, ..., x_n).$$

Тестом для неисправности $x_i/0$ является значение логических переменных, при которых

$$\frac{x_i df(x)}{dx_i} = 1,$$

для неисправностей $x_i/1$

$$\frac{\overline{x_i} df(x)}{dx_i} = 1.$$

Таким образом, требование инверсии сигнала на входе – не что иное, как реализация условия проявления неисправности, а равенство производной единице – не что иное, как выполнение условия транспортировки (1).

Пример. Для исследуемой схемы найдем тест для неисправностей контрольной точки X_5, применяя метод булевой производной.

В ходе решения задачи следует восстановить сокращенную дизъюнктивную нормальную форму (СДНФ) с использованием эквивалентных булевых преобразований:

$$\frac{\partial f(x)}{\partial x_5} = (\overline{x_1 x_2} \vee (x_3 \vee x_4))1 \oplus (\overline{x_1 x_2} \vee (x_3 \vee x_4))0.$$

Используя правило де Моргана $\overline{x_1 + x_2} = \overline{x_1}\,\overline{x_2}$ и раскрывая скобки, получим:

$$\frac{\partial f(x)}{\partial x_5} = (\overline{x_1 x_2} \vee (x_3 \vee x_4))1 = \overline{x_1 x_2} \vee (x_3 \vee x_4) = \overline{x_1} \vee \overline{x_2} \vee x_3 \vee x_4 = \overline{x_1}(x_2 \vee \overline{x_2})(x_3 \vee \overline{x_3})(x_4 \vee \overline{x_4}) \vee$$
$$(x_1 \vee \overline{x_1})\overline{x_2}(x_3 \vee \overline{x_3})(x_4 \vee \overline{x_4}) \vee (x_1 \vee \overline{x_1})(x_2 \vee \overline{x_2})x_3(x_4 \vee \overline{x_4}) \vee (x_1 \vee \overline{x_1})(x_2 \vee \overline{x_2})(x_3 \vee \overline{x_3})x_4 =$$
$$= \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4} \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3}x_4 \vee \overline{x_1}\,\overline{x_2}x_3\overline{x_4} \vee \overline{x_1}\,\overline{x_2}x_3x_4 \vee \overline{x_1}x_2\overline{x_3}\,\overline{x_4} \vee \overline{x_1}x_2x_3\overline{x_4} \vee x_1\overline{x_2}x_3\overline{x_4} \vee \overline{x_1}x_2x_3x_4 \vee$$
$$x_1\overline{x_2}\,\overline{x_3}x_4 \vee x_1x_2x_3x_4 \vee x_1\overline{x_2}x_3x_4 \vee x_1\overline{x_2}x_3\overline{x_4} \vee x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4} \vee x_1x_2x_3\overline{x_4} \vee x_1x_2\overline{x_3}x_4.$$

Тестовые наборы для неисправности $x_5/0$ вычислим из условия $\frac{x_5 df(x)}{dx_5} = 1$:

$$\overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}x_5 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_3}x_4x_5 \vee \overline{x_1}\,\overline{x_2}x_3\overline{x_4}x_5 \vee \overline{x_1}\,\overline{x_2}x_3x_4x_5 \vee \overline{x_1}x_2\overline{x_3}\,\overline{x_4}x_5 \vee \overline{x_1}x_2x_3\overline{x_4}x_5 \vee x_1\overline{x_2}x_3\overline{x_4}x_5 \vee$$
$$\overline{x_1}x_2x_3x_4x_5 \vee x_1\overline{x_2}\,\overline{x_3}x_4x_5 \vee x_1x_2x_3x_4x_5 \vee x_1\overline{x_2}x_3x_4x_5 \vee x_1\overline{x_2}x_3\overline{x_4}x_5 \vee x_1\overline{x_2}\,\overline{x_3}\,\overline{x_4}x_5 \vee x_1x_2x_3\overline{x_4}x_5 \vee$$
$$x_1x_2\overline{x_3}x_4x_5 = 1.$$

Из СДНФ получим наборы, выявляющие неисправность $x_5/0$, которые приведены в табл. 2.2.

Тестовые наборы для неисправности $x_5/1$ вычислим из условия $\frac{\overline{x_5} df(x)}{dx_5} = 1$:

$$\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5 \vee \bar{x}_1\bar{x}_2\bar{x}_3 x_4\bar{x}_5 \vee \bar{x}_1\bar{x}_2 x_3\bar{x}_4\bar{x}_5 \vee \bar{x}_1\bar{x}_2 x_3 x_4\bar{x}_5 \vee \bar{x}_1 x_2\bar{x}_3\bar{x}_4\bar{x}_5 \vee \bar{x}_1 x_2 x_3\bar{x}_4\bar{x}_5 \vee x_1\bar{x}_2 x_3\bar{x}_4\bar{x}_5 \vee$$
$$\bar{x}_1 x_2 x_3 x_4\bar{x}_5 \vee x_1\bar{x}_2\bar{x}_3 x_4\bar{x}_5 \vee x_1 x_2 x_3 x_4\bar{x}_5 \vee x_1\bar{x}_2 x_3 x_4\bar{x}_5 \vee x_1\bar{x}_2 x_3\bar{x}_4\bar{x}_5 \vee x_1\bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5 \vee x_1 x_2 x_3\bar{x}_4\bar{x}_5 \vee$$
$$x_1 x_2\bar{x}_3 x_4\bar{x}_5 = 1$$

Из СДНФ получим наборы, выявляющие неисправность $x_5/1$, которые также приведены в табл. 2.2.

Таблица 2.2

Наборы, выявляющие неисправности 5/0 и 5/1

Для неисправности 5/0	Для неисправности 5/1
00001	00000
00011	00010
00101	00100
00111	00110
01001	01000
01101	01100
01011	01010
01111	01110
10011	10010
11111	11110
10111	10110
10101	10100
10001	10000
11101	11100
11011	11010

Результаты, полученные методом булевой производной, должны совпадать с ТФН.

Метод активизации путей. Метод активизации путей является инженерным методом, не гарантирующим получение идеального результата, т.е. минимальной совокупности наборов, выявляющих все заданные неисправности.

Суть метода заключается в следующем:

1. Для заданной неисправности выбрать путь – маршрут транспортировки от места возникновения к выходу.

2. Обеспечить условие проявления неисправности, т.е. подать такие сигналы на входы схемы, чтобы в контрольной точке сигнал был противоположен заданной неисправности.

3. Активизировать путь, т.е. подать такие сигналы на входы схемы, чтобы обеспечить условие транспортировки.

Для построения полного теста (теста, обнаруживающего все неисправности) без перебора всех возможных неисправностей необходимо воспользоваться следующей теоремой. *Проверка неисправностей «константа 0» и «константа 1» на входе пути выявляет все неисправности, лежащие на этом пути. Следовательно, чтобы построить проверяющий тест комбинационной* *схемы, достаточно вычислить тестовые наборы для неисправностей «константа 0» и «константа 1» на входах всех возможных путей.*

Таким образом, поочередно активизируя пути и проверяя неисправности «константа 0» и «константа 1», можно построить полный проверяющий тест.

Пример. Для схемы (рис. 2.3) вычислить наборы, выявляющие одиночные константные неисправности.

Решение:

1. В данной древовидной схеме для неисправности 1/0 имеется только один путь к выходу схемы.

2. Установим условие активности выбранного первого пути (транспортируем неисправность 1/0 со входа X_1): X_2=1, $X_3 \vee X_4$=0, X_5=1.

3. Вычислим возможное значение входных сигналов X_2=1, X_3=0, X_4=0, X_5=1.

4. Тогда набор $X_1X_2X_3X_4X_5$=11001 обнаруживает неисправность 1/0.

При активизации всех путей исследуемой схемы и проверке неисправностей «константа 0» и «константа 1» получим следующие результаты (табл. 2.3).

Исключив повторяющиеся наборы {11001, 11001}, получим проверяющий тест комбинационной схемы, состоящий из семи наборов. Очевидно, что число тестовых наборов при методе активизации путей не является минимальным.

Таблица 2.3

Результаты, полученные для исследуемой схемы при активизации путей

	X_1	X_2	X_3	X_4	X_5
1/0	1	1	0	0	1
1/1	0	1	0	0	1
2/0	1	1	0	0	1
2/1	1	0	0	0	1
3/0	1	1	1	0	1
3/1	1	1	0	0	1
4/0	1	1	0	1	1
4/1	1	1	0	0	1
5/0	1	1	0	1	1
5/1	1	1	0	1	0

Если сравнить тест, полученный методом активизации путей с тестом, полученным при покрытии ТФН, то несложно заметить, что число и состав тестовых наборов различны. Как правило, число тестовых наборов при синтезе методом активизации путей будет большим, однако трудоемкость построения (вычислительные затраты) гораздо ниже, поскольку задача решалась перебором путей, а не наборов неисправностей.

Особенности построения тестов для реконвергентных схем. Схемы, имеющие разветвления со схождением, более сложны для анализа по сравнению с древовидными схемами. Сложность заключается в том, что условия проявления неисправности в исследуемой точке и транспортировки неисправности к выходу могут быть взаимозависимы и даже взаимоисключаемы.

Построить тест для сходящегося разветвления возможно через активизацию по отдельности каждого возможного пути либо через одновременную активизацию нескольких путей – т.е. активизацию *многомерного* пути.

Теорема синтеза полного теста (теорема Акерса) для реконвергентных схем заключается в следующем. *Множество тестовых наборов, обнаруживающих неисправности всех входов и сходящихся разветвлений, обнаруживает все одиночные константные неисправности.*

<u>Пример</u>. Дана схема со сходящимся разветвлением (рис. 2.6). Построить тест для входа x_2.

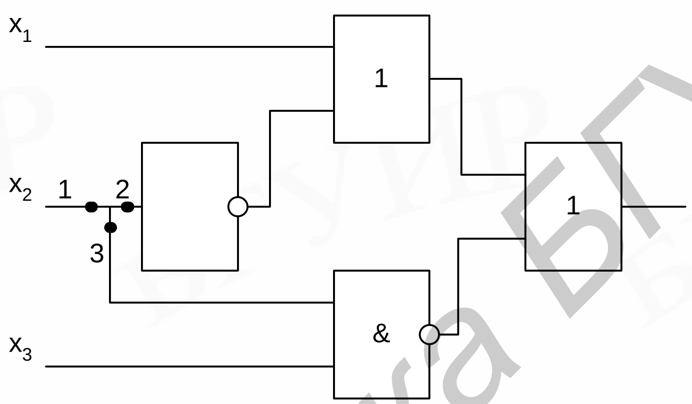


Рис. 2.6. Схема со сходящимися разветвлениями

Вход x_2 имеет сходящееся разветвление. Тогда для полного теста одиночных константных неисправностей необходимо моделировать неисправности в контрольных точках 1, 2, 3. Полагается, что неисправности в этих точках независимы. В противном случае рассматривалась бы лишь одна неисправность в КТ1.

1. Неисправности в КТ1 (1/0 и 1/1) будут выявляться соответственно наборами x_1, x_2, x_3 {011} и {001}. При этом одновременно активизируются два пути, т.е. принято говорить, что неисправность транспортируется по многомерному пути.

2. Неисправность в КТ2 (2/0) также выявляется набором x_1, x_2, x_3 {011}, а при обнаружении неисправности 2/1 возникает препятствие, так как условие ее проявления (x_2=0) несовместимы с условием транспортировки (x_2=1). Поскольку для данной неисправности существует только один путь, то эта неисправность в принципе не может быть обнаружена, т.е. для нее не существует проверяющего теста.

Смоделируем данную неисправность на схеме. В исправном состоянии схема реализует функцию $f=(x_1 \vee x_2)\vee\overline{x_2 x_3}=x_1 \vee \overline{x_2} \vee \overline{x_2} \vee \overline{x_3}=x_1 \vee \overline{x_2} \vee \overline{x_3}$.

При неисправности в КТ2 (2/1) $f=(x_1 \vee 0)\vee\overline{x_2 x_3}=x_1 \vee \overline{x_2} \vee \overline{x_3}$.

Отсюда видим, что неисправность не влияет на реализуемую функцию, а следовательно, и не имеет обнаруживающего тестового набора.

4. В КТЗ неисправность 3/0 обнаруживается набором x_1, x_2, x_3 {011}, а для неисправности 3/1 не существует тестового набора, так как условия проявления и транспортировки данной неисправности несовместимы. Моделирование данной неисправности также свидетельствует, что она не изменяет функцию схемы: $f = (x_1 \vee x_2) \vee \overline{x_3} = x_1 \vee \overline{x_2} \vee \overline{x_3}$.

Из рассмотренного примера следует, что реконвергентные схемы могут быть функционально избыточными. В функционально избыточных схемах возможны неисправности, для которых не существует обнаруживающих тестовых наборов. Результаты более глубокого анализа схем свидетельствуют, что появление необнаруживаемых неисправностей может приводить к появлению логических состязаний, паразитных импульсов и другим коллизиям в схеме.

3. ЭВРИСТИЧЕСКОЕ ТЕСТИРОВАНИЕ ТИПОВЫХ УЗЛОВ И УСТРОЙСТВ ПО ИЗВЕСТНОЙ ФУНКЦИИ

В разделе рассматриваются:

1. Тесты типовых узлов (триггера, сумматора, мультиплексора, компаратора).
2. Тестирование оперативных запоминающих устройств (ОЗУ).
3. Тестирование микропроцессоров и микропроцессорных систем.

На практике часто встречаются ситуации, когда принципиальная схема объекта диагностирования неизвестна либо применение точных детерминированных методов синтеза тестов не оправдано по временным либо экономическим соображениям. Тогда объект диагностирования рассматривается как черный ящик, реализуемая функция которого известна. Если реализовать исчерпывающий тест функции не представляется возможным, то в этом случае можно ограничиться серией отдельных функциональных проверок. Такой ***проверяющий тест*** в некоторой степени будет характеризовать работоспособность объекта диагностирования. Очевидно, что полнота теста будет не определена. Понятие функционального теста не является математически строгим и базируется на позитивном использовании имеющихся данных об объекте. Наиболее очевидными являются функциональные тесты для типовых узлов, функции которых известны.

Например, не представляет труда предложить функциональный тест четырехразрядного сумматора, который проверяет, как минимум, константные неисправности на входах и выходах сумматора, а также цепь переноса:

Операнды		Сумма S	Перенос P
а	в		
0000	0000	0000	0
1111	1111	1110	1
0000	1111	1111	0
1111	0000	1111	0

Функциональный тест мультиплексора 4→1 должен проверять прохождение 0 и 1 с информационных входов a_i на выход f, при соответствующем коде настройки U_0, U_1:

a_0	a_1	a_2	a_3	U_0	U_1	f
0	x	x	x	0	0	0
1	x	x	x	0	0	1
x	0	x	x	0	1	0
x	1	x	x	0	1	1
x	x	0	x	1	0	0
x	x	1	x	1	0	1
x	x	x	0	1	1	0
x	x	x	1	1	1	1

Функциональный тест триггеров вне зависимости от типа триггера должен проверить возможность записи информации (0 и 1) и хранение информации (0 и 1).

Функциональный тест ПЗУ заключается в считывании данных по всем адресам и сравнении их с эталоном.

Тестирование оперативных запоминающих устройств. Особенности ОЗУ как объекта диагностирования состоят в следующем:

– нет возможности прямого доступа к внутренним точкам схемы, имеется доступ, опосредованный через пару операций – запись / считывание;

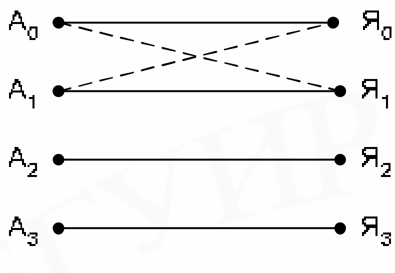
– все ячейки памяти имеют электрическую связь посредством адресных шин и шин считывания, и, как следствие, при дефектах и деградационных процессах может возникать взаимное влияние «соседних» ячеек, искажающее исправную функцию устройства.

Соседней называется та ячейка, которая может оказать влияние на хранение информации в рассматриваемой ячейке. В зависимости от характера дефекта либо деградационного процесса на кристалле соседними ячейками могут быть: а) ячейки, находящиеся по периферии рассматриваемой; б) ячейки, находящиеся в одной строке либо в одном столбце; в) любая ячейка накопителя.

В ОЗУ принято выделять следующие классы неисправностей:

1. Неисправности сумматора.

В исправном состоянии каждому адресу должна взаимно однозначно соответствовать конкретная ячейка памяти. Неисправности дешифраторов ОЗУ выделены в самостоятельный класс неисправностей:



При неисправности возможны случаи одновременного обращения к двум ячейкам по одному адресу, тогда, возможно, произойдет наложение информации либо обращение с двух адресов к одной ячейке памяти. Такая ситуация, как запись «не в заданную ячейку» и считывание из той же ячейки, не может быть протестирована одним циклом запись / считывание нулей либо единиц. Это обстоятельство накладывает специфические особенности на алгоритмы тестирования неисправностей дешифратора.

2. Неисправности накопителя, которые могут быть как кодозависимые, так и кодонезависимые.

Неисправности, возникающие вследствие физического влияния соседних ячеек, называют кодозависимыми, поскольку они проявляются не на всех сочетаниях хранимых данных, а лишь на определенных кодах. Кроме кодозависимых в накопителе возможны и кодонезависимые (константные) неисправности, обнаружение которых не представляет существенной проблемы.

Запоминающие устройства имеют регулярную структуру, поэтому функциональные тесты представляют собой последовательности записи / считывания данных при регулярном изменении адресов. Это позволяет представить тест в виде алгоритмов изменения адресов, данных, команд записи / считывания и сравнения. Функциональные тесты запоминающих устройств часто называют алгоритмами тестирования.

Самый простой тест – *«Все нули (все единицы)»***.** Во все ячейки ОЗУ производится запись всех нулей, после чего производятся последовательное считывание и проверка данных. Затем – запись всех единиц и последующие считывание и проверка. Этот тест не выявляет неисправности DC и кодозависимые неисправности.

*Адресный***.** В каждую ячейку ОЗУ записывается код собственного адреса, затем производятся последовательное считывание и проверка данных. Адресный тест обеспечивает проверку адресных дешифраторов ОЗУ.

*Шахматный***.** В ОЗУ записывается информация, имеющая шахматное распределение, затем производятся последовательное считывание и проверка данных. Шахматный код проверяет на взаимное влияние ячейки, хранящие информацию в обратном коде.

*Марш***.** По всем адресам записывается фон – нули.

Затем осуществляется прямой перебор адресов, при котором:

для каждого адреса выполняется считывание 0 и запись 1;

для каждого адреса выполняется считывание 1 и запись 0.

Затем осуществляется обратный перебор адресов, при котором:

для каждого адреса выполняется считывание 0 и запись 1;

для каждого адреса выполняется считывание 1 и запись 0.

После этого меняется фон – все единицы и цикл проверки повторяется.

*Крест***.** В каждую ячейку ОЗУ записываются тестовые данные (нули или единицы), а в каждую из четырех смежных ячеек – фоновые данные (единицы или нули соответственно). Затем информация фона инвертируется и проверяется влияние этого изменения на тестируемую ячейку.

С помощью этого теста проверяется чувствительность ячейки к изменениям состояний крестообразно расположенных соседних ячеек.

Бегущий. В первую ячейку ОЗУ записывается 1 (0), а во все остальные – фон – нули (единицы). Затем все адреса последовательно считываются с проверкой; последней считывается первая ячейка с последующей записью в нее 0 (1). Последовательность повторяется для второй ячейки, третей и т.д. вплоть до последней.

Тест «бегущий» предназначен для обнаружения сбоев в ОЗУ, вызванных в разрядных цепях, так как перемещение 0 на фоне 1 (и наоборот) создает наихудшие условия для усилителей считывания.

Галоп. В первую ячейку ОЗУ записывается 1, а во все остальные – нули. Затем последовательно считываются и проверяются ячейки 2, 1, 2, потом 3, 1, 3 и т.д., пока все пары переходов, включая первую ячейку, не будут проверены. После этого в первую ячейку записывается 0 и данные считываются. Последовательность операций повторяется для ячеек 2, 3 и т.д. вплоть до последней.

Очевидно, что тесты имеют различное число операций (тестовых наборов L). При этом тесты можно объединить в три больших класса – пропорциональные $N, N^2 N^{3/2}$, где N – число ячеек накопителя.

$$L = K_1 N^2;$$

$$L = K_2 N^{3/2};$$

$$L = K_3 N\ .$$

K_1, K_2, K_3 – коэффициенты пропорциональности, обусловленные спецификой теста.

Тесты, пропорциональные N («марш», «крест»), используются в качестве встроенных тестов при проверке работоспособности вычислительных средств и наиболее широко применяются.

Тесты, пропорциональные $N^2, N^{3/2}$ («бегущий», «галоп», «пинг-понг»), используются в условиях производства, при проведении исследовательских испытаний, так как требуют больших временных ресурсов.

Тестирование микропроцессоров и микропроцессорных систем. Микропроцессоры и микропроцессорные системы представляют собой сложные объекты диагностирования, для которых не существует общих формальных методов синтеза тестов, таких, как мы имеем для комбинационных схем. «Сложность» объекта диагностирования заключается в следующих схемных особенностях.

Объем дискретных элементов составляет сотни тысяч и миллионы вентилей. Анализ объекта в целом на вентильном уровне практически неосуществим.

Структура объекта диагностирования неоднородна, иерархична, т.е. состоит из различных функциональных узлов с самыми различными видами топологических соединений: возможны аналоговые узлы, асинхронные узлы, генераторы, формирователи и т.п. Ограничен доступ к требуемому участку схемы для постановки диагностического эксперимента. Поэтому для микропроцессоров, как объектов диагностирования, класс константных неисправностей является неприемлемым. В данном случае микропроцессор либо микропроцессорная система рассматривается как программно-управляемый объект, для которого наиболее удобно рассматривать неисправности типа: неисполнение команды, неверное исполнение команды, исполнение незаданной команды и т.п.

Тест микропроцессора выглядит как проверка отдельных функций системы команд. Очевидно, что полнота теста будет определяться объемом выполненных команд, причем, особенно для команд регистровых пересылок, следует проверять правильность передачи как нулей, так и единиц.

При составлении теста микропроцессорной системы используют так называемую стратегию «раскрутки». То есть определяется некоторое ядро (как правило, ввод/вывод, простейшие команды пересылок и минимальная индикация), которое тестируется в первую очередь. Последующие проверки включают новые узлы объекта диагностирования (при этом полагают, что проверенная часть работоспособна).

Тест микропроцессора строится как совокупность функциональных тестов отдельных узлов и модулей:

– тест канала ввода / вывода;
– тест арифметико-логического устройства;
– тест программно-доступных регистров общего назначения;
– тест таймера;
– тест ОЗУ, ПЗУ;
– тест интерфейсов и периферийных устройств.

На этапе проектирования вычислительной системы синтез теста сопровождается логическим и событийным моделированием отдельных неисправностей на заданных функциональных проверках.

4. СРЕДСТВА ТЕСТОВОГО ДИАГНОСТИРОВАНИЯ: СИГНАТУРНЫЕ АНАЛИЗАТОРЫ, ГЕНЕРАТОРЫ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Выполнение тестирования при большом объеме тестовых воздействий требует наличия устройств для автоматической генерации тестов, для хранения эталонов реакций объекта на эти тесты, а также схемы сравнения. На производстве электронных устройств подобные тестеры являются необходимым компонентом технологического оборудования, основу которых, как правило, составляет персональная ЭВМ с необходимым специализированным интерфейсом, посредством которого объект диагностирования (ОД) с ней взаимодействует. Очевидно, что стоимость такого оборудования будет значительной и экономически оправданной только при соответственном объеме выпускаемой продукции. Альтернативой подобного рода универсальным тестерам являются *специализированные аппаратные средства компактного тестирования*.

Общая идея методов компактного тестирования состоит в том, что тестовые наборы не хранятся в памяти, а аппаратно генерируются в реальном времени, а реакция объекта на тест сжимается в некоторую интегральную характеристику, ключевое слово, которое и сравнивается с эталоном. По способу применения такие средства тестирования могут быть как *внешними* по отношению к объекту диагностирования, так и *встроенными* в ОД.

Так, например, для комбинационного устройства обычный счетчик может служить генератором исчерпывающего теста, а в качестве интегральной характеристики соответствия анализируемой и эталонной последовательностей может служить информация о четности (нечетности) единиц в анализируемой последовательности; информация о числе единиц (нулей); информация о числе переходов из нуля в единицу либо из единицы в ноль. Однако наибольшее распространение для сжатия информации при тестировании устройств получили устройства, названные сигнатурными анализаторами (СА).

Сигнатурный анализ. Первый сигнатурный анализатор HP5004A был выпущен фирмой «Hewlett – Packard» в середине 70-х гг. прошлого века. Он представлял собой несложный прибор, предназначенный для обнаружения ошибок в последовательности анализируемых данных, вызванных неисправностями контролируемого цифрового устройства.

Его схема (рис. 4.1) состояла из 16-разрядного циклического регистра сдвига, 5-входового сумматора по модулю два, а также нескольких логических элементов для формирования сигналов управления – СТАРТ и СТОП. Анализируемая последовательность поступала на сумматор и складывалась по MOD2 с содержимым отдельных разрядов регистра (в частности 7, 9, 12, 16) и циклически подавалась на вход регистра. Выбор обратных связей с разрядов 7, 9, 12 и 16 в сигнатурном анализаторе HP5004A был определен полиномом делителя, о чем подробно будет говориться ниже. Шестнадцать разрядов, разделенных на четыре тетрады, через дешифраторы подключались к семисегментным индикаторам с целью удобного наблюдения результатов сжатия содержимого регистра в виде четырех шестнадцатеричных цифр.

Данный анализатор долгое время являлся неофициальным стандартом устройств подобного типа и, можно сказать, стал прототипом для целого класса устройств – сигнатурных анализаторов.

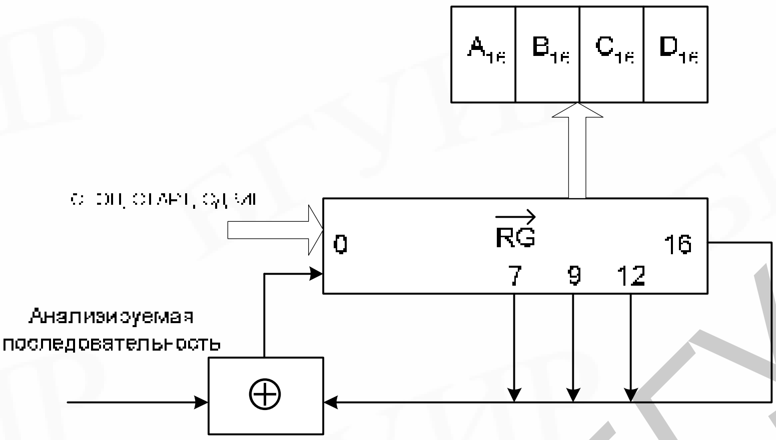


Рис. 4.1. Схема первого сигнатурного анализатора

Анализируемая последовательность синхронно последовательным кодом поступает на вход сигнатурного анализатора и «сжимается» в ключевое слово – *сигнатуру* (от англ. signature – подпись). В дальнейшем в технической диагностике слово «сигнатура» стало иметь собственное значение – как ключевое слово сжатой последовательности данных. Управляющими сигналами сигнатурного анализатора являются СТАРТ, СТОП и СДВИГ. Сигналы СТАРТ и СТОП формируют временной интервал («окно»), в течение которого осуществляется процедура сжатия информации на анализаторе. Под действием сигнала СТАРТ элементы памяти регистра сдвига устанавливаются в исходное состояние, как правило нулевое, а сам регистр начинает выполнять функцию сдвига на один разряд вправо под действием синхронизирующих сигналов СДВИГ.

В основе сигнатурного анализа лежит аппарат деления полиномов, а сам циклический регистр сдвига с сумматором по MOD2 обратными связями является несложным устройством, реализующим такое деление. Так, любую анализируемую последовательность бит можно представить в виде полинома n-й степени:

$$y(x)=c_n x^n \oplus c_{n-1}x^{n-1} \oplus \ldots \oplus c_1 x^1 \oplus c_0 x^0,$$

где c_i – значение i-го разряда последовательности.

Эту последовательность можно разделить на некоторый полином – делитель и получить в результате частное и остаток:

$$\frac{y(x)}{g(x)}=p(x)+S(x),$$

где $y(x)$ – делимое; $g(x)$ – делитель; $p(x)$ – частное; $S(x)$ – остаток, или сигнатура.

Идея сжатия состоит в том, что разрядность сигнатуры ***S(x)*** много меньше разрядности анализируемой последовательности ***y(x),*** следовательно, гораздо экономнее сохранять компактную сигнатуру-эталон, нежели эталонную последовательность – ***y(x).***

Деление полиномов выполняется на циклическом регистре сдвига, у которого определенные обратные связи складываются по MOD2. Различают два варианта структур сигнатурных анализаторов: с внутренними сумматорами (рис. 4.2) и с внешними сумматорами (рис. 4.3).

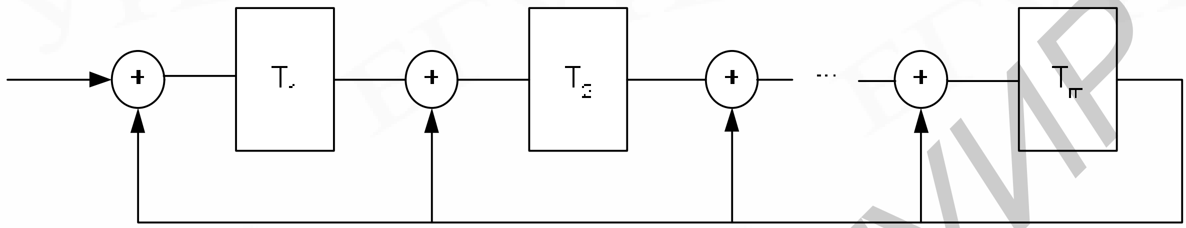


Рис. 4.2. Сигнатурный анализатор с внутренними сумматорами

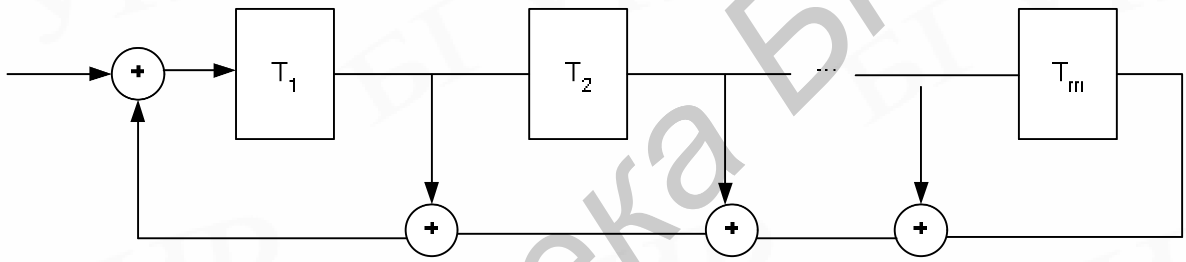


Рис. 4.3. Сигнатурный анализатор с внешними сумматорами

Структура сигнатурного регистра однозначно соответствует полиному делителя.

В общем виде структура СА описывается полиномом m-й степени:

$$\boldsymbol{g(x)=b_m x^m \oplus b_{m-1}x^{m-1} \oplus + \ \dots \oplus b_1 x^1 \oplus b_0 x^0,}$$

где $\boldsymbol{b_j}$=1 соответствуют наличию обратной связи, причем первый и последний член в делителе присутствуют всегда.

Функционирование сигнатурного анализатора (с внутренними сумматорами) описывается следующими рекуррентными выражениями. С приходом каждого синхронизирующего сигнала СДВИГ в первый разряд регистра a_1 записывается значение

$$a_1(i) = y(i) \oplus a_m(i-1),$$

где $y(i) \in \{0,1\}$ – символ текущего разряда сжимаемой последовательности; $a_m(i-1)$ – содержимое старшего разряда регистра в предыдущем такте.

Сдвиг данных в регистре описывается соотношением

$$a_j(i) = a_{j-1}(i-1) \oplus b_j a_m(i-1), \ \ j = \overline{2,m}.$$

Здесь $a_j(i) \in \{0,1\}$ – содержимое j-го разряда регистра сдвига в $\boldsymbol{i}$-й такт; $b_j \in \{0,1\}$ – коэффициенты полинома делителя $g(x)$.

Аналогичная математическая модель сигнатурного анализатора для структур с внешними сумматорами записывается в виде

$$a_1(i) = y(i) \oplus \sum_{j=1}^{m} b_j a_j(i-1).$$

Пример 1

Рассмотрим аналитический и имитационный способы получения остатка для сигнатурного анализатора с внутренними сумматорами.

1. Зададим анализируемую последовательность из 12 бит: 110100111011.

2. Выберем порождающий полином из перечня примитивных неприводимых: $g(x) = x^5 \text{ Å } x^3 \text{ Å } 1$.

3. Представим анализируемую последовательность в виде полинома:

$y(x)=1\cdot x^{11}\text{Å}1\cdot x^{10}\text{Å}0\cdot x^{9}\text{Å}1\cdot x^{8}\text{Å}0\cdot x^{7}\text{Å}0\cdot x^{6}\text{Å}1\cdot x^{5}\text{Å}1\cdot x^{4}\text{Å}1\cdot x^{3}\text{Å}0\cdot x^{2}\text{Å}1\cdot x^{1}\text{Å}1=$
$=x^{11}\text{Å}x^{10}\text{Å}x^{8}\text{Å}x^{5}\text{Å}x^{4}\text{Å}x^{3}\text{Å}x\text{Å}1$.

Разделим полученный полином на выбранный порождающий.

Аналитический вариант деления полинома:

$x^{11}\oplus x^{10}\oplus x^{8}\oplus x^{5}\oplus x^{4}\oplus x^{3}\oplus x\oplus 1$ | $\underline{x^5\oplus x^3\oplus 1}$ - делитель

$\underline{x^{11}\oplus x^{9}\oplus x^{6}}$ | $x^6\oplus x^5\oplus x^4\oplus x^2\oplus x\oplus 1$ – q(x), частное

$x^{10}\oplus x^{9}\oplus x^{8}\oplus x^{6}\oplus x^{5}\oplus x^{4}\oplus x^{3}\oplus x\oplus 1$

$\underline{x^{10}\oplus x^{8}\oplus x^{5}}$

$x^{9}\oplus x^{6}\oplus x^{4}\oplus x^{3}\oplus x\oplus 1$

$\underline{x^{9}\oplus x^{7}\oplus x^{4}}$

$x^{7}\oplus x^{6}\oplus x^{3}\oplus x\oplus 1$

$\underline{x^{7}\oplus x^{5}\oplus x^{2}}$

$x^{6}\oplus x^{5}\oplus x^{3}\oplus x^{2}\oplus x\oplus 1$

$\underline{x^{6}\oplus x^{4}\oplus x}$

$x^{5}\oplus x^{4}\oplus x^{3}\oplus x^{2}\oplus 1$

$\underline{x^{5}\oplus x^{3}\oplus 1}$

$x^4\oplus x^2$ – S(x), остаток (сигнатура)

Числовой вариант деления полинома:

```
   1 1 0 1 0 0 1 1 1 0 1 1 | 1 0 1 0 0 1
⊕  1 0 1 0 0 1             | 1 1 1 0 1 1 1
   -----------
     1 1 1 0 1 1
⊕    1 0 1 0 0 1
     -----------
       1 0 0 1 0 1
⊕      1 0 1 0 0 1
       -----------
         1 1 0 0 1 0
⊕        1 0 1 0 0 1
         -----------
           1 1 0 1 1 1
⊕          1 0 1 0 0 1
           -----------
             1 1 1 1 0 1
⊕            1 0 1 0 0 1
             -----------
               1 0 1 0 0   →   x^4⊕x^2
```

4. Синтезируем сигнатурный анализатор с внутренними сумматорами на основе полинома $g(x) = x^5$ Å x^3 Å 1 и смоделируем динамику получения сигнатуры для анализируемой последовательности (необходимо обратить внимание на то, что анализируемая последовательность подается старшими разрядами вперед) – рис. 4.4.

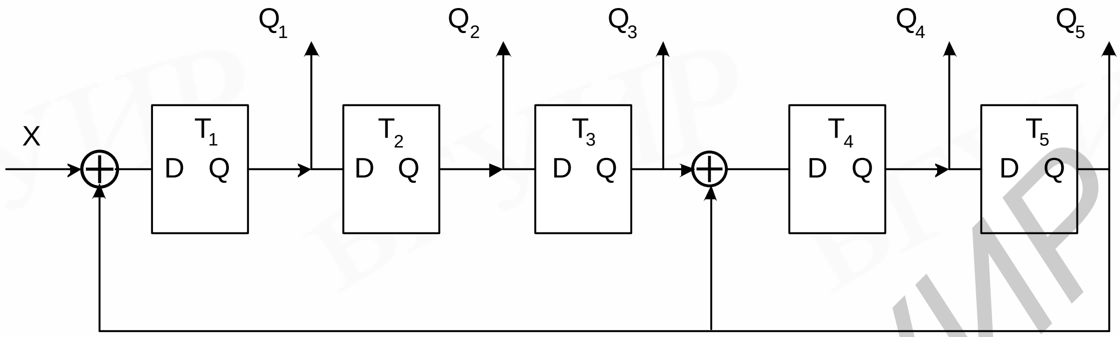


Рис. 4.4. Имитационное моделирование деления полиномов на сигнатурном анализаторе с внутренними сумматорами с делителем $g(x) = x^5 \oplus x^3 \oplus 1$

110111001011	0	0	0	0	0	
11011100101	1	0	0	0	0	
1101110010	1	1	0	0	0	
110111001	0	1	1	0	0	
11011100	1	0	1	1	0	
1101110	0	1	0	1	1	
110111	1	0	1	1	1	1
11011	0	1	0	0	1	11
1101	0	0	1	1	0	111
110	1	0	0	1	1	0111
11	1	1	0	1	1	10111
1	0	1	1	1	1	110111
	0	0	1	0	1	1110111
	сигнатура					частное

5. Сравнивая сигнатуры, полученные аналитически и в результате моделирования, наблюдаем идентичные результаты.

6. Теперь изменим один бит анализируемой последовательности и вместо последовательности 1101001**1**1011 возьмем последовательность 1101001**0**1011.

7. Повторно смоделируем динамику получения сигнатуры для последовательности с заданной ошибкой либо получим сигнатуры в результате деления полинома. Сигнатура, полученная при ошибочной последовательности (00100), не совпадает с сигнатурой, полученной при исходной (верной) последовательности (10100).

Существует формальное преобразование для сопоставления результатов, соответствующих делению полиномов на СА с внешними и внутренними сумматорами:

$\frac{y(x)}{g(x)} = p(x) + S(x)$ – с внутренними сумматорами;

$\frac{y(x)}{\Psi(x)}$=*p*$^{/}$*(x)*+*S*$^{/}$*(x)* – с внешними сумматорами, где $\Psi(x)$ – полином, обратный *g(x)*.

$$\Psi(\mathrm{x}) = g^{-1}(x).$$

Тогда *S(x)=M×S*$^{/}$*(x)*, где *M* – матрица, составленная из коэффициентов $\boldsymbol{b_j}$ полинома делителя g(x).

$$M = \begin{vmatrix} b_m & 0 & 0 & \dots & 0 \\ b_{m-1} & b_m & 0 & \dots & 0 \\ & & \dots. & & \\ & & & & \\ b_1 & b_2 & b_3 & \dots & b_m \end{vmatrix}.$$

Продолжим пример:

1. Для полинома $g(x) = x^5$ Å x^3 Å 1 найдем обратный полином:

$\Psi(x) = x^m g^{-1}(x) = x^5(x^{-5}$ Å x^{-3} Å $1) = 1$Å x^2 Åx^5.

2. С помощью имитационного моделирования найдем $S^{/}(x)$=10000.

3. Матрица, составленная из коэффициентов $\boldsymbol{b_j}$ полинома делителя *g(x)*, следующая:

$$M = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{vmatrix}.$$

4. Найдем *S(x)=M×S*$^{/}$*(x)*:

$$S(x) = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{vmatrix} \times \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{vmatrix} = \begin{vmatrix} 10+00+00+00+01 \\ 00+10+00+00+01 \\ 10+00+10+00+01 \\ 00+10+00+10+01 \\ 10+00+10+00+11 \end{vmatrix} = \begin{vmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{vmatrix}.$$

Достоверность сигнатурного анализа. Из элементарной алгебры известно, что существуют различные числа (последовательности), которые при делении на один делитель имеют одинаковые остатки.

Например, возьмем две последовательности {11110101} и {11101111} и разделим на полином {1101}:

```
⊕ 1 1 1 1 0 1 0 1 | 1 1 0 1
  1 1 0 1         | 1 0 1 1 1
  -------
 ⊕    1 0 0 1
      1 1 0 1
      -------
  ⊕     1 0 0 0
        1 1 0 1
        -------
   ⊕      1 0 1 1
          1 1 0 1
          -------
            1 1 0
```

```
⊕  1 1 1 0 1 1 1 1 | 1 1 0 1
   1 1 0 1         | 1 0 1 0 1
   ---------
    ⊕  1 1 1 1
       1 1 0 1
       ---------
        ⊕  1 0 1 1
           1 1 0 1
           -------
             1 1 0
```

Получили одинаковые остатки {110}. Это означает, что существует вероятность необнаружения ошибки в анализируемой последовательности.

Вероятность необнаружения ошибки:

$$P_H = Q_H / Q_{ВОЗМ},$$

где $Q_{ВОЗМ}$ – общее число ошибочных последовательностей; Q_H - количество необнаруживаемых ошибок.

Если обозначить n – число разрядов анализируемой последовательности, то 2^n – общее число возможных комбинаций. Число необнаруживаемых ошибок будет определяться как число комбинаций частных $Q_H = 2^{m-n} - 1$, где только одна комбинация истинна.

$$P_H = \frac{2^{n-m} - 1}{2^n - 1} \approx \frac{2^{n-m}}{2^n} = 2^{-m}.$$

Тогда вероятность обнаружения ошибки для m=16:

$$P_{обн} = 1 - P_H = 1 - \frac{1}{2^m} = 1 - \frac{1}{2^{16}} = 0{,}999984\ldots$$

Вывод. Сигнатурные анализаторы с разрядностью 16 и выше обладают очень высокой вероятностью обнаружения ошибки.

Генератор псевдослучайных последовательностей (ГПСП). В детерминированных методах тестирования для синтеза тестового набора необходима подробная информация о структуре устройства на уровне функциональной схемы, что представляет определенную проблему. Появившиеся методы компактного тестирования позволили добиться значительных преимуществ за счет отказа от анализа структуры схемы, рассматривая ее как черный ящик. Вместо перебора всех возможных неисправностей схемы произошел поворот к перебору состояний при помощи аппаратной генерации сверхдлинной тестовой последовательности.

В качестве источников тестовых воздействий можно использовать память полупроводниковую и память на магнитных носителях либо аппаратные генераторы, формирующие тесты рекуррентно в реальном масштабе времени. Примерами подобных генераторов могут служить генераторы случайных наборов, двоичные счетчики, генераторы псевдослучайных последовательностей (ГПСП).

ГПСП предназначены для аппаратного формирования тестовых последовательностей. Основу ГПСП составляют сдвигающие регистры с циклическим сложением по MOD2 некоторых обратных связей. Номера связей, участвующих в сложении по MOD2, определяются порождающим (образующим) полиномом. Так, например, полиному $j(x) = x^4 \oplus x \oplus 1$ будет соответствовать следующая структура ГПСП с внешним сумматором (рис. 4.5).

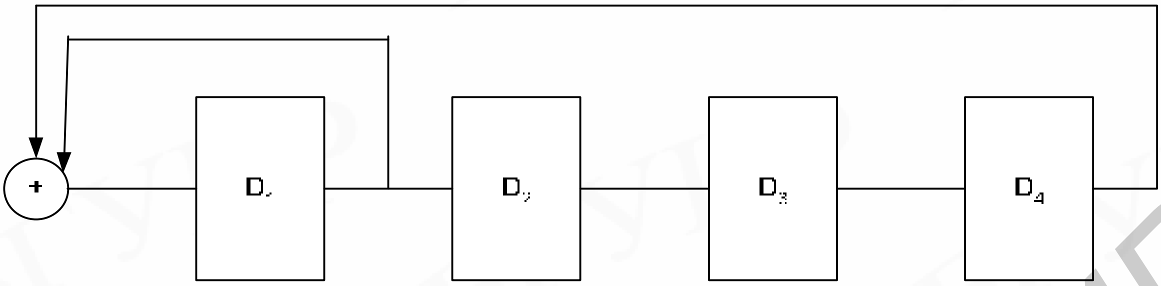


Рис. 4.5. Структура ГПСП для полинома $j(x) = x^4 \oplus x \oplus 1$

Если в качестве исходного состояния выбрать {1100} и подавать импульс сдвига, то получим последовательность.

1 1 0 0	1 0 1 1	0 1 1 0	0 0 1 0
1 1 1 0	0 1 0 1	0 0 1 1	0 0 0 1
1 1 1 1	1 0 1 0	1 0 0 1	1 0 0 0
0 1 1 1	1 1 0 1	0 1 0 0	1 1 0 0

Эта последовательность вся или частично может использоваться в качестве тестовой при условии, что тест должен начинаться с одной тестовой комбинации.

Если образующий полином удовлетворяет критерию примитивности и неприводимости, то длина последовательности $L=2^n-1$. Эта последовательность является *М*-последовательностью, т.е. последовательностью максимальной длины.

Задача синтеза генератора *М*-последовательности состоит в нахождении полинома $j(x)$, удовлетворяющего условиям формирования последовательности максимальной длины. С ростом m число полиномов $j(x)$ степени m, порождающих *М*-последовательность, быстро растет.

Например, для генератора ПСП (5 разрядов) возможны следующие полиномы (табл. 4.1).

Таблица 4.1

Возможные полиномы для ГПСП (5 разрядов)

№ пп.	**Примитивный полином**
1	$g(x) = 1 \oplus x^2 \oplus x^3$
2	$g(x) = 1 \oplus x^3 \oplus x^5$
3	$g(x) = 1 \oplus x \oplus x^2 \oplus x^3 \oplus x^5$
4	$g(x) = 1 \oplus x \oplus x^2 \oplus x^4 \oplus x^5$
5	$g(x) = 1 \oplus x \oplus x^3 \oplus x^4 \oplus x^5$
6	$g(x) = 1 \oplus x^2 \oplus x^3 \oplus x^4 \oplus x^5$

Среди множества полиномов $j(x)$, имеющих одинаковую старшую степень m, можно найти полиномы с наименьшим числом единичных коэффициентов a_i, $\overline{1,m}$. Обычно такие полиномы приводят в справочной литературе. Например, в табл. 4.2 приведены варианты полиномов, порождающих *M*-последовательность, с минимальным числом входов сумматора по MOD2.

Таблица 4.2

Примеры полиномов с наименьшим числом коэффициентов

m	g(x)
3	$1 \oplus x \oplus x^3$
4	$1 \oplus x \oplus x^4$
5	$1 \oplus x^2 \oplus x^5$
6	$1 \oplus x \oplus x^6$
7	$1 \oplus x \oplus x^7$
8	$1 \oplus x \oplus x^5 \oplus x^6 \oplus x^8$
9	$1 \oplus x^4 \oplus x^9$
10	$1 \oplus x^3 \oplus x^{10}$
11	$1 \oplus x^2 \oplus x^{11}$
12	$1 \oplus x^3 \oplus x^4 \oplus x^7 \oplus x^{12}$
13	$1 \oplus x \oplus x^3 \oplus x^4 \oplus x^{13}$
14	$1 \oplus x \oplus x^{11} \oplus x^{12} \oplus x^{14}$
15	$1 \oplus x \oplus x^{15}$
16	$1 \oplus x^2 \oplus x^3 \oplus x^5 \oplus x^{16}$

ЛИТЕРАТУРА

1. Яромолик В.Н. Контроль и диагностика цифровых узлов ЭВМ. Мн.: Наука и техника, 1988.

2. Каган В.М., Мкртумян И.В. Основы эксплуатации ЭВМ. М.: Энергоатомиздат, 1988.

3. Ярмолик В.Н., Калоша Е.П. и др. Проектирование самотестируемых СБИС: Науч. монография: В 2 т. Т. 1, 2. Мн.: БГУИР, 2001.

4. Садыхов Р.Х., Татур М.М. Технический сервис однородных вычислительных устройств. Мн.: Университетское, 2001.

Св. план 2005, поз. 54

Учебное издание

КРАТКИЙ КУРС ТЕХНИЧЕСКОЙ ДИАГНОСТИКИ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ

Методическое пособие по дисциплине
«Контроль и диагностика средств вычислительной техники»
для студентов специальности I-40 02 01
«Вычислительные машины, системы и сети»
вечерней и заочной форм обучения

Авторы-составители:
Лукашевич Марина Михайловна,
Татур Михаил Михайлович

Редактор Т.А. Лейко
Корректор Н.В. Гриневич

Подписано в печать 16.09.2005.	Формат 60×84 1/16.	Бумага офсетная.
Гарнитура «Таймс».	Печать ризографическая.	Усл. печ. л. 2,33.
Уч.-изд. л. 2,0.	Тираж 170 экз.	Заказ 166.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Лицензия на осуществление издательской деятельности №02330/0056964 от 01.04.2004.
Лицензия на осуществление полиграфической деятельности №02330/0131518 от 30.04.2004.
220013, Минск, П. Бровки, 6