

Б. М. Каган,
И. Б. Мкртумян

ОСНОВЫ ЭКСПЛУАТАЦИИ ЭВМ

Под редакцией Б. М. Кагана

Допущено Министерством высшего и среднего специального образования СССР в качестве учебного пособия для студентов вузов, обучающихся по специальности «Электронные вычислительные машины»



МОСКВА: ЭНЕРГОАТОМИЗДАТ · 1983

ББК 32.97
К12
УДК 681.3.004

Рецензенты: кафедра вычислительной техники
Московского лесотехнического института;
чл.-корр. АН СССР Г. П. Лопато.

Каган Б. М., Мкртумян И. Б.

К12 Основы эксплуатации ЭВМ: Учеб. пособие для вузов/Под ред. Б. М. Кагана. — М.: Энергоатомиздат, 1983. — 376 с., ил.

В пер.: 1 р. 10 к.

Рассмотрены основы теории эксплуатации, процессы и средства эксплуатационного обслуживания ЭВМ, их программного и информационного обеспечения. Изложены принципы построения и использования средств поддержки эксплуатационного обслуживания — систем автоматизации контроля правильности функционирования, диагностирования неисправностей, профилактических испытаний, защиты и восстановления баз данных и др.

Для студентов вузов и специалистов в области разработки и эксплуатации ЭВМ, АСУ.

К 2405000000-589 205-83
051(01)-83

ББК 32.97

6Ф7

© Энергоатомиздат, 1983

ПРЕДИСЛОВИЕ

Широкий масштаб использования ЭВМ в различных областях народного хозяйства, науки и культуры, большое число работающих в стране вычислительных машин делают исключительно важной проблему эффективного эксплуатационного обслуживания ЭВМ. Большая стоимость оборудования ЭВМ, их высокая производительность при обработке данных обуславливают значительные экономические потери при простое ЭВМ из-за неисправности или плохо организованной эксплуатации.

Электронная вычислительная машина является сложным объектом эксплуатационного обслуживания ввиду большой сложности ее логической организации и процессов функционирования, большого количества и разнообразия входящих в ее состав элементов и устройств, большой плотности монтажа электронного оборудования, сложного взаимодействия аппаратурных и программных средств, большого объема эксплуатационной документации. Это порождает определенные требования к квалификации обслуживающего персонала.

Повышение эффективности применения ЭВМ в народном хозяйстве и других областях в большой степени зависит от снижения трудовых затрат на обслуживание машин.

Решение проблемы эффективной эксплуатации ЭВМ требует наряду с соответствующей организацией их эксплуатационного обслуживания и широкой подготовкой специалистов для работы в этой области придания ЭВМ свойств обслуживаемости, т. е. приспособленности машины к процессам обслуживания. Последнее предполагает наличие специальных аппаратурно-программных средств поддержки эксплуатации.

В настоящее время разработка концепции эксплуатационного обслуживания машины и аппаратурно-программных средств поддержки эксплуатации стала неотъемлемой частью общего процесса проектирования ЭВМ. Такой подход

в большей степени реализован при создании ЕС ЭВМ II очереди.

С ростом парка ЭВМ растет число инженеров и математиков-программистов, профессиональная деятельность которых связана с эксплуатационным обслуживанием вычислительных машин. Многие выпускники высшей школы, обучающиеся по специальности «Электронные вычислительные машины», направляются на работу по эксплуатационному обслуживанию ЭВМ.

Специалисты, занятые разработкой вычислительной техники, и в особенности ее эксплуатацией, должны иметь хорошую подготовку по научным основам, методам и средствам эксплуатационного обслуживания ЭВМ.

Возникла необходимость включения в учебные планы соответствующих специальностей курса по основам эксплуатации ЭВМ. Ощущается нужда в книгах по научным основам, средствам и методам эксплуатационного обслуживания ЭВМ. Сказанное выше побудило авторов написать эту книгу.

В книге рассмотрены элементы теории эксплуатации, методы, средства и процессы эксплуатационного обслуживания ЭВМ и их программного и информационного обеспечения. Это соответствует современному представлению об ЭВМ как совокупности ее аппаратурных, программных и информационных средств.

В книге изложены элементы теории и аналитические модели надежности как аппаратурных, так и программных средств. Рассмотрены особенности эксплуатации операционных систем и баз данных.

Большое внимание уделяется проблеме обслуживаемости ЭВМ и аппаратурно-программным средствам поддержки эксплуатационного обслуживания.

В книге изложены принципы построения и методы использования аппаратурно-программных систем автоматического контроля правильности работы машины, автоматического диагностирования неисправностей, автоматического восстановления вычислительного процесса после сбоев, автоматизации профилактических испытаний, сбора и накопления информации о поведении машины в процессе эксплуатации.

Рассмотрены особенности диагностирования периферийных устройств, средств телеобработки данных, методы проектирования диагностических тестов. Излагаются также методы повышения надежности систем электропитания ЭВМ, сервисная и контрольно-измерительная аппаратура.

В заключительной части описаны основные процессы эксплуатационного обслуживания.

Широкое распространение микропроцессорной аппаратуры и микро-ЭВМ потребовало новых подходов и средств для их обслуживания и диагностирования. В книге излагается метод сигнатурного анализа, применяемый для проверки работоспособности этих устройств.

Предполагается, что читатель знаком с основами вычислительной техники, изложенными, например, в книге Б. М. Кагана «Электронные вычислительные машины и системы», выпущенной издательством «Энергия» в 1979 г.

Данная книга предназначена в качестве учебного пособия для студентов вузов, обучающихся по специальности «Электронные вычислительные машины», а также по смежным специальностям «Конструирование и производство электронно-вычислительной аппаратуры», «Автоматизированные системы управления», «Прикладная математика». Книга представляет интерес для специалистов, работающих в области разработки, применения и эксплуатации ЭВМ, АСУ и их программного обеспечения.

Главы 1, 4, 5 (кроме § 5.8), 16 (кроме § 16.7, 16.8) и § 7.4, 9.7, 14.4 написаны Б. М. Каганом, гл. 6, 7 (кроме § 7.4), 9 (кроме § 9.7), 10—13, 14 (кроме § 14.4), 15 и § 5.8, 16.7, 16.8 — И. Б. Мкртумяном. Гл. 8 написана совместно Б. М. Каганом и И. Б. Мкртумяном, гл. 2 — Б. М. Каганом и В. Н. Котляровским, гл. 3 — Б. М. Каганом и Г. А. Гришиным.

Авторы выражают благодарность рецензентам чл.-корр. АН СССР Г. П. Лопато, д-ру техн. наук, проф. Б. И. Зобову, кафедре «Вычислительная техника» МЛТИ, а также редактору книги доц. Ф. Б. Поволоцкому за ценные замечания и советы, которые способствовали улучшению содержания книги.

Авторы будут весьма признательны читателям, пожелавшим прислать замечания по книге, которые следует направлять по адресу 113114, Москва М-114, Шлюзовая наб., 10, Энергоатомиздат.

Глава первая

ОСНОВНЫЕ ПОНЯТИЯ ЭКСПЛУАТАЦИОННОГО ОБСЛУЖИВАНИЯ ЭВМ

1.1. ОСОБЕННОСТИ ЭВМ КАК ОБЪЕКТА ЭКСПЛУАТАЦИОННОГО ОБСЛУЖИВАНИЯ

Эксплуатация любого объекта (системы, машины), в общем случае состоит из его (ее) эксплуатационного использования, другими словами, использования по прямому назначению, и эксплуатационного обслуживания. Под *эксплуатационным обслуживанием* понимается совокупность операций, процедур и процессов, предназначенных для обеспечения работоспособности объекта (системы, машины).

Эксплуатируемый объект может находиться в работоспособном или неработоспособном состоянии. *Работоспособным* называется состояние объекта, при котором он способен выполнять заданные функции при сохранении значений его характеристик и параметров в пределах, установленных технической документацией. *Неработоспособным* называется состояние объекта, при котором он не в состоянии выполнить хотя бы одну из заданных функций ~~или~~ хотя бы один его параметр или одна характеристика не удовлетворяет требованиям, указанным в технической документации.

В зависимости от особенностей рассматриваемого объекта, в частности от степени автоматизации его основных технологических процессов, меняются соотношения между эксплуатационным использованием и эксплуатационным обслуживанием в общем процессе эксплуатации объекта. В соответствии с этим меняется содержание, вкладываемое в понятие эксплуатации того или иного объекта.

Например, на железнодорожном транспорте, где организация, планирование и управление железными дорогами, в работе которых участвуют тысячи рабочих и специалистов, ведется людьми (пусть даже с применением ЭВМ) и принятие управленческих решений во многих случаях не поддается формализации, под эксплуатацией железных дорог понимают организацию, планирование и управление

перевозочным процессом, т. е. эксплуатационное использование железных дорог.

Наоборот, в ЭВМ, где управление вычислительным процессом формализовано и осуществляется автоматически операционной системой, процедуры работы операторов машины достаточно просты, под эксплуатацией ЭВМ понимают в основном ее эксплуатационное обслуживание.

В дальнейшем в книге под термином «эксплуатация ЭВМ» будет пониматься главным образом ее эксплуатационное обслуживание.

Рассмотрим основные особенности ЭВМ как объекта эксплуатационного обслуживания.

ЭВМ — сложная техническая система. Современные ЭВМ средней и большой производительности следует отнести к сложным техническим объектам. Эта сложность определяется огромным количеством входящих в них элементов — в первую очередь электронных компонентов, большим числом связей между элементами, наличием в составе машины разнообразного по принципу действия и выполняемым функциям оборудования (в том числе прецизионного), в ряде случаев работающего вблизи предела физических возможностей (по скоростям передачи информации, по плотности записи информации на носителе и т. п.). Сложность ЭВМ обуславливается также сложностью выполняемых ею функций по обработке информации, которые еще недавно считались прерогативой человеческого мозга.

ЭВМ — совокупность аппаратных и программных средств. Сложность ЭВМ как объекта эксплуатации усугубляется тем, что современная ЭВМ — это неразделимый комплекс аппаратуры и программных средств. Программные средства в тесном взаимодействии с аппаратурой управляют вычислительным процессом, трансляцией программ с одного языка на другой, процессами взаимодействия ЭВМ с пользователями и операторами. Операционные системы современных ЭВМ без преувеличения можно считать наиболее «интеллектуальными» искусственными системами, когда либо созданными человеком. Эксплуатационное обслуживание ЭВМ должно охватывать в совокупности ее аппаратные и программные средства.

ЭВМ — универсальный преобразователь информации. В отличие от машин, осуществляющих преобразование видов энергии, состава или формы материала, ЭВМ имеет дело с преобразованиями информации. Это заставляет ввести для определения состояния ЭВМ наряду с понятием работоспособности — готовности выполнять предусмотренные

техническими условиями на машину преобразования данных — понятие *верности* или *достоверности* в функционировании ЭВМ, определяющей степень безошибочности работы машины.

Ошибки в работе ЭВМ могут не только искажать конечные результаты расчета, но и приводить к порче хранимой в ее памяти информации (данных и программ), которую следует рассматривать как составную часть вычислительной системы.

ЭВМ — человеко-машинная система. Даже в простейших режимах работы ЭВМ выступает как человеко-машинная система, в которой оператор определенным образом взаимодействует и общается с машиной (точнее, с ее операционной системой). Этот человеко-машинный характер вычислительной системы значительно усиливается при реализации систем коллективного пользования, разделения времени, телеобработки, диалоговых режимов.

Неправильные действия операторов и пользователей могут вызывать ненормальности (причем часто трудно обнаруживаемые и устранимые) в функционировании ЭВМ.

ЭВМ — объект, функционирующий в условиях действия случайных факторов. Случайными, заранее не предусмотренными являются моменты обращения к ЭВМ с запросами на выполнение тех или иных программ, необходимые для выполнения запросов ресурсы ЭВМ (время работы процессора, каналов, других устройств, требуемые объемы памяти), траектории прохождения запросов по вычислительной системе (используемое запросами оборудование) и, наконец, возможные нарушения нормального функционирования ЭВМ из-за возникновения различных неисправностей.

ЭВМ — объект со сложным описанием и очень большим объемом технической документации. Вычислительная машина как сложная система требует многоуровневого описания — на уровнях структур, функциональных схем, принципиальных схем, временных диаграмм, микропрограмм и т. д. Во многом эффективность эксплуатационного обслуживания зависит от структуры, полноты и качества документации, приспособленности ее для ведения эксплуатационного обслуживания.

Массовость объектов вычислительной техники и их рассредоточенность по территории страны

Эксплуатационное обслуживание сопровождает ЭВМ на протяжении всей ее «жизни» от момента изготовления до снятия с эксплуатации.

В общем случае можно выделить следующие виды обслуживания ЭВМ: хранение; установку, наладку на месте эксплуатации, ввод в эксплуатацию; обслуживание при нормальной работе;

планово-профилактические работы; устранение неисправностей (ремонт); обслуживание программного обеспечения; обслуживание информационных баз.

Вычислительные машины можно классифицировать по характеру обслуживания, что определяется их назначением и условиями эксплуатации. Вычислительная машина может быть *восстанавливаемым* или *невосстанавливаемым*, *обслуживаемым* или *необслуживаемым* объектом.

ЭВМ называется восстанавливаемым объектом, если во время эксплуатации может производиться ремонт по устранению возникающих неисправностей. В противном случае машина называется невосстанавливаемым объектом.

ЭВМ считается обслуживаемым объектом, если допускается периодическое проведение профилактических испытаний с целью выявления, замены или подрегулировки элементов и узлов, параметры которых близки к предельно допустимым. В противном случае ЭВМ является необслуживаемым объектом.

ЭВМ общего назначения представляет собой восстанавливаемые обслуживаемые объекты. К невосстанавливаемым объектам относятся, например, бортовые вычислительные машины ракет.

Сложность ЭВМ как объекта эксплуатационного обслуживания, вытекающие отсюда требования к высокому уровню квалификации обслуживающего персонала, большое число ЭВМ, находящихся в эксплуатации, рассредоточенность их по территории страны делают весьма острой проблему обеспечения эксплуатационного обслуживания ЭВМ.

Решение этой проблемы следует искать в соответствующей организации обслуживания ЭВМ, широкой подготовке специалистов по эксплуатационному обслуживанию и придании ЭВМ специального свойства высокой степени *обслуживаемости*, т. е. приспособленности машины к процессам обслуживания.

Степень обслуживаемости ЭВМ тем выше, чем меньше количество и ниже квалификация труда, затрачиваемого на эксплуатацию ЭВМ. Повышение степени обслуживаемости ЭВМ достигается с помощью специальных аппаратных и программных средств, автоматизирующих отдельные процессы обслуживания, в том числе средств автоматизации поиска неисправностей, автоматического контроля

правильности работы ЭВМ, автоматизации профилактических испытаний, автоматизации накопления и обработки информации о нарушениях нормального процесса работы при ее эксплуатации.

1.2. ОСНОВНЫЕ ЭКСПЛУАТАЦИОННЫЕ ХАРАКТЕРИСТИКИ ЭВМ

Пользователя в первую очередь интересуют параметры ЭВМ и особенности ее архитектуры, определяющие вычислительные возможности машины. К ним относятся производительность (скорость работы) процессора (число выполняемых операций в секунду), емкость памяти (число байт), состав периферийных устройств, а также следующие архитектурные характеристики: способ организации системы обмена информацией между ядром ЭВМ и периферийными устройствами, пропускная способность этой системы (число байт в секунду), формы представления данных в ЭВМ, способы адресации, особенности системы команд и др.

Важнейшими эксплуатационными характеристиками ЭВМ являются ее производительность Π и общий коэффициент эффективности машины

$$\mathcal{E} = \Pi / (C_{\text{ЭВМ}} + C_{\text{вс}}), \quad (1.1)$$

представляющий собой отношение ее производительности к сумме стоимости самой машины $C_{\text{ЭВМ}}$ и затрат на ее эксплуатацию за определенный период времени (например, период окупаемости капитальных затрат) $C_{\text{эк}}.$

Так как часто бывает трудно оценить затраты на эксплуатацию данной модели ЭВМ, то оценивают эффективность машины по упрощенной формуле:

$$\mathcal{E} = \Pi / C_{\text{ЭВМ}}. \quad (1.2)$$

Оценка и сопоставление производительности различных ЭВМ представляют собой сложную проблему, в сущности не получившую до сих пор удовлетворительного решения.

Интуитивно под производительностью ЭВМ понимают количество «вычислительной работы» или, другими словами, число задач, выполняемых машиной в единицу времени. Однако на производительность ЭВМ, оцениваемую по числу решенных в единицу времени задач, влияет слишком много факторов, в том числе тип задач, число тех или иных операций, выполняемых при решении задач, стиль программирования и другие особенности программ, логические возможности системы команд, структура процессора, характеристики и организация оперативной и внешней

памяти, особенности операционной системы, состав и характеристики периферийных устройств и др.

В этих условиях оценка производительности по некоторому набору задач применяется только тогда, когда вычислительная установка предназначается для решения достаточно узкого круга задач.

В настоящее время производительность ЭВМ общего назначения оценивается упрощенно по скорости выполнения некоторых «смесей» команд («смесей Гибсона»), формируемых путем анализа частоты исполнения разного вида команд при выполнении программ решения задач некоторого класса, например научно-технических задач или задач обработки данных. На основе такого анализа отдельным видам команд присваиваются определенные весовые коэффициенты.

Ниже приведены весовые коэффициенты отдельных видов команд для смеси команд, служащей для оценки производительности ЭВМ при решении научно-технических задач («смесь Гибсона 1»):

Вид команды (операции)	Весовой коэффициент K
Сложение (вычитание) с фиксированной запятой	33
Умножение с фиксированной запятой	0,6
Деление с фиксированной запятой	0,2
Сложение (вычитание) с плавающей запятой	7,3
Умножение с плавающей запятой	4,0
Деление с плавающей запятой	1,6
Логические операции	1,7
Безусловные передачи управления	17,5
Условные передачи управления	6,5

При использовании «смесей» команд производительность ЭВМ определяется по формуле

$$P = \frac{\sum_{s=1}^h k_s}{\sum_{s=1}^h k_s t_s}, \quad (1.3)$$

где k_s и t_s — соответственно весовой коэффициент и продолжительность выполнения s -й команды; h — число различных команд в «смеси».

Рассмотренные характеристики являются только номинальными показателями, они реализуются, если ЭВМ работоспособна. Суждение о реальных возможностях выполнения машиной определенных функций, решения определенного комплекса задач, особенно задач управления,

требующих обработки данных в реальном масштабе времени, может быть вынесено только с учетом характеристики эксплуатационной надежности машины.

Общие закономерности нарушений работоспособности различных объектов, закономерности процессов обеспечения и поддержания работоспособного состояния объектов изучает теория надежности.

Согласно современным представлениям под *надежностью* понимается свойство изделия (элемента, узла, устройства, машины, системы) выполнять заданные функции, сохраняя во времени свои характеристики в установленных пределах при определенных режимах и условиях использования, технического обслуживания, режимах хранения и транспортирования.

При оценке и анализе надежности какого-либо объекта приходится иметь дело со случайными событиями и случайными величинами, что заставляет использовать понятия и методы теории вероятностей.

Надежность вычислительной машины определяется безотказностью, достоверностью функционирования и характеристиками обслуживаемости, в первую очередь ремонтпригодностью (восстанавливаемостью после отказов), восстанавливаемостью информации после сбоев и проверкопригодностью.

Безотказность ЭВМ — свойство машины сохранять работоспособность в течение определенного промежутка времени при условии удовлетворения заданных ограничений на условия эксплуатации. Безотказность ЭВМ характеризуется закономерностями возникновения отказов. Под отказом понимается событие, заключающееся в утрате машиной работоспособности, т. е. в переходе в неработоспособное состояние. *Отказ ЭВМ* — это такое нарушение ее работоспособности, для устранения которого требуются определенные действия обслуживающего персонала по ремонту, замене и регулировке неисправного элемента, узла или устройства.

Отказы делятся на внезапные и постепенные.

Внезапный отказ — это практически мгновенное изменение какого-либо параметра (например, возрастание сопротивления цепи вследствие ее обрыва, электрический пробой диода и т. п.). *Постепенный отказ* — это происходящее во времени накопление количественных изменений некоторого параметра, достигнувшего значения, при котором машина оказывается в неработоспособном состоянии.

Внезапные отказы возникают, а постепенные отказы проявляются как устойчивые нарушения нормального функционирования машины в случайные моменты времени. Интервалы между отказами являются случайными величинами с некоторым законом распределения.

Безотказность ЭВМ может быть оценена *средним временем наработки машины на один отказ* ($\bar{T}_0$).

Последствия отказов характеризуются случайными величинами продолжительности перерыва в функционировании машины из-за ремонтных работ (случайными величинами продолжительности ремонта).

Ремонтпригодность ЭВМ — это степень приспособленности машины к предупреждению, обнаружению и устранению отказов. Ремонтпригодность определяет потерю работоспособности машины вследствие необходимости производить устранение неисправностей. Количество времени, затрачиваемого на ремонт для устранения отказа, является случайной величиной. Ремонтпригодность ЭВМ можно оценить *средним временем устранения неисправности, другими словами, средним значением времени восстановления работоспособности после отказа* ($\bar{T}_{в.о}$).

Работа ЭВМ заключается в выполнении преобразований информации.

В силу указанной специфики рабочего процесса надежность ЭВМ наряду с безотказностью определяется также *достоверностью функционирования*.

Достоверность функционирования ЭВМ — это свойство машины, определяющее безошибочность производимых машиной преобразований информации и характеризующееся закономерностями появления ошибок из-за сбоев.

Сбоем называют кратковременное самоустраниющееся нарушение нормального функционирования вследствие кратковременного воздействия на некоторый элемент (или элементы) внешних помех, а также вследствие кратковременного изменения параметров элементов (кратковременные нарушения контактов и т. п.). После сбоя машина длительное время может работать нормально. Сбой сопровождается искажением информации при операциях передачи, хранения и обработки. Следовательно, если не устранить последствия сбоя, то задача может оказаться неправильно решенной из-за искажений в данных, промежуточных результатах или в самой программе.

Однако если при отказе для восстановления работоспособности машины или системы необходимо устранить неисправность в аппаратуре, то при сбое необходимо вос-

становить только достоверность информации, что хотя и связано с потерями рабочего времени ЭВМ (например, на повторный пуск программы или ее части), но не требует ремонта или регулировки аппаратуры. В силу этого восстановление достоверности функционирования сравнительно легко может быть автоматизировано. Достоверность функционирования ЭВМ можно оценить средним временем наработки машины на один сбой $\bar{T}_c$.

Для оценки достоверности функционирования некоторых устройств (например, запоминающих устройств на магнитных лентах) удобно пользоваться средним количеством правильно переданной (считанной-записанной) информации (измеряемой в битах), приходящейся на одну ошибку.

Для более полной оценки достоверности функционирования введем в состав эксплуатационных характеристик ЭВМ восстанавливаемость информации, оцениваемую средним временем восстановления достоверности информации после сбоя ($\bar{T}_{в.с}$).

В процессе развития вычислительной техники повышение безотказности и достоверности функционирования ЭВМ достигалось путем использования более надежных элементов (переход от ламповых схем к полупроводниковым и затем к интегральным микросхемам), применения облегченных нагрузочных режимов для схемных компонентов, совершенствования конструкции и технологии (печатный, в том числе многослойный монтаж, плоские кабели, монтаж «накруткой», улучшение вентиляции блоков машин, согласование линий передачи сигналов, термоэлектротренировки элементов и др.), автоматизации проектирования, изготовления и контроля узлов и соответствующих решений в отношении структуры и организации вычислительного процесса ЭВМ.

Для современных ЭВМ среднее время наработки на отказ составляет от нескольких сотен часов для больших и средних ЭВМ общего назначения до нескольких тысяч часов для мини- и микро-ЭВМ. Среднее время восстановления после отказа составляет 0,5—1 ч, а среднее время восстановления информации после сбоя, если оно требует манипуляций со стороны обслуживающего персонала, 0,2—0,3 ч.

В большинстве случаев ЭВМ представляют собой восстанавливаемые и обслуживаемые объекты. Восстановление заключается в устранении неисправностей (отказов) и по-

следствий сбоев. Важной функцией обслуживания является проведение профилактических проверочных испытаний (профилактического контроля) для выявления элементов и узлов, параметры которых близки к предельно допустимым, и замены или подрегулировки последних. Кроме того, профилактические испытания предназначены для выявления неисправностей в аппаратуре, не охваченной средствами автоматического контроля правильности работы.

Целью профилактических испытаний является увеличение наработки на отказ в период между профилактическими испытаниями. Поэтому эффективность профилактических испытаний определяется отношением средних величин наработки на отказ при наличии профилактических испытаний ($\bar{T}_{0.пф}$) и без них ($\bar{T}_0$):

$$\mathcal{E}_{пф} = \frac{\bar{T}_{0.пф}}{\bar{T}_0} \quad (1.4)$$

Для оценки профилактических испытаний используют также коэффициент результативности профилактических испытаний, равный отношению числа отказов, выявленных при профилактических испытаниях ($n_{пф}$), к общему числу отказов, обнаруженных во время работы машины и профилактических испытаний ($n_{пф} + n_{раб}$):

$$k_{пф} = \frac{n_{пф}}{n_{пф} + n_{раб}} \quad (1.5)$$

Значения $\mathcal{E}_{пф}$ и $k_{пф}$ зависят от промежутка времени между профилактическими испытаниями и объема профилактических испытаний.

Профилактическое обслуживание сопряжено с потерями рабочего времени машины. Эти потери по своему характеру близки к потерям, связанным с устранением отказов.

Поэтому одной из важных эксплуатационных характеристик является *проверкопригодность ЭВМ*, которую можно оценить средней продолжительностью профилактических проверочных испытаний ($\bar{T}_{пф}$).

Введенные выше понятия ремонтпригодности, восстанавливаемости информации и проверкопригодности в совокупности определяют важный комплексный эксплуатационный показатель — *обслуживаемость ЭВМ*, или степень приспособляемости машины к процедурам ее эксплуатационного обслуживания.

Однако приведенные выше оценки этой приспособленности — величины $\bar{T}_{в.о.}$, $\bar{T}_{в.с.}$, $\bar{T}_{пф}$ — определяют приспособ-

денность к обслуживанию только с точки зрения связанных с обслуживанием потерь рабочего времени машины. Для более полной оценки обслуживаемости ЭВМ необходимо учитывать затраты труда обслуживающего персонала и требования к его квалификации.

Очевидно, чем выше надежность ЭВМ, тем больше может быть период между профилактическими работами, тем реже возникает потребность в восстановительных работах. Чем меньше тратится машинного времени и квалифицированного труда на профилактику и устранение возникших неисправностей, тем выше степень обслуживаемости ЭВМ.

Для более полной оценки свойств обслуживаемости ЭВМ введем следующие коэффициенты эффективности организации основных процессов обслуживания:

коэффициент эффективности организации процесса восстановления работоспособности после отказа

$$k_{\text{сф.в.о}}^{\text{орг}} = 1/\bar{T}_{\text{в.о}} \bar{c}_{\text{в.о}}; \quad (1.6)$$

коэффициент эффективности организации процесса восстановления достоверности информации после сбоев

$$k_{\text{сф.в.с}}^{\text{орг}} = 1/\bar{T}_{\text{в.с}} \bar{c}_{\text{в.с}}; \quad (1.7)$$

коэффициент эффективности организации профилактических испытаний

$$k_{\text{сф.пф}}^{\text{орг}} = 1/\bar{T}_{\text{пф}} \bar{c}_{\text{пф}}; \quad (1.8)$$

где $\bar{c}_{\text{в.о}}$, $\bar{c}_{\text{в.с}}$, $\bar{c}_{\text{пф}}$ — средние затраты в единицу времени на оплату труда персонала, привлекаемого соответственно для восстановления работоспособности после отказа, восстановления достоверности информации после сбоя, проведения профилактических испытаний.

Коэффициенты эффективности организации процессов обслуживания зависят от оснащенности этих процессов аппаратными и программными средствами поддержки, позволяющими сократить время выполнения соответствующих процедур и уменьшить трудовые затраты.

Для оценки совокупного влияния на продуктивность работы ЭВМ рассмотренных выше отдельных показателей надежности введем комплексный коэффициент эксплуатационной надежности (комплексный коэффициент использования)

$$k_n^k = \frac{\sum_{i=1}^n t_i - \sum_{r=1}^m \tau_{\text{в.ср}} - \tau_k - \sum_{s=1}^k \tau_{\text{пф}}}{\sum_{i=1}^n t_i + \sum_{j=1}^n \tau_{\text{в.о}}}; \quad (1.9)$$

где n , m , k — соответственно число отказов, сбоев и профилактических обслуживаний за рассматриваемый достаточно большой период наблюдения за эксплуатацией (без учета отказов и сбоев во время профилактических испытаний); t_i — интервал времени исправной работы между $(i-1)$ -м и i -м нарушениями функционирования машины из-за отказов; $\tau_{в.о.}$ — время восстановления после j -го отказа; $\tau_{в.ср}$ — время восстановления достоверности информации после r -го сбоя (время, потраченное на повторное выполнение программы, части программы, команды и т. д.); τ_k — суммарное машинное время, затраченное в рассматриваемый период на контроль достоверности информации (из-за двойного подсчета, контрольных вариантов, потерь машинного времени на процедуры автоматического контроля правильности работы ЭВМ и т. п.); $\tau_{пф}$ — время, затраченное на s -е профилактическое обслуживание.

В знаменателе (1.9) стоит полное время наблюдения за эксплуатацией ЭВМ. Оно равно сумме интервалов времени между отказами и общего времени, затраченного на их устранение. При этом принято, что при операциях контроля достоверности работы ЭВМ, восстановления достоверности информации и при профилактических проверках машина находится в потенциально работоспособном состоянии, хотя и не используется для полезной работы.

Выражение (1.9) определяет отношение полезного или, точнее, потенциально возможного полезного времени работы машины к общему времени наблюдения за эксплуатацией ЭВМ.

Если не учитывать сбои и их последствия и потери времени на контроль достоверности работы машины, можно получить приводимое в литературе выражение для коэффициента использования

$$\bar{k}_z = T_o / (\bar{T}_o + \bar{T}_{в.о.} + \bar{\tau}_{пф}), \quad (1.10)$$

где $\bar{\tau}_{пф}$ — среднее время профилактики, пересчитанное на один отказ. Приведенное выражение позволяет определить относительное время работоспособного состояния ЭВМ.

В технических условиях на ЭВМ обычно указывают коэффициент готовности ЭВМ

$$k_r = \bar{T}_o / (\bar{T}_o + \bar{T}_{в.о.}), \quad (1.11)$$

который определяет вероятность пребывания ЭВМ в работоспособном состоянии в любой произвольный момент времени (в период между профилактиками).

Между коэффициентами использования и готовности существует соотношение

$$k_{\text{и}} \leq k_{\text{г}}. \quad (1.12)$$

Помимо рассмотренных следует учитывать эксплуатационные характеристики ЭВМ, перечисленные ниже.

Долговечность ЭВМ — свойство машины при установленном для нее обслуживании сохранять указанные в технической документации характеристики в течение определенного времени хранения и эксплуатации. При этом *сохранностью машины* называют свойство ЭВМ сохранять исправное состояние при хранении в условиях, оговоренных технической документацией.

Эксплуатационные ресурсы — это ресурсы, необходимые для нормальной эксплуатации машины: площадь помещений, штат обслуживающего персонала, особые требования к параметрам окружающей среды в помещениях ЭВМ (температура, вентиляция, пылезащищенность и др.).

Глава вторая

ЭЛЕМЕНТЫ ТЕОРИИ ЭКСПЛУАТАЦИИ ЭВМ

2.1. НЕКОТОРЫЕ СВЕДЕНИЯ ИЗ ТЕОРИИ ВЕРОЯТНОСТЕЙ

Введенные в предыдущей главе характеристики далеко не полностью описывают процесс эксплуатации ЭВМ, носящий случайный характер. Так, если встает задача определения вероятности того, что в течение некоторого интервала времени t ЭВМ сохранит работоспособность, то для ее решения надо знать закон распределения случайной величины — времени безотказной работы ЭВМ. Процессы поиска неисправности, ремонта отказавшего устройства также являются случайными, поэтому в момент возникновения отказа можно говорить только о вероятности того, что работоспособность ЭВМ будет восстановлена к заданному сроку. Конечно, определенные организационные и технические мероприятия могут повлиять на время восстановления ЭВМ, однако оно по-прежнему остается случайным, непредсказуемым (можно говорить только о вероятности того, что восстановление будет закончено к такому-то сроку, а для определения этой вероятности необходимо знание закона распределения случайной величины — времени восстановления).

Законы распределения случайных величин (интервалов времени между отказами, времени восстановления и т. д.) дают исчерпывающую информацию о потоках отказов и восстановлений (в частности, зная их, можно определить и введенные в предыдущей главе характеристики).

Таким образом, отказы ЭВМ, завершения поиска неисправности, окончания ремонтных работ и т. д. — это случайные события, которые в течение некоторого интервала времени могут произойти или не произойти; предсказать заранее момент наступления случайного события невозможно.

Однако это не означает, что персонал, обслуживающий ЭВМ, не может определенным образом прогнозировать поведение ЭВМ в процессе эксплуатации и принимать своевременные меры для поддержания ее в работоспособном состоянии.

Существуют методы оценки (с помощью системы численных показателей) надежности ЭВМ, основанные на случайном (вероятностном) характере поведения вычислительной машины в процессе эксплуатации.

При этом создается математическая модель процесса эксплуатации ЭВМ как сложной системы, подверженной влиянию случайных факторов. Очевидно, что математическая модель процесса эксплуатации ЭВМ должна основываться на методах теории вероятностей. Так, законы распределения случайных величин (времени безотказной работы, поиска неисправности, ремонта и др.) задают в совокупности математическую модель процесса эксплуатации ЭВМ на языке теории вероятностей, а порознь — модели отдельных эксплуатационных явлений.

Поэтому в теории эксплуатации ЭВМ широко используются понятия и методы теории вероятностей. Рассмотрим кратко некоторые сведения из теории вероятностей.

Введем понятие *вероятности события* — численной меры объективной возможности случайного события, обозначаемой как $P(A)$, где A — случайное событие. Вероятность события равна 1, если оно всегда наступает в результате испытания (события, которые можно заранее предсказать); это событие называют достоверным. Вероятность события равна 0, если оно никогда не может произойти в результате проведения испытания (невозможное событие). Таким образом, для случайного события $0 < P(A) < 1$.

В результате одного испытания может наступить и не одно случайное событие. *Полной группой событий* называют несколько событий, одно из которых обязательно должно иметь место в процессе испытания.

Если несколько событий никогда не могут появиться одновременно в процессе испытания, то это *несовместные события*.

Вероятность события может быть оценена из рассмотрения возможных результатов испытания. Так, если имеется n различных исходов испытания, причем все они равновозможны, то вероятность $P(A)$ события A

$$P(A) = m/n, \quad (2.1)$$

где m — число исходов испытания, которые означают, что наступило событие A . Предположение о равновозможности исходов может вводиться, например, из соображения симметрии.

Другой способ определения вероятности события основан на том факте, что для широкого круга явлений частота появления определенного события в серии независимых испытаний, проведенных при однотипных условиях, подчиняется устойчивым закономерностям: отношение m/n , где n — число проведенных испытаний, а m — число появлений события A при достаточно больших n , во многих случаях сохраняет почти постоянное значение. При этом большие отклонения появляются тем реже, чем больше n . Величину m/n называют статистической вероятностью события A . Из сказанного следует, что статистическая вероятность при увеличении числа испытаний стремится к некоторому значению, которое и принимают равным вероятности события A .

Например, если в течение периода эксплуатации ЭВМ возникает несколько типов отказов (событий различного типа), то, подсчитав за достаточно длинный интервал времени общее число отказов n и число m_A отказов данного типа (событие A), можно определить вероятность события A как

$$P(A) = m_A/n. \quad (2.2)$$

Если события A и B несовместны, то тогда вероятность события, состоящего в появлении хотя бы одного из событий A или B (так называемой суммы событий A и B), определится выражением

$$P(A + B) = P(A) + P(B), \quad (2.3)$$

где $P(A)$, $P(B)$ — вероятности появления событий A и B соответственно.

Если события совместны, то

$$P(A + B) = P(A) + P(B) - P(AB), \quad (2.4)$$

где $P(AB)$ — вероятность одновременного появления событий A и B (произведения событий).

Если появление события A влияет на вероятность появления события B , то говорят, что данные события зависимы. Условной вероятностью $P(A/B)$ называют вероятность появления события A с учетом того, что событие B имело место, причем

$$P(AB) = P(A) P(B|A); \quad (2.5)$$

$$P(AB) = P(B) P(A/B). \quad (2.6)$$

Если события независимы, то

$$P(AB) = P(A)P(B). \quad (2.7)$$

Эти правила могут быть обобщены и на случай нескольких событий.

В процессе эксплуатации эффективность функционирования ЭВМ можно оценивать по числу отказов определенного типа, по времени, прошедшему от момента отказа до окончания ремонта, по времени от момента окончания ремонта до момента следующего отказа и т. д. Другими словами, для описания процесса эксплуатации можно использовать случайные величины различного рода: дискретные (такие как число отказов) и непрерывные (такие как длительность восстановления или время между отказами). Случайная величина описывается соотношениями между значениями случайной величины и вероятностями их появления. Эти соотношения задаются *законом распределения случайной величины*.

Так, задать закон распределения дискретной случайной величины X означает перечислить все ее возможные значения $x_1, x_2, \dots$ (их может быть конечное или счетное множество) и соответствующие им вероятности появления $P_1, P_2, \dots$:

$$P_i = P\{X = x_i\}, \quad i = 1, 2, \dots,$$

причем

$$\sum P_i = 1. \quad (2.8)$$

Описать аналогично непрерывную случайную величину невозможно. Можно говорить только о вероятности того, что значение непрерывной случайной величины X не будет больше x :

$$F(x) = P\{X < x\}, \quad (2.9)$$

где $F(x)$ — функция распределения вероятностей для случайной величины X .

Очевидно, что какой бы ни была случайная величина X , $F(-\infty) = 0$, $F(+\infty) = 1$.

В теории эксплуатации ЭВМ приходится иметь дело только со случайными величинами, определенными на интервале $[0, \infty)$; для них $F(0) = 0$, $F(\infty) = 1$.

Если случайная величина X может принимать только значения из конечного интервала $[a, b]$, то говорят, что X имеет усеченное распределение.

Для описания непрерывной случайной величины используется также плотность распределения вероятностей $f(x)$, представляющая собой производную от функции распределения $F(x)$:

$$f(x) = F'(x), \quad (2.10)$$

откуда

$$F(x) = \int_{-\infty}^x f(u) du. \quad (2.11)$$

График $f(x)$ называется кривой распределения вероятностей.

Вероятность попадания случайной величины X в интервал $[x, x + \Delta x]$ приближенно равна $f(x) \Delta x$.

Вероятность попадания X в интервал $[a, b]$ выражается формулой

$$P\{a \leq X \leq b\} = \int_a^b f(x) dx = F(b) - F(a), \quad (2.12)$$

причем так как $P(X=a)=0$, то

$$P\{a < X < b\} = F(b) - F(a). \quad (2.13)$$

Иногда для определения вероятностных характеристик процесса эксплуатации ЭВМ необходимо отыскивать закон распределения функции случайных величин, например закон распределения суммы двух независимых случайных величин X и Y . Если плотности распределения случайных величин X и Y — соответственно $f_1(x)$ и $f_2(y)$, то плотность распределения $g(z)$ случайной величины $Z=X+Y$ может быть определена по одной из формул [4]:

$$g(z) = \int_{-\infty}^{\infty} f_1(x) f_2(z-x) dx; \quad (2.14)$$

$$g(z) = \int_{-\infty}^{\infty} f_2(y) f_1(z-y) dy. \quad (2.15)$$

Все множество значений, которые может принять случайная величина из области определения, называется *генеральной совокупностью*.

Генеральная совокупность значений случайной величины кроме закона распределения характеризуется рядом числовых величин, таких как *момент (начальный и центральный) r -го порядка*.

Начальным моментом r -го порядка непрерывной случайной величины (а именно такими величинами являются времена между отказами и времена восстановления) называется величина, определяемая выражением [4]:

$$\alpha_r[X] = \int_{-\infty}^{\infty} x^r f(x) dx, \quad (2.16)$$

где $f(x)$ — плотность распределения вероятностей случайной величины x ; $\alpha_r[X]$ — r -й начальный момент.

С понятием центрального момента связано понятие *центрированной* случайной величины

$$X^* = X - \alpha_1[X], \quad (2.17)$$

где $\alpha_1[X]$ — математическое ожидание величины X , которое обычно обозначается $M[X]$.

Центрированная случайная величина X^* показывает отклонение X от ее математического ожидания.

Центральным моментом r -го порядка μ_r называется величина, определяемая формулой

$$\mu_r = \int_{-\infty}^{\infty} (x - M[X])^r f(x) dx. \quad (2.18)$$

Для описания генеральных совокупностей кроме математического ожидания используют центральные моменты 2—4-го порядков.

Центральный момент второго порядка, называемый *дисперсией случайной величины*, определяется формулой

$$D[X] = \int_{-\infty}^{\infty} f(x) (x - M[X])^2 dx. \quad (2.19)$$

Дисперсия характеризует степень отклонения значений случайной величины от ее математического ожидания. Величина

$$\sigma_X = \sqrt{D[X]}$$

называется *средним квадратическим отклонением*; эта величина используется наряду с $D[X]$. Безразмерный коэффициент вариации, определяемый как отношение среднего квадратического отклонения к математическому ожиданию, численно определяет степень отклонения случайной величины от математического ожидания.

Отношение центрального момента третьего порядка к кубу среднего квадратического отклонения называется *коэффициентом асимметрии*

$$\eta_3 = \mu_3 / \sigma^3, \quad (2.20)$$

который используется для описания формы кривой плотности распределения вероятностей.

Для той же цели используют и так называемый *коэффициент эксцесса*

$$\eta_4 = \mu_4 / \sigma^4. \quad (2.21)$$

Рассмотрим законы распределения случайных величин, наиболее широко используемые в теории эксплуатации ЭВМ.

Для описания дискретных случайных величин используются следующие законы распределения вероятностей.

1. Дискретное пуассоновское распределение с параметром λ :

$$P\{X = k\} = \frac{\lambda^k}{k!} e^{-\lambda}, \quad 0 \leq k < \infty, \quad (2.22)$$

где k — целое число.

Для пуассоновского распределения математическое ожидание, дисперсия и центральный момент третьего порядка равны λ , а коэффициент вариации равен $1/\sqrt{\lambda}$.

2. Геометрическое распределение с параметром p :

$$P\{X = k\} = (1 - p)^{k-1} p, \quad 1 \leq k < \infty, \quad (2.23)$$

где k — целое число; $0 < p < 1$.

Математическое ожидание случайной величины X , имеющей геометрическое распределение, равно $1/p$, дисперсия $(1-p)/p^2$, коэффициент вариации $\sqrt{1-p}$.

3. Если дискретная случайная величина является усеченной, принимающей значения, например, в интервале $0 < X < n$, то возможно применение биномиального распределения, задаваемого выражением

$$P\{X = k\} = C_n^k p^k (1-p)^{n-k}, \quad (2.24)$$

где k — целое число; $0 < p < 1$.

Математическое ожидание случайной величины, имеющей биномиальное распределение, равно np , дисперсия $np(1-p)$, коэффициент вариации $\sqrt{(1-p)/np}$.

4. Иногда используется и дискретное равномерное распределение для случайных величин, определенных на интервале $a < X < a+b-1$; в этом случае вероятность того, что случайная величина примет значение, равное k ($a < k < a+b-1$), равно $1/b$. Математическое ожидание такой случайной величины равно $a + (b-1)/2$, дисперсия $(b^2-1)/12$.

Для описания непрерывных случайных величин широко используются следующие законы распределения вероятностей.

1. Экспоненциальное распределение с плотностью

$$f(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad (2.25)$$

где λ — параметр распределения, связанный с математическим ожиданием выражением

$$\lambda = 1/M[X]. \quad (2.26)$$

Особенностью экспоненциального распределения является то, что коэффициент вариации этого распределения равен 1. Графики $f(x)$ и $F(x)$ для экспоненциального распределения приведены на рис. 2.1.

2. Распределение Эрланга k -го порядка с плотностью

$$f(x) = \frac{\lambda (\lambda x)^{k-1}}{(k-1)!} e^{-\lambda x}, \quad x > 0, \quad (2.27)$$

где $k > 1$ — целое число, а параметр λ связан с математическим ожиданием следующим выражением:

$$M[X] = k/\lambda. \quad (2.28)$$

Легко заметить, что при $k=1$ имеем экспоненциальное распределение.

Можно показать, что случайная величина, имеющая эрланговское распределение, может рассматриваться как сумма k случайных величин, имеющих экспоненциальное распределение с параметром λ . Другими словами, если в потоке событий длительность интервалов между моментами возникновения событий будет иметь эрланговское распределение, то такой поток событий можно рассматривать как поток, по-

лученный путем регулярного прореживания пуассоновского потока [удаления подряд следующих $(k-1)$ событий].

Эрланговское распределение времени между событиями при $k > 1$ характеризуется меньшей дисперсией, чем при экспоненциальном распределении. Его дисперсия

$$D[X] = k/\lambda^2. \quad (2.29)$$

Графики $f(x)$ и $F(x)$ для эрланговских распределений при различных k приведены на рис. 2.2, а, б.

3. Гамма-распределение с параметрами α и λ , определяемое плотностью

$$f(x) = \frac{\lambda (\lambda x)^{\alpha-1}}{\Gamma(\alpha)} e^{-\lambda x}, \quad x \geq 0, \quad (2.30)$$

где $\Gamma(\alpha) = \int_0^{\infty} e^{-u} u^{\alpha-1} du$ — гамма-функция; $\alpha > 0$ — параметр формы.

Математическое ожидание случайной величины T_0 , имеющей гамма-распределение, определяется формулой

$$M[X] = \alpha/\lambda, \quad (2.31)$$

а дисперсия

$$D[X] = 1/\lambda^2 \alpha. \quad (2.32)$$

Рис. 2.1. Плотность $f(x)$ и функция $F(x)$ для экспоненциального распределения ($\lambda=1$)

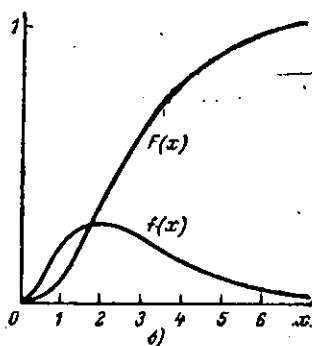
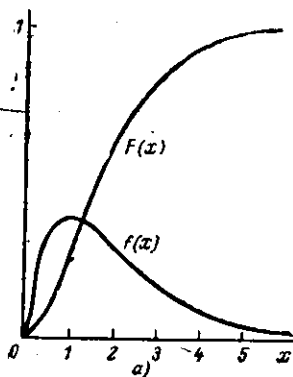
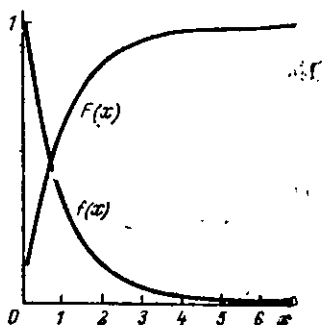


Рис. 2.2. Плотность $f(x)$ и функция $F(x)$ распределения Эрланга второго (а) и третьего (б) порядков ($\lambda=1$)

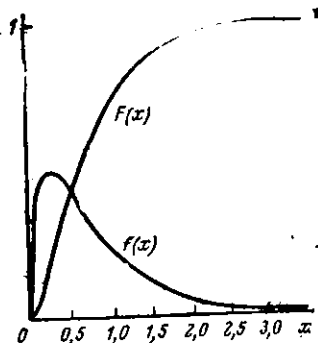


Рис. 2.3. Плотность $f(x)$ и функция $F(x)$ гамма-распределения ($\alpha=1,5$; $\lambda=1$)

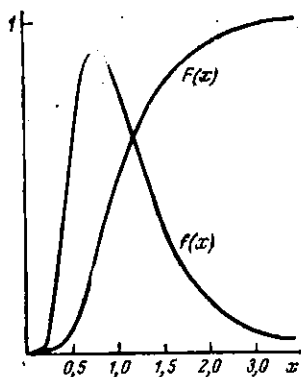


Рис. 2.4. Плотность $f(x)$ и функция $F(x)$ распределения Вейбулла ($\alpha=2$; $\lambda=1$)

При целых $\alpha > 1$ гамма-распределение совпадает со специальным распределением Эрланга. Графики $f(x)$ и $F(x)$ гамма-распределения приведены на рис. 2.3.

4. Распределение Вейбулла с параметрами α и λ , задаваемое плотностью

$$f(x) = \alpha \lambda (\lambda x)^{\alpha-1} \exp [-(\lambda x)^\alpha], \quad x > 0, \quad (2.33)$$

где $\alpha > 0$ — параметр формы распределения.

Математическое ожидание в этом случае определяется выражением

$$M[X] = \Gamma((\alpha + 1)/\alpha) / \lambda, \quad (2.34)$$

дисперсия

$$D[X] = \left\{ \Gamma\left(\frac{\alpha+2}{\alpha}\right) - \left[\Gamma\left(\frac{\alpha+1}{\alpha}\right) \right]^2 \right\} / \lambda^2. \quad (2.35)$$

Функция распределения для закона Вейбулла записывается в виде

$$F(x) = \exp [-(\lambda x)^\alpha], \quad (2.36)$$

откуда следует, что распределение Вейбулла может быть получено из экспоненциального распределения путем степенного преобразования масштаба времени. Графики плотности распределения $f(x)$ и функции распределения $F(x)$ для случая распределения Вейбулла приведены на рис. 2.4.

5. Нормальное распределение, определяемое плотностью

$$f(x) = \frac{1}{\sqrt{2\pi} \sigma} \exp \left\{ -\frac{(x-m)^2}{2\sigma^2} \right\}, \quad -\infty < x < \infty, \quad (2.37)$$

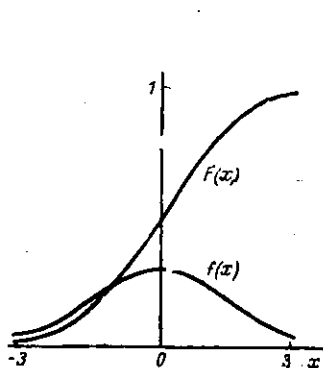


Рис. 2.5. Плотность $f(x)$ и функция $F(x)$ нормированного $N(0,1)$ нормального распределения

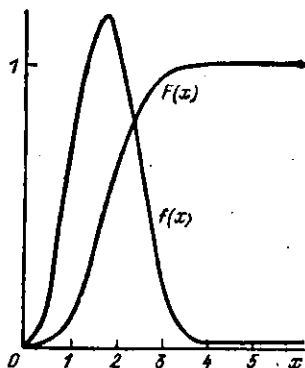


Рис. 2.6. Плотность $f(x)$ и функция $F(x)$ логарифмически-нормального распределения

где $m=M[X]$ — первый параметр распределения; σ — среднее квадратическое отклонение.

Графики $f(x)$ и $F(x)$ нормального распределения для $M[X]=0$, $\sigma=1$ [это распределение называется нормированным и обозначается $N(0, 1)$] приведены на рис. 2.5.

6. Логарифмически-нормальное распределение

$$f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\ln(x/m))^2}{2\sigma^2}\right\}, \quad x > 0, \quad (2.38)$$

где $m > 0$ — параметр формы, связанный с математическим ожиданием случайной величины x соотношением

$$M[X] = \ln m. \quad (2.39)$$

Дисперсия логарифмически-нормального распределения

$$D[X] = m^2 w / (w - 1), \quad (2.40)$$

где

$$w = \exp\{\sigma^2/2\}. \quad (2.41)$$

Графики $f(x)$ и $F(x)$ логарифмически-нормального распределения приведены на рис. 2.6.

Кроме перечисленных распределений могут использоваться и другие, в частности композиции из нескольких распределений (см., например, [4]).

Остановимся дополнительно на экспоненциальном распределении, играющем исключительно важную роль в теории эксплуатации ЭВМ,

так как это распределение связано с таким важным понятием, как пуассоновский поток событий.

Потоком событий называется последовательность событий, наступающих одно за другим в случайные моменты времени.

Пуассоновский поток событий — поток, для которого длительность интервалов между событиями является случайной величиной, имеющей экспоненциальное распределение, задаваемое выражением (2.25), в котором параметр λ интерпретируется как интенсивность (скорость) потока событий.

Важное свойство пуассоновского потока событий заключается в том, что вероятность появления нового события в течение интервала времени Δt определяется только длиной интервала Δt и интенсивностью потока событий λ , но не зависит от предыстории, т. е. от того, сколько и когда произошло событий в прошлом. Таким образом, для пуассоновского потока характерно отсутствие последствия.

Промежутки времени между моментами наступления событий в пуассоновском потоке — независимые случайные величины, а вероятность $p_k(t)$ того, что за время $[0, t]$ наступит точно k событий, определяется выражением

$$p_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad (2.42)$$

Другими словами, дискретная случайная величина — число наступивших событий, например число отказов ЭВМ в течение интервала времени t , — определяется дискретным распределением Пуассона с параметром λt (математическим ожиданием числа отказов, равным λt). Данное свойство характеризует пуассоновский поток как стационарный.

Отметим следующие свойства пуассоновского потока.

I. Пуассоновский поток ординарен — вероятность двух событий и более в момент времени t практически равна нулю.

II. При случайном прореживании пуассоновского потока с интенсивностью λ , когда с вероятностью α наступившее событие оставляется, а с вероятностью $1 - \alpha$ отбрасывается, получается пуассоновский поток с интенсивностью $\alpha\lambda$.

III. При наложении k независимых пуассоновских потоков с интенсивностями λ_i , $i = \overline{1, k}$ образуется пуассоновский поток с суммарной интенсивностью $\lambda = \sum_{i=1}^k \lambda_i$.

IV. Остаток времени от случайно взятой точки до ближайшего события в пуассоновском потоке с интенсивностью λ есть случайная величина, имеющая экспоненциальное распределение с тем же параметром λ .

2.2. МОДЕЛИ ПОТОКОВ ОТКАЗОВ И СБОЕВ

Под аналитической моделью некоторого процесса понимают совокупность математических зависимостей, описывающих его протекание с подробностью и точностью, соответствующей решаемой задаче исследования процесса. Поведение ЭВМ при эксплуатации зависит от ряда случайных факторов, таких как возникновение отказов, сбоев, восстановление работоспособности машины.

В § 2.2—2.4 рассматриваются модели отдельных процессов, влияющих на эксплуатационные характеристики ЭВМ.

Рассмотрим основные характеристики потока отказов, базирующиеся на законе распределения интервала времени между отказами.

Основной характеристикой является вероятность безотказной работы ЭВМ $P(t)$ — вероятность того, что ЭВМ сохранит работоспособность до момента времени t . Если $F(t)$ — функция распределения времени безотказной работы ЭВМ T_0 , то вероятность безотказной работы $P(t)$ — это функция, дополнительная к $F(t)$:

$$P(t) = P\{T_0 \geq t\} = 1 - F(t), \quad (2.43)$$

причем $P(0) = 1$; $P(\infty) = 0$, т. е. в момент начала эксплуатации с вероятностью 1 ЭВМ исправна, при завершении срока эксплуатации ЭВМ практически непригодна к использованию.

Функцию $F(t)$ в литературе по надежности часто называют вероятностью отказа, а также функцией ненадежности или функцией риска и обозначают $Q(t)$:

$$Q(t) = 1 - P(t) = F(t). \quad (2.44)$$

Используя функцию распределения $F(t)$, можно определить такой параметр, как среднее время безотказной работы:

$$\bar{T}_0 = M[T_0] = \int_0^{\infty} t f(t) dt, \quad (2.45)$$

где $f(t) = dF(t)/dt$ — плотность распределения случайной величины T_0 .

С помощью $P(t)$ можно оценить также и среднее число элементов (устройств) ЭВМ $n(t)$, которые откажут за интервал времени Δt :

$$n(t) = N_0 P(t) - N_0 P(t + \Delta t), \quad (2.46)$$

где N_0 — число исправных элементов (устройств) ЭВМ в начале ее эксплуатации.

Введем понятие интенсивности отказов $\lambda(t)$ как условную плотность вероятности времени до возникновения отказа при условии, что до момента времени t отказа не было.

Таким образом, вероятность того, что первый отказ произойдет в интервале времени $t < T < t + \Delta t$,

$$P\{t < T < t + \Delta t | T > t\} = [P(t) - P(t + \Delta t)]/P(t) \approx \lambda(t) \Delta t. \quad (2.47)$$

Аналогично

$$[F(t + \Delta t) - F(t)]/P(t) \approx \lambda(t) \Delta t, \quad (2.48)$$

откуда

$$\lambda(t) = - \left(\frac{dP}{dt} \right) / P(t), \quad (2.49)$$

или

$$\lambda(t) = \left(\frac{dF(t)}{dt} \right) / P(t). \quad (2.50)$$

Из (2.49) получаем

$$P(t) = \exp \left\{ - \int_0^t \lambda(\tau) d\tau \right\} + C. \quad (2.51)$$

Постоянную C можно найти, воспользовавшись начальными условиями при $t=0$; $P(0)=1$, откуда $C=0$.

Таким образом, вероятность безотказной работы связана с интенсивностью отказов $\lambda(t)$ выражением

$$P(t) = \exp \left\{ - \int_0^t \lambda(\tau) d\tau \right\}, \quad (2.52)$$

или

$$- \ln P(t) = \int_0^t \lambda(\tau) d\tau. \quad (2.53)$$

Если интенсивность отказов $\lambda = \text{const}$, то можно показать, что

$$\bar{T}_0 = M[T_0] = 1/\lambda. \quad (2.54)$$

Другими словами, λ — среднее число отказов в единицу времени.

Приведенные результаты еще раз подчеркивают важность экспоненциального распределения в теории эксплуа-

тации ЭВМ, так как для основных режимов функционирования ЭВМ (исключая период первичной приработки и периоды физического износа узлов и элементов) предположение $\lambda(t) = \text{const}$ практически всегда выполняется.

ЭВМ состоит из большого числа подсистем — отдельных устройств, таких как процессоры, каналы, внешние устройства и т. д. В свою очередь каждая из этих подсистем (устройств) ЭВМ состоит из отдельных логических элементов (интегральных микросхем), расположенных на различных типовых элементах замены (ТЭЗ).

Число ТЭЗ в устройствах, как и число логических элементов на разных ТЭЗ, различно. Имеются различия в режимах работы ТЭЗ и устройств в целом. Это обуславливает неравномерность выхода из строя устройств ЭВМ: каждое устройство характеризуется своей интенсивностью отказов λ_i , $i = 1, N$, где N — число типов устройств ЭВМ.

Если время безотказной работы i -го устройства ЭВМ есть случайная величина, имеющая экспоненциальное распределение с параметром λ_i , то вероятность его безотказной работы за время t

$$P_i(t) = 1 - \int_0^t \lambda_i e^{-\lambda_i \tau} d\tau = e^{-\lambda_i t}. \quad (2.55)$$

Будем считать, что события — отказы устройств различного типа — независимы между собой, тогда вероятность безотказной работы ЭВМ

$$P(t) = \prod_{i=1}^N P_i(t), \quad (2.56)$$

где $P_i(t)$ — вероятность отказа устройства i -го типа за время t ; N — число устройств в составе ЭВМ.

На основании (2.55) получаем

$$P(t) = \prod_{i=1}^N P_i(t) = \prod_{i=1}^N e^{-\lambda_i t} = \exp \left\{ -t \sum_{i=1}^N \lambda_i \right\}. \quad (2.57)$$

Нетрудно заметить, что формула (2.57) отражает одно из свойств пуассоновского потока: при наложении нескольких пуассоновских потоков вновь имеем пуассоновский поток с интенсивностью, равной сумме интенсивностей отдельных потоков.

Рассмотрим, как можно вычислить вероятность безотказной работы, если известна функция распределения времени наработки на отказ отдельных устройств ЭВМ. Очевидно, что если ЭВМ содержит N устройств, то для нере-

зерирированной ЭВМ выход одного устройства из строя означает отказ ЭВМ в целом. Другими словами, если через T_{oi} обозначить случайную реализацию времени наработки на отказ i -го устройства, то случайная реализация времени наработки на отказ ЭВМ в целом

$$T_o = \min_{i=\overline{1,N}} T_{oi}. \quad (2.58)$$

Так как $T_o \geq t$ только в том случае, если $T_{oi} \geq t$, $i = \overline{1, N}$, то вероятность безотказной работы ЭВМ

$$P\{T_o \geq t\} = P\{T_{oi} \geq t, i = \overline{1, N}\} = \prod_{i=1}^N (1 - F_{oi}(t)). \quad (2.59)$$

Для упрощения можно считать, что надежность всех устройств ЭВМ одинакова; более того, так как время наработки на отказ одного устройства значительно больше времени наработки на отказ ЭВМ в целом, в первом приближении можно считать, что при малых t $F_{oi}(t) \approx \lambda t^\alpha$ ($\alpha > 0$). Тогда при достаточно больших N и одинаковых $F_{oi}(t)$

$$P\{T_o \geq t\} = (1 - F_{oi}(t))^N \approx \exp(-NF_{oi}(t)) = \exp(-N\lambda t^\alpha). \quad (2.60)$$

Введем в рассмотрение случайную величину $T_o^* = T_o/k_N$, где k_N — нормирующая константа, выбранная таким образом, чтобы T_o^* имело предельное распределение при $N \rightarrow \infty$ [54].

Тогда

$$P\{T_o^* \geq t\} = P\{T_o \geq k_N t\} \approx \exp\{-N\lambda k_N^\alpha t^\alpha\}. \quad (2.61)$$

Если положить $k_N = (N\lambda)^{-1/\alpha}$, то будем иметь

$$P\{T_o^* \geq t\} \approx \exp\{-t^\alpha\}. \quad (2.62)$$

Таким образом, нормированная величина $(N\lambda)^{1/\alpha} T_o$ в качестве предельного имеет распределение Вейбулла.

Полученные выражения позволяют оценить вероятность безотказной работы ЭВМ в течение заданного интервала времени.

Выше были рассмотрены простейшие модели потока отказов ЭВМ в предположении, что интервалы между моментами появления отказов являются одинаково распределенными случайными величинами.

Модели потоков сбоев

Сбои — кратковременные и самоустраниющиеся нарушения нормальной работы ЭВМ, так же как и устойчивые отказы, появляются в случайные моменты времени.

Подобно случайной величине — интервалу между моментами появления отказов T_0 — можно говорить о том, что интервал времени между сбоями T_c есть случайная величина, задаваемая функцией распределения $F_c(t) = P\{T_c \leq t\}$ или плотностью распределения $f_c(t) = dF_c(t)/dt$. В этом случае среднее время наработки на сбой будет определяться как математическое ожидание случайной величины T_c , заданной функцией распределения $F_c(t)$.

Появление сбоев в процессе выполнения программ обуславливает увеличение общего времени решения, так как при обнаружении сбоя схемами контроля ЭВМ некоторое время тратится на восстановление правильности информации и повторение операции (части или всей программы в целом).

В целом простейшие модели потоков сбоев аналогичны моделям потоков отказов; различие состоит только в законах распределения случайной величины — интервала между моментами появления сбоя (в частности, различные математическое ожидание и дисперсия случайных величин — интервалов между отказами и сбоями).

2.3. МОДЕЛИ ПОТОКОВ ВОССТАНОВЛЕНИЯ

Большинство ЭВМ, как отмечалось в гл. 1, является восстанавливаемыми системами. Время восстановления $T_{в.о}$ работоспособного состояния ЭВМ после наступления устойчивого отказа — величина случайная, характеризующаяся функцией распределения

$$F_{в.о}(t) = P\{T_{в.о} \leq t\} \quad (2.63)$$

или плотностью распределения

$$f_{в.о}(t) = dF_{в.о}(t)/dt. \quad (2.64)$$

По аналогии с интенсивностью отказов $\lambda(t)$ вводится интенсивность восстановления $\mu_{в.о}(t)$ — условная плотность вероятности времени до восстановления ЭВМ при условии, что до момента времени t работоспособность ЭВМ восстановлена не была.

В ряде случаев время восстановления (суммарное время работы обслуживающего персонала по поиску неисправности, замене отказавшего элемента и проверке работоспособности с помощью специальных тестов) можно считать случайной величиной, имеющей экспоненциальное распределение с постоянной интенсивностью $\mu_{в.о}(t) = \text{const}$, тогда

$$f_{в.о}(t) = \mu_{в.о} e^{-\mu_{в.о} t}. \quad (2.65)$$

В этом случае среднее время восстановления

$$\bar{T}_{в.о} = M[T_{в.о}] = \int_0^{\infty} t \mu_{в.о} e^{-\mu_{в.о} t} dt = \frac{1}{\mu_{в.о}}, \quad (2.66)$$

т. е. интенсивность восстановления в этом случае — величина, обратная математическому ожиданию времени восстановления.

При появлении отказа в ЭВМ невозможна мгновенная замена отказавшего элемента, так как время поиска и замены ТЭЗ (не говоря уже о замене логического элемента в ТЭЗ) случайно и отлично от нуля.

В самом простейшем случае, когда численность обслуживающего персонала такова, что поиск неисправности и ее устранение начинается практически сразу после ее возникновения, основной характеристикой восстановления ЭВМ является вероятность того, что работоспособность ЭВМ будет восстановлена за заданное время t ; эта вероятность определяется по (2.63).

Случайная величина $T_{в.о}$ представляет собой, как отмечалось ранее, сумму двух случайных величин: времени поиска причины отказа t_n и времени его устранения (замены ТЭЗ) с учетом времени проверки работоспособности ЭВМ после ремонта t_y :

$$T_{в.о} = t_n + t_y. \quad (2.67)$$

Функция плотности распределения $T_{в.о}$ может быть найдена по формуле композиции законов распределения (2.14):

$$f_{в.о}(t) = \int_0^t f_n(u) f_y(t-u) du, \quad (2.68)$$

где $f_n(t)$, $f_y(t)$ — плотности распределения времен поиска и устранения отказа.

Если предположить t_n и t_y — случайные величины, имеющие экспоненциальные распределения с одним и тем же параметром $\mu_{в.о}$, то будем иметь

$$f_{в.о}(t) = \int_0^t \mu_{в.о}^2 e^{-\mu_{в.о} \tau} e^{-\mu_{в.о} (t-\tau)} d\tau = \mu_{в.о}^2 t e^{-\mu_{в.о} t}. \quad (2.69)$$

Как следует из (2.69), время восстановления имеет эрланговское распределение второго порядка. Данный факт имеет достаточно простую физическую интерпретацию: в силу сложности устройств ЭВМ вероятность восстано-

ния ЭВМ за очень малое время практически равна нулю. Практически равна нулю и вероятность длительной работы по восстановлению, так как современные ЭВМ снабжаются развитой системой диагностики и укомплектованы ЗИП.

Функция распределения времени восстановления при $f_{в.о}(t)$, определенной (2.69), имеет вид

$$F(t) = \int_0^t f_{в.о}(\tau) d\tau = 1 - \left(1 + \frac{2t}{\bar{T}_{в.о}}\right) e^{-2t/\bar{T}_{в.о}}, \quad (2.70)$$

причем $\mu_{в.о} = 2/\bar{T}_{в.о}$ — интенсивность восстановления; $\bar{T}_{в.о}$ — среднее время восстановления.

Рассмотрим, каким образом может быть определен коэффициент готовности ЭВМ с учетом интенсивностей потока отказов и восстановлений. Сделаем предположение о том, что длительность интервалов работы ЭВМ между моментами появления отказов есть случайные величины, имеющие экспоненциальное распределение с параметром λ_0 , а длительность восстановления — экспоненциально распределенная случайная величина с параметром $\mu_{в.о}$. Будем считать, что ремонт (устранение отказа) начинается сразу после наступления отказа (это можно интерпретировать так: время поиска включено во время ремонта). После окончания ремонта ЭВМ сразу приступает к выполнению заданий.

Таким образом, ЭВМ может находиться в двух состояниях: работоспособном и неработоспособном, когда проводится восстановление. Обозначим эти состояния соответственно символами e_0 и e_1 .

Пусть вероятности нахождения ЭВМ в состояниях e_0 и e_1 обозначены как $P_0(t)$ и $P_1(t)$. Если ЭВМ исправна в момент времени $t=0$, то $P_0(0)=1$, $P_1(0)=0$.

Для определения аналитической зависимости $P_0(t)$, $P_1(t)$ от времени вычислим вероятность того, что ЭВМ будет находиться в состоянии e_0 в момент времени $t+\Delta t$, где Δt — малый интервал времени.

Если в момент времени t ЭВМ находилась в состоянии e_0 , то она в течение времени Δt останется в этом состоянии с вероятностью $e^{-\lambda_0 \Delta t} P_0(t)$. Если в момент времени t ЭВМ находилась в состоянии e_1 , то за интервал времени Δt она перейдет в состояние e_0 с вероятностью $(1 - e^{-\mu_{в.о} \Delta t}) P_1(t)$. Таким образом, можно записать

$$P_0(t + \Delta t) = e^{-\lambda_0 \Delta t} P_0(t) + (1 - e^{-\mu_{в.о} \Delta t}) P_1(t).$$

С учетом того, что при малых Δt $e^{-\lambda_0 \Delta t} \approx (1 - \lambda_0 \Delta t)$, $e^{-\mu_{в.о} \Delta t} \approx (1 - \mu_{в.о} \Delta t)$, имеем

$$P_0(t + \Delta t) = P_0(t) - \lambda_0 \Delta t P_0(t) + \mu_{в.о} P_1(t) \Delta t. \quad (2.71)$$

Преобразовав (2.71), получим

$$[P_0(t + \Delta t) - P_0(t)]/\Delta t = -\lambda_0 P_0(t) + \mu_{в.о} P_1(t), \quad (2.72)$$

что при $\Delta t \rightarrow 0$ дает следующее дифференциальное уравнение:

$$dP_0(t)/dt = -\lambda_0 P_0(t) + \mu_{в.о} P_1(t). \quad (2.73)$$

Так как для любого момента времени t выполняется равенство

$$P_0(t) + P_1(t) = 1, \quad (2.74)$$

уравнение (2.73) может быть переписано в виде

$$dP_0(t)/dt = -(\lambda_0 + \mu_{в.о}) P_0(t) + \mu_{в.о}. \quad (2.75)$$

Уравнение (2.75) описывает процесс эксплуатации ЭВМ, на функционирование которой оказывают влияние потоки отказов и восстановлений. Его аналитическое решение имеет следующий вид:

$$P_0(t) = \frac{\mu_{в.о}}{\lambda_0 + \mu_{в.о}} [1 - e^{-(\lambda_0 + \mu_{в.о})t}] + P_0(0) e^{-(\lambda_0 + \mu_{в.о})t}, \quad (2.76)$$

где $P_0(0) = 1$.

Нетрудно заметить, что при $t \rightarrow \infty$ полученное выражение стремится к пределу, равному π_0 — стационарной (установившейся) вероятности нахождения ЭВМ в состоянии e_0 :

$$\pi_0 = \lim_{t \rightarrow \infty} P_0(t) = \mu_{в.о}/(\lambda_0 + \mu_{в.о}). \quad (2.77)$$

Используя (2.74), получаем стационарную вероятность

$$\pi_1 = \lim_{t \rightarrow \infty} P_1(t) = \lambda_0/(\lambda_0 + \mu_{в.о}). \quad (2.78)$$

Стационарная вероятность π_0 есть не что иное, как доля времени работоспособного состояния машины или, другими словами, коэффициент готовности. Действительно,

$$\pi_0 = \frac{\mu_{в.о}}{\lambda_0 + \mu_{в.о}} = \frac{1/\bar{T}_{в.о}}{1/\bar{T}_0 + 1/\bar{T}_{в.о}} = \frac{\bar{T}_0}{\bar{T}_{в.о} + \bar{T}_0} = k_r. \quad (2.79)$$

Восстановление после сбоев

Время восстановления после сбоя $T_{в.с}$ складывается из времени идентификации ошибки как сбоя, времени фиксации сбоя (с целью накопления статистики о сбоях) и вре-

мени автоматического рестарта. Время восстановления после сбоя можно считать случайной величиной, задаваемой функцией распределения $F_{в.с}(t) = P\{T_{в.с} \leq t\}$ [плотностью распределения $f_{в.с}(t) = dF_{в.с}(t)/dt$]. Среднее время восстановления после сбоя в этом случае оценивается как математическое ожидание случайной величины $T_{в.с}$, заданной функцией распределения $F_{в.с}(t)$.

2.4. МОДЕЛЬ ПРОФИЛАКТИЧЕСКИХ ИСПЫТАНИЙ.

ВЛИЯНИЕ ПЕРИОДИЧНОСТИ ПРОФИЛАКТИЧЕСКИХ ИСПЫТАНИЙ НА КОЭФФИЦИЕНТ ГОТОВНОСТИ ЭВМ С ЧАСТИЧНЫМ КОНТРОЛЕМ

Профилактические испытания наряду с выявлением и заменой элементов с параметрами, близкими к предельным, имеют целью также выявление отказов в оборудовании, не охваченном контролем или не участвующем в вычислительном процессе.

Возможность необнаруженных отказов всегда имеется в ЭВМ, так как невозможно охватить контролем всю аппаратуру. Это приводит к ухудшению достоверности функционирования машины.

Достоверность работы ЭВМ с частичным контролем может быть повышена путем периодических профилактических испытаний не охваченной контролем аппаратуры ЭВМ.

Вероятность наличия необнаруженных отказов в не охваченном контролем оборудовании зависит от того, как давно по отношению к текущему моменту проводились профилактические испытания:

$$P(\Delta T) = 1 - e^{-\lambda_n \Delta T}, \quad (2.80)$$

где ΔT — время, прошедшее от последней полной проверки до начала текущего момента ($0 \leq \Delta T \leq T_{пф}$); $T_{пф}$ — период между профилактическими испытаниями («период профилактических испытаний»), причем $\lambda_{пф} = 1/T_{пф}$ — частота профилактических испытаний; λ_n — интенсивность потока отказов в неохваченной контролем части оборудования.

Таким образом, вероятность нахождения ЭВМ в работоспособном состоянии в любой момент времени зависит не только от характеристики контроля, но и от периодичности профилактических испытаний. К началу следующих профилактических испытаний вероятность наличия необнаруженных отказов в неконтролируемой части аппаратуры

$$P(T_{пф}) = 1 - e^{-\lambda_n T_{пф}}. \quad (2.81)$$

Для правильного выбора периода профилактических испытаний необходимо проанализировать его влияние на коэффициент готовности ЭВМ.

Рассмотрим поведение ЭВМ с частичным контролем при отсутствии и наличии периодических профилактических испытаний.

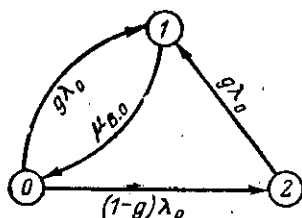


Рис. 2.7. Граф состояний ЭВМ с частичным контролем при отсутствии периодических профилактических испытаний:

0 — работоспособное состояние; 1 — состояние обнаруженного отказа; 2 — состояние необнаруженного отказа; λ_0 — интенсивность потока отказов ЭВМ; $\mu_{в.о.}$ — интенсивность потока восстановления после отказа ЭВМ; g — доля аппаратуры, охваченной контролем

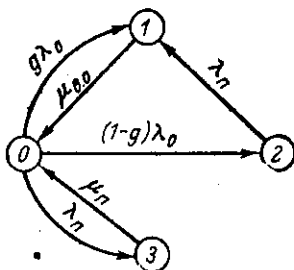


Рис. 2.8. Граф состояний ЭВМ с частичным контролем при наличии периодических профилактических испытаний:

0 — 2 — то же, что на рис. 2.7; 3 — состояние выполнения профилактических тестов (профилактических испытаний); $\lambda_n = 1/T_n$, где T_n — период профилактических испытаний; $\mu_n = 1/\tau_n$, где τ_n — длительность профилактических испытаний

Примем, что интервалы времени между моментами возникновения отказов в неконтролируемой части аппаратуры и времена, расходуемые на устранение этих отказов, — случайные величины, распределенные по экспоненциальному закону, причем интенсивности потоков отказов и восстановлений равны соответственно λ_0 и $\mu_{в.о.}$ При этих предположениях рассматриваемый случайный процесс смены состояний ЭВМ является марковским.

На рис. 2.7 приведен граф состояний ЭВМ с частичным контролем при отсутствии периодических профилактических испытаний, а на рис. 2.8 — при их наличии. Предполагается, что в охваченном контролем оборудовании отказы обнаруживаются сразу после их возникновения и начинается ремонт. Модель, показанная на рис. 2.7, построена в предположении, что для перевода ЭВМ в ремонт после не обнаруживаемого схемами контроля отказа необходим второй обнаруживаемый отказ.

Будем рассматривать влияние периода между профилактическими испытаниями на коэффициент простоя

$$k_n = 1 - k_r, \quad (2.82)$$

где k_r — коэффициент готовности.

Для определения оптимальной частоты профилактических испытаний, при которой коэффициент простоя имеет наименьшее значение, необходимо взять производную k_n по λ_n и приравнять ее нулю.

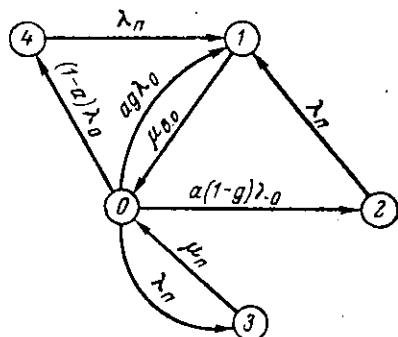


Рис. 2.9. Граф состояний ЭВМ с учетом аппаратуры, не участвующей в вычислительном процессе:

0—3 — то же, что на рис. 2.8; 4 — состояние отказа в аппаратуре, не участвующей в вычислительном процессе; α — доля аппаратуры, участвующей в вычислительном процессе

Приближенное значение коэффициента простоя можно определить в виде [49]

$$k_n = g\lambda_0/\mu_{в.о} + (1 - g)\lambda_0/2\lambda_n + \lambda_n/\mu_n. \quad (2.83)$$

Возьмем производную

$$\partial k_n / \partial \lambda_n = - (1 - g)\lambda_0 / 2\lambda_n^2 + 1/\mu_n$$

и, приравняв ее нулю, найдем оптимальное значение

$$\lambda_n = \sqrt{\frac{1}{2} (1 - g)\lambda_0 \mu_n}.$$

Подставив вместо λ_n и μ_n их значения, выраженные через период и длительность профилактических испытаний T_n и τ_n , получим оптимальное значение периода профилактических испытаний

$$T_n = \sqrt{2\tau_n / (1 - g)\lambda_0}. \quad (2.84)$$

Обычно в ЭВМ имеется довольно значительное количество аппаратуры, не участвующей в вычислительном процессе. Сюда относятся, например средства обслуживания. Отказ в этой аппаратуре не приводит к отказу ЭВМ. Однако для дальнейших рассуждений имеет смысл изменить модель, показанную на рис. 2.8, так, чтобы выделить отказы в этой аппаратуре из общего потока отказов. Эта модель показана на рис. 2.9.

Поскольку отказ в аппаратуре, не участвующей в вычислительном процессе, не приводит к отказу ЭВМ, оптимальный период профилактических испытаний для этого случая совпадает с (2.84).

Введем понятие относительного времени простоя

$$t_{п.о} = k_{п2}/k_{п1}, \quad (2.85)$$

где $k_{п2}$ — коэффициент простоя в модели, учитывающей частичный контроль и периодические профилактические испытания, а $k_{п1}$ — коэффициент простоя для «идеальной модели», т. е. $k_{п1} = \lambda_0/\mu_{в.о.}$

Под идеальной моделью подразумевается модель для ЭВМ со 100 %-ным контролем и без периодических профилактических испытаний.

Зависимость относительного времени простоя от длительности профилактических испытаний при их оптимальной периодичности имеет вид, показанный на рис. 2.10.

Из рис. 2.10 видно, что наличие не обнаруживаемых схемами контроля отказов затрудняет уменьшение времени простоя до значения, соответствующего идеальной модели. На рис. 2.11 показана зависимость коэффициента простоя от частоты профилактических испытаний.

Из приведенного анализа следует, что при возрастании доли необнаруженных отказов и уменьшении длительности профилактических испытаний последние должны выполняться чаще. Дополнительная составляющая (по отношению к значению коэффициента простоя в идеальной модели, равному $\lambda_0/\mu_{в.о.}$) определяется временем нахождения ЭВМ в состоянии отказа и профилактических испытаний.

Для сокращения времени простоя ЭВМ при профилактических испытаниях часть этих испытаний выполняется в мультипрограммном режиме параллельно с решением задач на ЭВМ. Обычно организуется параллельное (мультипрограммное) выполнение тестов аппаратуры, не охваченной контролем и не участвующей в вычислительном процессе (подробнее см. гл. 12).

Для определения оптимальной периодичности профилактических испытаний как требующих, так и не требующих останова ЭВМ, преобразуем модель, приведенную на рис. 2.9, в модель, приведенную на рис. 2.12. Тесты, выполняемые при профилактических испытаниях, не требующих останова ЭВМ, назовем параллельными, а тесты, выполняемые при останове ЭВМ, — автономными.

Модель, приведенная на рис. 2.12, сочетает как параллельное, так и автономное тестирование.

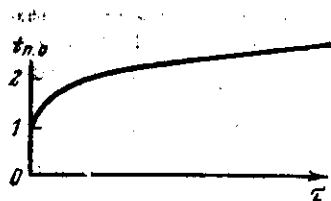


Рис. 2.10. Влияние длительности профилактических испытаний на относительное время простоя

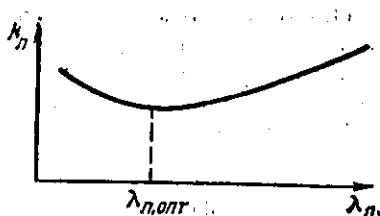


Рис. 2.11. Влияние частоты профилактических испытаний на коэффициент простоя:

$\lambda_{п.опт}$ — оптимальная частота профилактических испытаний

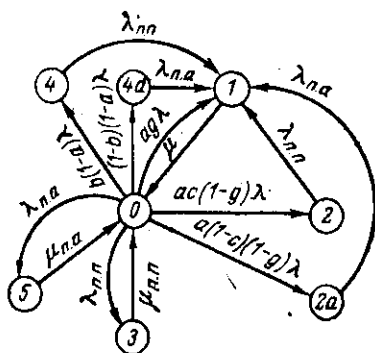


Рис. 2.12. Граф состояний ЭВМ с учетом аппаратуры, проверяемой параллельными и автономными тестами:

b — доля аппаратуры, не участвующей в вычислительном процессе, которую можно проверить путем параллельного тестирования; $(1-b)$ — доля аппаратуры, не участвующей в вычислительном процессе, которая проверяется только с помощью автономных тестов; c — доля аппаратуры, не охваченной контролем, которую можно проверить путем параллельного тестирования; $(1-c)$ — доля аппаратуры, не охваченной контролем, которая проверяется только с помощью автономных тестов; $\lambda_{п.п}$ — частота выполнения параллельных тестов, причем $\lambda_{п.п} = 1/T_{п.п}$, где $T_{п.п}$ — период параллельных тестов; $\lambda_{п.а}$ — частота выполнения автономных тестов, причем $\lambda_{п.а} = 1/T_{п.а}$, где $T_{п.а}$ — период выполнения автономных тестов; $\mu_{п.п} = 1/\tau_{п.п}$, где $\tau_{п.п}$ — длительность параллельных тестов; $\mu_{п.а} = 1/\tau_{п.а}$, где $\tau_{п.а}$ — длительность автономных тестов; 2, 2a — состояния отказа в аппаратуре, не охваченной контролем, обнаруживаемые соответственно с помощью параллельных и автономных тестов; 3 — состояние выполнения автономных тестов; 4, 4a — состояния отказа в аппаратуре, не участвующей в вычислительном процессе, обнаруживаемые соответственно с помощью параллельных и автономных тестов

Для модели, приведенной на рис. 2.12, коэффициент простоя

$$k_{\Pi} \approx \frac{\lambda_0}{\mu_{\Pi.0}} + \frac{\lambda_{\Pi.П}}{\mu_{\Pi.П}} + \frac{\lambda_{\Pi.А}}{\mu_{\Pi.А}} + \frac{ac(1-g)\lambda_0}{\lambda_{\Pi.П}} + \frac{a(1-c)(1-g)\lambda_0}{\lambda_{\Pi.А}} \quad (2.86)$$

Определив частные производные $\partial k_{\Pi}/\partial \lambda_{\Pi.П}$ и $\partial k_{\Pi}/\partial \lambda_{\Pi.А}$ и приравняв частные производные нулю, получим оптимальные частоты выполнения параллельных и автономных тестов:

$$\lambda_{\Pi.П.опт} = \sqrt{\mu_{\Pi.П} ac(1-g)\lambda_0}; \quad (2.87)$$

$$\lambda_{\Pi.А.опт} = \sqrt{\mu_{\Pi.А} a(1-c)(1-g)\lambda_0}. \quad (2.88)$$

2.5. МОДЕЛИ ЗИП

В процессе эксплуатации ЭВМ отказавшие элементы, ТЭЗ и узлы в ходе ремонтных работ заменяются работоспособными; как правило, новые элементы и узлы берутся из комплекта запасных инструментов и приборов (ЗИП).

Комплект ЗИП формируется на заводе, поставляющем ЭВМ, и затем пополняется по заявкам вычислительного центра. Завод-изготовитель формирует комплект ЗИП из различных элементов, ТЭЗ и узлов ЭВМ — на основе статистических данных об эксплуатации ЭВМ.

В силу того что отказы ЭВМ возникают в случайные моменты времени, планирование объема ЗИП по отдельным компонентам должно производиться на основе вероятностных математических моделей процессов расхода и пополнения ЗИП.

Рассмотрим методику построения таких моделей.

Будем считать, что ЗИП ЭВМ содержит n_i запасных элементов i -го типа; общее число элементов (считая и тот, что входит в состав ЭВМ) равно $n_i + 1 = k_i$. Определим вероятность того, что запасные элементы i -го типа будут израсходованы не раньше чем за время T [54].

Если обозначить через $\xi_1, \xi_2, \dots, \xi_k$ случайные величины — времена работы элемента, входящего в состав ЭВМ, и 1-го, ..., $(k-1)$ -го элементов из ЗИП, то случайное время от момента начала работы ЭВМ с основным элементом до момента отказа ЭВМ из-за того, что отказал последний, n_i -й элемент из ЗИП, есть случайная величина

$$S_i = \sum_{j=1}^{n_i+1} \xi_j. \quad (2.89)$$

Очевидно, что выражение верно для случая, когда отказавший элемент заменяется новым из ЗИП, а ЗИП на всем интервале S_i не пополняется элементами i -го типа.

Если предположить, что $\xi_1, \xi_2, \dots, \xi_k$ — независимые в совокупности и одинаково распределенные случайные величины, имеющие, например, экспоненциальное распределение с параметром λ_i , то случайная величина S_i будет иметь распределение Эрланга n_i+1 -го порядка с плотностью (см. § 2.1)

$$f(t) = \lambda_i (\lambda_i t)^{k-1} e^{-\lambda_i t} / (k-1)!, \quad k \geq 0, \quad (2.90)$$

где $k = n_i + 1$ — порядок распределения.

Нетрудно заметить, что вероятность безотказной работы ЭВМ (в течение времени T) при наличии в ЗИП n_i запасных элементов i -го типа

$$P(S_i \geq T) = 1 - F_k(t) = 1 - \int_0^T \frac{\lambda_i (\lambda_i t)^{k-1}}{(k-1)!} e^{-\lambda_i t} dt, \quad (2.91)$$

где $F_k(t)$ — функция распределения Эрланга k -го порядка, причем

$$F_k(t) = 1 - \sum_{r=0}^{k-1} (\lambda_i t)^r e^{-\lambda_i t} / r! \quad (2.92)$$

С учетом (2.92) выражение (2.91) можно представить в виде

$$P(S_i \geq T) = \sum_{r=0}^{k-1} (\lambda_i t)^r e^{-\lambda_i t} / r! \quad (2.93)$$

Анализируя полученное выражение, можно сделать вывод, что для выполнения условия $S_i \geq T$ необходимо, чтобы за время T число отказавших элементов m было меньше n_i+1 . Действительно, r -е слагаемое в (2.93) есть не что иное, как выражение для вероятности того, что за время T откажет ровно r элементов:

$$P(m=r) = (\lambda_i T)^r e^{-\lambda_i T} / r!, \quad r = \overline{0, n_i}. \quad (2.94)$$

Таким образом, вероятность безотказной работы ЭВМ в течение интервала T есть вероятность того, что за интервал T откажет не более n_i элементов:

$$P(S_i \geq T) = \sum_{r=0}^{n_i} P(m=r) = \sum_{r=0}^{n_i} (\lambda_i T)^r e^{-\lambda_i T} / r! \quad (2.95)$$

Выражение (2.94) фактически определяет закон Пуассона для дискретной случайной величины — числа отказов за интервал T .

Этот результат имеет достаточно простую физическую интерпретацию; закон Пуассона — закон редких событий, а именно такими и являются отказы ЭВМ.

Если интервал T достаточно велик, то тогда величина S_i будет иметь распределение, близкое к нормальному с функцией распределения

$$F(t_i) = 1 - \Phi \left[\frac{t - M[S_i]}{\sqrt{D[S_i]}} \right],$$

где $M[S_i]$, $D[S_i]$ — математическое ожидание и дисперсия случайной величины S_i ; $\Phi[t] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-\tau^2/2} d\tau$ — функция

Лапласа.

Таблицы значений функций Лапласа приводятся во многих справочниках.

Если ЭВМ содержит N различных типов элементов, ТЭЗ и устройств, заменяемых в момент отказа за счет ЗИП, общая вероятность безотказной работы ЭВМ в течение интервала T в предположении, что отказы всех типов независимы между собой,

$$P(T_0 \geq T) = P^* \prod_{i=1}^N P(S_i \geq T), \quad (2.96)$$

где T_0 — время безотказной работы ЭВМ; P^* — вероятность безотказной работы в течение времени T элементов, ТЭЗ и устройств ЭВМ, не обеспеченных комплектом ЗИП.

Зная интенсивности потоков отказов элементов каждого типа, на основании (2.96) можно вычислить вероятность безотказной работы ЭВМ при известных числах ЗИП n_i . Для упрощения расчетов [при определении таких n_i , при которых вероятность $P(T_0 \geq T)$ не меньше некоторого заданного значения], можно воспользоваться номограммами, приведенными в [52].

Полученная модель расхода ЗИП не учитывает весьма существенного фактора: пополнения ЗИП в том случае, если из-за расхода элементов в ходе ремонтных работ число каких-либо элементов в ЗИП станет меньше числа, определенного правилами эксплуатации ЭВМ.

Действительно, как только из ЗИП изымается какой-то элемент, в организацию, снабжающую ВЦ запасными частями, направляется запрос на поставку элементов соответствующего типа. Время удовлетворения заявки случайно и не зависит от того, на сколько единиц запасных частей и какого типа оформлена заявка.

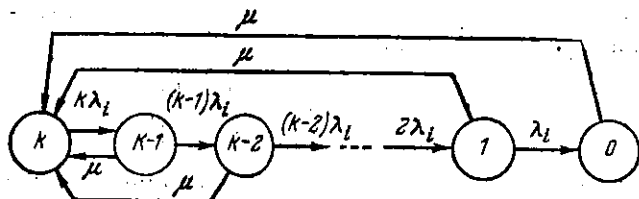


Рис. 2.13. Граф состояний ЗИП при расходе и пополнении элементов i -го типа

Процесс расхода и пополнения ЗИП элементами i -го типа может быть представлен в виде графа (рис. 2.13). Будем считать, что состояние ЗИП по наличию i -х элементов в некоторый момент времени t описывается числом соответствующих элементов, находящихся в ЗИП, плюс элемент, работающий в составе ЭВМ. Очевидно, если нормативный запас ЗИП по i -му типу элементов равен n_i , то исходное состояние ЗИП $k = n_i + 1$. Так как в конкретный момент времени работает только один элемент, подверженный отказам с интенсивностью λ_i , то возможен переход из начального состояния $k = n_i + 1$ только в состояние $k - 1 = n_i$ с интенсивностью λ_i . На графе этому переходу соответствует дуга из состояния k в состояние $k - 1$ (дуга помечена интенсивностью λ_i). В этот момент из ЗИП извлекается запасной элемент для замены отказавшего. Из состояния $k - 1$ возможен и обратный переход, если будет удовлетворена заявка на выбывший элемент из ЗИП, а за это время ЭВМ, работая с элементом, изъят из ЗИП, не откажет из-за отказа элемента того же типа. Если отказ того же типа произойдет за время, меньшее времени удовлетворения заявки, то ЭВМ переходит в состояние $k - 2$ с интенсивностью λ_i , и снова из ЗИП будет извлечен элемент. Теперь будет аннулирована предыдущая заявка и оформлена новая на два элемента, и поэтому переходы возможны как в состояние $k - 3$, так и в состояние k , причем с той же интенсивностью, что и переход из состояния $k - 1$ в состояние

k . На графе эта интенсивность перехода, обозначенная μ , есть величина, обратная среднему времени удовлетворения заявки на запасные элементы для пополнения ЗИП.

Эффективность работы ЭВМ с подобным процессом расхода и пополнения ЗИП может быть оценена с помощью вероятности того, что ЭВМ будет простаивать из-за отсутствия элементов i -го типа, т. е. вероятности того, что ЭВМ будет в состоянии $k=0$. Обозначим вероятность нахождения в этом состоянии в установившемся режиме работы (при достаточно большой длительности эксплуатации ЭВМ) π_0 . Через π_j , $j=\overline{0, k}$ обозначим вероятность нахождения процесса в j -м состоянии.

Если предположить, что длительность исправной работы элемента j -го типа есть случайная величина, имеющая экспоненциальное распределение с математическим ожиданием $1/\lambda_i$ (интенсивностью отказов λ_i), а время удовлетворения заявки на элементы для ЗИП — случайная величина, имеющая экспоненциальное распределение с интенсивностью μ , то можно записать систему уравнений для вероятностей π_j ($j=\overline{0, k+1}$):

$$\left. \begin{aligned} -\lambda_i \pi_k + \mu \sum_{j=1}^k \pi_{k-j} &= -(\lambda_i + \mu) \pi_k + \mu = 0; \\ -(\lambda_i + \mu) \pi_{k-j} + \lambda_i \pi_{k-j+1} &= 0, \quad j = \overline{1, k-1}; \\ \lambda_i \pi_1 - \mu \pi_0 &= 0. \end{aligned} \right\} \quad (2.97)$$

Решая (2.97), получаем

$$\left. \begin{aligned} \pi_k &= \mu / (\mu + \lambda_i); \\ \pi_{k-j} &= \mu \lambda_i^j / (\mu + \lambda_i)^{j+1}, \quad j = \overline{1, k-1}; \\ \pi_0 &= \lambda_i^k / (\mu + \lambda_i)^k. \end{aligned} \right\} \quad (2.98)$$

Таким образом, получено выражение, с помощью которого можно оценить вероятность того, что ЭВМ будет простаивать из-за отсутствия i -х элементов в ЗИП при условии, что нормативное число элементов i -го типа в ЗИП равно n_i . Оценив вероятности простоя по каждому из типов элементов, ТЭЗ и узлов, на основании (2.96) можно оценить вероятность безотказной работы ЭВМ в целом.

В заключение отметим следующее. Если в состав ЭВМ входит C_i одновременно работающих элементов i -го типа, то граф состояний ЗИП останется практически тем же. Различие будет состоять только в том, что переходы из состоя-

ний $k-j$ в состоянии $k-j-1$ будут происходить не с интенсивностью λ_i , а с интенсивностью $C_i \lambda_i$. Соответственно в (2.98) необходимо вместо λ_i подставить $C_i \lambda_i$.

2.6. ПОСТРОЕНИЕ МОДЕЛЕЙ НАДЕЖНОСТИ ПО ЭКСПЕРИМЕНТАЛЬНЫМ ДАННЫМ

Приведенные в настоящей главе аналитические модели надежности носят в определенной степени абстрактный характер. Конкретный вид и параметры аналитической модели надежности (модели отказов и др.) могут быть определены по экспериментальным данным о работе машины в процессе ее испытаний или эксплуатации.

В процессе эксплуатации ЭВМ в «Эксплуатационном журнале» должны регистрироваться данные о продолжительности интервалов безотказной (и бессбойной) работы ЭВМ в целом и отдельных ее устройств, моментах возникновения отказов (с указанием отказавшего устройства и причины отказа), моментах появления сбоев (с указанием узла, в котором обнаружены сбои), продолжительности работ по устранению отказов и работ по восстановлению достоверности информации после сбоев, о времени проведения, продолжительности и результатах профилактических испытаний и другие данные о работе машины.

Эти данные могут быть использованы для определения введенных в гл. 1 количественных значений основных эксплуатационных характеристик ЭВМ. Вместе с тем эти данные позволяют получать и такие вероятностные характеристики надежности, как законы распределения времени между отказами, времени восстановления и некоторых других случайных величин, т.е. по существу позволяют строить экспериментальные модели надежности машины.

Эти модели могут быть положены в основу расчетов и имитационного моделирования надежности соответствующих вычислительных комплексов.

Зафиксированные в журнале длительности интервалов нормальной работы между отказами или продолжительности работ по восстановлению можно рассматривать как некоторую выборку из всего множества значений, которые принимает данная случайная величина (выборку конечного объема из генеральной совокупности данной случайной величины).

Пусть $x_1, x_2, \dots, x_N$ — независимые измерения исследуемой случайной величины, причем x_i называется выборочным значением, а N — объемом выборки.

Рассмотрим, каким образом, обрабатывая результаты наблюдений, можно установить, какому теоретическому закону распределения подчиняется исследуемая случайная величина.

Выделим три основные задачи, которые следует последовательно решить. Первая задача — это задача об определении числовых харак-

характеристик случайной величины: математического ожидания, дисперсии (среднего квадратического отклонения), асимметрии, эксцесса и т. п.; другими словами, это задача определения по выборке объемом N выборочных начальных и центральных моментов случайной величины [14].

Вторая задача состоит в определении соответствующего теоретического распределения. Это — задача сглаживания или выравнивания статистических данных.

На основании выборочных значений случайной величины, сгруппированных специальным образом, выдвигается гипотеза о том, что исследуемая случайная величина — время наработки на отказ, время восстановления и т. д. — может быть описана тем или иным теоретическим распределением. При этом чаще всего параметры теоретического распределения определяются также по самой выборке, например исходя из того, что моменты теоретического распределения должны быть равны выборочным моментам (метод моментов).

Как правило, гипотеза о теоретическом распределении выдвигается на основании опыта исследователя; очевидно, что возможно выдвижение и неверной гипотезы.

Третья задача состоит в проверке гипотезы (правдоподобия гипотезы). Она решается путем вычисления специальных статистик — величин, характеризующих степень расхождения между опытными данными и теоретическим распределением. Статистика формируется таким образом, что заранее известно ее распределение; это позволяет определить вероятность того, что выборка может быть описана принятым теоретическим распределением.

Кроме перечисленных задач могут возникнуть и другие, например задача об определении объема выборки для получения достоверных оценок и т. д.

Первая из перечисленных задач решается достаточно просто (особенно при использовании ЭВМ).

Статистические оценки моментов получают как значения определенных функций от выборочных значений случайной величины. В частности, статистическая оценка r -го момента (начального или центрального) может быть принята равной выборочному r -му моменту (начальному или центральному). Последние вычисляются по формулам

$$\alpha_r^*[X] = \frac{1}{N} \sum_{i=1}^N x_i^r \quad (2.99)$$

$$\mu_r^*[X] = \frac{1}{N} \sum_{i=1}^N (x_i - \alpha_1^*[X])^r, \quad (2.100)$$

где α_r^* и μ_r^* — соответственно начальный и центральный выборочные моменты r -го порядка. Очевидно, что статистические оценки моментов являются случайными величинами.

Первый начальный момент — это выборочное среднее для его обозначения часто применяется символ $\bar{X}$.

Если выборочное среднее значение $\bar{X}$ вычисляется на выборке объемом N , то среднее квадратическое отклонение случайной величины $\bar{X}$ от математического ожидания $M[\bar{X}]$ равно

$$s_{\bar{X}} = \sqrt{s_X/N}, \quad (2.101)$$

где $s_X = \sigma_X^* = \sqrt{\mu_2^*}$ — выборочное среднее квадратическое отклонение.

Можно показать [14], что $\bar{X}$ имеет нормальное распределение; это позволяет решить задачу об определении объема выборки.

Аналогично по выборке вычисляются и коэффициенты асимметрии и эксцесса.

Построение статистического ряда и гистограммы. Весь интервал наблюдения случайной величины X разбивают на k интервалов (разрядов); величина интервала Δx определяется по формуле

$$\Delta x = (x_{\max} - x_{\min})/k, \quad (2.102)$$

где $x_{\max}$, $x_{\min}$ — соответственно максимальное и минимальное значение случайной величины, определенное по выборке объемом N . Затем подсчитывается число попаданий случайной величины в интервалы:

$[x_{\min}, x_{\min} + \Delta x)$, $[x_{\min} + \Delta x, x_{\min} + 2\Delta x)$, ..., $[x_{\max} - \Delta x, x_{\max}]$.

Обозначим число попаданий в i -й интервал через N_i . Отношение

$$p_i = N_i/N \quad (2.103)$$

означает частоту попадания случайной величины в интервал $[x_i, x_i + \Delta x)$.

Величины p_i задают статистический ряд случайной величины, построенный по выборке конечного объема.

По статистическому ряду можно построить гистограмму — эмпирическую плотность распределения вероятностей случайной величины x . Для этого вычисляются величины

$$p_i^* = p_i/\Delta x = N_i/\Delta x N. \quad (2.104)$$

Тогда кусочно-линейная функция, принимающая значения p_i^* на интервале $[x_i, x_i + \Delta x)$, будет ограничивать площадь, равную 1; график этой функции и представляет собой гистограмму.

На основании внешнего вида гистограммы можно сделать выводы о том, какие теоретические плотности распределения вероятностей целесообразно попытаться использовать для выравнивания гистограммы. Анализ численных значений выборочных моментов, в частности таких как коэффициент вариации, асимметрия и эксцесс, позволит отобрать из всего множества теоретических распределений подходящие для выравнивания полученной гистограммы.

Пример. В процессе эксплуатации ЭВМ фиксировались отказы в механизме устройства ввода с перфокарт; значения интервалов между отказами (в часах) приведены в табл. 2.1. Общее число измерений $N = 100$.

Для построения гистограммы разобьем весь интервал от минимального значения, равного 0,81, до максимального, равного 571,55 ≈ 580 ч, на k интервалов. Для определения значения k воспользуемся часто применяемой в математической статистике формулой Старгесса [2]

$$k = 3,3 \lg N + 1 = 3,3 \cdot 2 + 1 = 6,6 + 1 = 7,6 \approx 8.$$

Значение разряда гистограммы Δx равно, таким образом, 72,5 ч. Числа попаданий в интервал N_i ; значения частоты p_i^* приведены в табл. 2.2 с учетом того, что в интервал [435; 507,5] не попало ни одного измерения.

Т а б л и ц а 2.1. Случайные значения времен наработки на отказ

N	x_i , ч	N	x_i , ч	N_i	x_i , ч	N	x_i , ч	N	x_i , ч
1	72,5	21	112,11	41	26,66	61	1,91	81	218,84
2	90,35	22	186,9	42	128,04	62	1,16	82	92,66
3	58,37	23	70,83	43	129,53	63	50,91	83	14,32
4	161,36	24	271,09	44	44,37	64	156,11	84	199,4
5	39,32	25	76,24	45	12,14	65	104,87	85	65,76
6	27,85	26	35,53	46	324,59	66	33,78	86	95,94
7	3,32	27	67,98	47	27,02	67	46,76	87	199,53
8	70,92	28	146,16	48	31,65	68	17,57	88	14,15
9	29,29	29	110,53	49	571,55	69	11,14	89	72,46
10	60,08	30	91,67	50	3,86	70	114,5	90	51,97
11	258,65	31	8,43	51	141,54	71	102,62	91	34,6
12	408,75	32	87,65	52	121	72	24,05	92	93,43
13	7,99	33	32,12	53	226,5	73	0,81	93	71,37
14	11,88	34	225,01	54	82,27	74	97,93	94	10,17
15	65,55	35	50,19	55	160,53	75	19,05	95	67,05
16	47,22	36	169,45	56	28,5	76	253,92	96	82,28
17	56,38	37	187,19	57	158,58	77	62,47	97	63,03
18	119,78	38	26,15	58	4,02	78	524,32	98	29,86
19	166,17	39	30,82	59	86,63	79	36,64	99	183,76
20	8,08	40	2,04	60	97,4	80	44,33	100	25

На рис. 2.14 приведена гистограмма; исходя из вида эмпирической плотности распределения и значения коэффициента вариации, близкого к 1 (в рассматриваемом примере $s_x/\bar{x} = 1,09$), можно сделать предположение о том, что исследуемая случайная величина может быть описана экспоненциальным распределением с плотностью

$$f(x) = \exp(-x/96,48)/96,48. \quad (2.105)$$

Таблица 2.2. Пример расчета χ^2 -критерия (объем выборки $N=100$)

Номер интервала	Границы интервала, ч	N_i	Np_i	Номер интервала	Границы интервала, ч	N_i	Np_i
1	0—72,5	56	53,015	5	290—362,5	1	2,585
2	72,5—145	22	24,912	6	362,5—435	1	1,215
3	145—217,5	12	11,707	7	435—507,5	0	—
4	217,5—290	6	5,501	8	507,5—580	2	0,839

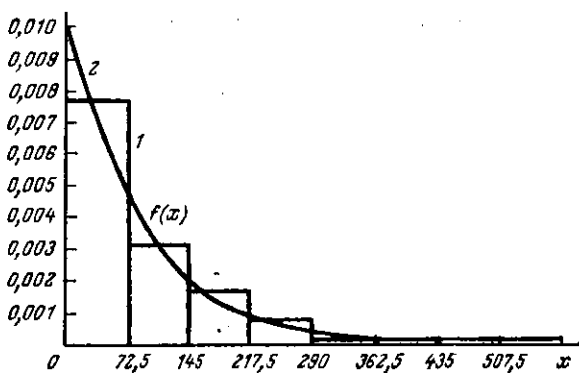


Рис. 2.14. Гистограмма (1) и теоретическая функция экспериментального распределения (2), построенная по выборке (см. табл. 2.1)

Проверка статистических гипотез. Выдвинутое предположение об экспоненциальном распределении случайной величины — это всего лишь гипотеза, и она должна быть подтверждена. При проверке гипотезы должен быть получен ответ на основной вопрос — расхождения между эмпирическими (полученными в ходе обработки выборки) и теоретическими значениями функции распределения вероятностей объясняются только конечным объемом выборки или причина в том, что произведена выборка из генеральной совокупности, подчиняющейся другому закону распределения?

Проверка гипотез выполняется с помощью специальных критериев согласия, служащих мерой расхождения эмпирического и теоретического распределений. Например, можно вычислить значение так называемого χ^2 -критерия:

$$\chi^2_{\text{расч}} = \sum_{i=1}^k \frac{(N_i - Np_i)^2}{Np_i}, \quad (2.106)$$

где p_i — теоретическая вероятность попадания случайной величины в i -й разряд гистограммы для предполагаемого теоретического распределения.

Очевидно, что в силу случайного характера величин N_i (принимających различные значения при различных выборках одного и того же объема N , не говоря уже о выборках с различным числом испытаний) χ^2 будут также случайными величинами.

Показано [14], что плотность распределения этой случайной величины имеет вид

$$f_r(u) = \frac{u^{(r/2)-1} e^{-u/2}}{2^{r/2} \Gamma\left(\frac{r}{2}\right)}, \quad u > 0, \quad (2.107)$$

где $\Gamma(r/2)$ — гамма-функция от аргумента $r/2$; r — число независимых случайных величин, число степеней свободы случайной величины χ^2 , для которых вычисляется сумма квадратов, причем предполагается, что суммируемые случайные величины имеют нормальное распределение $N(0, 1)$.

Так как распределение случайной величины χ^2 известно, то, получив в результате расчета по (2.106) ее значение, можно определить, с какой вероятностью для заданного r случайная величина $\chi^2_{\text{расч}}$ превзойдет значение χ^2 — вероятность того, что за счет малого объема выборки (случайного характера χ^2) расхождение между эмпирическим и теоретическим распределениями будет не меньше значения χ^2 , полученного в результате обработки эксперимента. Другими словами, это — вероятность справедливости гипотезы о том, что исследуемая случайная величина подчиняется данному закону распределения.

Распределение χ^2 табулировано для различных значений r , что упрощает процедуру проверки гипотез. Нужно только знать, чему равно r . Как было отмечено ранее, это — число независимых суммируемых случайных величин. В данном случае на частоты попадания в интервал наложено ограничение — их сумма равна 1. Кроме того, по результатам выборки определяют и сами параметры теоретического распределения, так как они заранее неизвестны. Отсюда следует, что при проверке гипотез r равно k , уменьшенному на 1 и на число параметров распределения, вычисляемых по выборке.

Процедура проверки гипотез может быть упрощена, если используется так называемый критерий согласия Романовского [14], вычисляемый по формуле

$$R = \left| \chi^2_{\text{расч}} - r \right| / \sqrt{2r}, \quad (2.108)$$

где $\chi^2_{\text{расч}}$ — расчетное значение χ^2 -критерия, полученное по (2.106).

Оказывается, что если $R < 3$, то согласие между эмпирическим и теоретическим распределениями можно считать удовлетворительным (гипотеза о теоретическом распределении при этом принимается).

Заметим, что выбирать разряды гистограммы нужно так, чтобы число попаданий в интервал было не менее 5, в противном случае значение χ^2 будет завышено. Если по результатам выборки получается, что N_i для i -го разряда меньше 5, следует объединить соседние разряды.

Если обратиться к рассмотренному ранее примеру по статистическому анализу времени наработки на отказ, то получим следующие значения критерия согласия: $\chi^2_{\text{расч}} = 3,176$; $R = 0,815$, число степеней свободы $k = 5$. По таблицам χ^2 -распределения находим, что при $k = 5$, $p = 0,7$ $\chi = 3,00$, а при $p = 0,5$ $\chi^2 = 4,31$. Отсюда следует, что вероятность справедливости гипотезы об экспоненциальном распределении (доверительная вероятность) не менее 0,68.

На практике гипотеза может приниматься, если p не менее 0,1, так что в рассматриваемом примере согласие между экспериментальным и теоретическим распределениями удовлетворительное. Тот же вывод можно сделать и при применении критерия Романовского (его значение в рассматриваемом случае меньше 3).

2.7. ИМИТАЦИОННЫЕ МОДЕЛИ ПРОЦЕССОВ ЭКСПЛУАТАЦИОННОГО ОБСЛУЖИВАНИЯ ЭВМ

В рассмотренных в предыдущих параграфах аналитических вероятностных моделях потоков отказов, сбоев, восстановлений принималось допущение об экспоненциальном распределении времен наработки на отказ, поиска и замены отказавшего элемента и др. Подобное распределение не всегда удовлетворительно описывает ту или иную выборку, полученную в процессе наблюдения за работой ЭВМ. Отказ от экспоненциального распределения делает вероятностную модель процесса эксплуатации настолько сложной, что оказывается невозможным построить модель в виде явных соотношений. В подобных случаях приходится прибегать к статистическому (имитационному) моделированию.

При реализации на ЭВМ статистических моделей процессов эксплуатации производится многократный розыгрыш случайных ситуаций, которые могут иметь место при известных законах распределения таких случайных величин, как время между отказами T_0 , время поиска отказавшего элемента t_n , время замены t_z и т. д. В процессе многократных розыгрышей производится сбор статистической информации, например конкретных реализаций суммы трех случайных величин T_0 , t_n , t_z с целью определения коэффициента

готовности k . Накопленные реализации могут рассматриваться как выборка конечного объема N (N — общее число розыгрышей) и обрабатывается теми же методами, что и выборка, полученная в ходе натурных наблюдений. Таким образом, натурный эксперимент по определению надежных и других эксплуатационных характеристик заменяется машинным, в ходе которого имитируется процесс функционирования ЭВМ. В силу этого статистические модели часто называют *имитационными моделями* процессов эксплуатации.

Очевидно, что для того чтобы иметь возможность поставить на ЭВМ имитационный эксперимент, необходимо:

- а) программным образом получать случайные величины с заданным законом распределения;
- б) описывать процессы порождения запросов на обслуживание, а также процессы формирования очередей и собственно обслуживания;
- в) накапливать статистическую информацию в ходе машинного эксперимента.

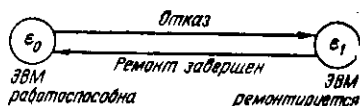
Автоматизировать процедуру построения имитационных моделей позволяют специальные системы моделирования, такие, например, как система GPSS (общецелевая система моделирования дискретных систем) [77]. В системе GPSS процессы порождения запросов (называемых в GPSS транзактами), формирования очередей обслуживания описываются с помощью так называемых объектов. В GPSS имеются четыре вида объектов:

- 1) динамические (транзакты), показывающие, каким образом продвигается запрос по системе, получая те или другие виды обслуживания;
- 2) статические, описывающие работу приборов обслуживания (как одноканальных, так и многоканальных) с очередями различных типов, а также оборудования того или иного рода;
- 3) статистические, предназначенные для накопления и обработки статистической информации в процессе моделирования;
- 4) операционные блоки, задающие изменение состояний транзактов, оборудования и статистических объектов в момент прохождения транзактом блока.

Особенностью GPSS является возможность использования эмпирических функций распределения случайных величин, полученных в результате натурального эксперимента, для построения датчиков случайных величин.

В качестве иллюстрации метода построения имитационных GPSS-моделей процесса эксплуатационного обслуживания рассмотрим простейший пример. Примем, что ЭВМ может находиться в двух состояниях: e_0 — ЭВМ работоспособна и e_1 — ЭВМ отказала и восстанавливается. Соответствующий граф состояний ЭВМ показан на рис. 2.15.

Рис. 2.15. Граф состояний ЭВМ (к примеру построения имитационной модели процесса эксплуатационного обслуживания)



Эмпирические функции распределения, полученные по результатам натурных наблюдений для времени между моментами отказов и длительности восстановительных работ, приведены на рис. 2.16. Имитационная GPSS-модель для рассматриваемого примера процесса эксплуатационного обслуживания с учетом эмпирических распределений случайных параметров процесса эксплуатации приведена на рис. 2.17.

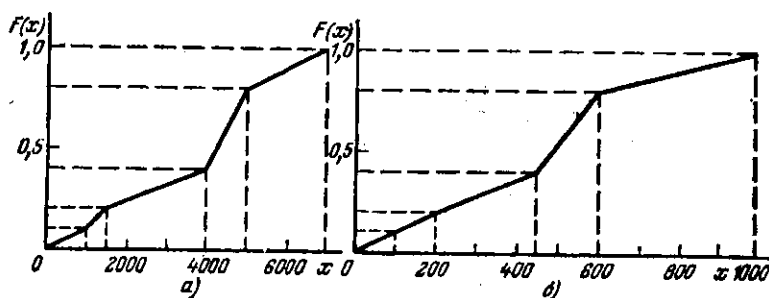


Рис. 2.16. Эмпирические функции распределения для времени (в единицах модельного времени) между отказами (а) и длительности восстановлений (б)

Модель содержит 12 операционных блоков; номера блоков проставлены в левой колонке рис. 2.17 (карты SIMULATE, RMULT, FUNCTION, TABLE не нумеруются, так как являются вспомогательными).

Блок 1 (GENERATE) обеспечивает порождение и ввод в модель одного транзакта — динамического объекта, который будет имитировать ЭВМ, находящуюся в различных состояниях. Блок 2 MARK с меткой INPUT запоминает мо-

мент входа транзакта в модель, причем запоминание производится в первом параметре транзакта. Тот факт, что в модели потребуется только один параметр для транзакта в виде целого 4-байтного числа, указывается в последних двух полях блока GENERATE.

BLOCK NUMBER	LOC	OPERATION	A,B,C,D,E,F,G	COMMENTS
		SIMULATE		
		RMULT	7,519	
		XTIME TABLE	MP1,0,400,20	
		TIME1 FUNCTION	RN1,C6	
		0,0/0,1,1000/0,2,1500/0,4,4000/0,8,5000/1,7000		
		TIME2 FUNCTION	RN2,C6	
		0,0/0,1,100/0,2,200/0,4,450/0,8,600/1,1000		
		GENERATE	1,1,1,F	
1				
2	INPUT	MANA	1	
3		SEIZE	COMP	
4		ADVANCE	FNQTIME1	
5		RELEASE	COMP	
6		SEIZE	SERV	
7		ADVANCE	FNQTIME2	
8		RELEASE	SERV	
9		TRANSLATE	XTIME	
10		SPLIT	1,INPUT	
11		TERMINATE	1	
		STANT	1000	
		END		

Рис. 2.17. Пример имитационной GPSS-модели эксплуатационного обслуживания

Транзакт, вошедший в модель, проходит через блок 3 (SEIZE); в момент прохождения транзакта через этот блок прибор обслуживания с именем COMP становится занятым. Этот прибор имитирует нормальную работу ЭВМ.

Прибор COMP находится в занятом состоянии в течение всего времени, пока транзакт задерживается блоком 4 (ADVANCE). Этот блок определяет время задержки с помощью функции с именем TIME1 (эта функция описывает эмпирическую функцию распределения случайной величины — времени между отказами ЭВМ).

Для розыгрыша конкретной случайной ситуации используется встроенный в GPSS датчик равномерно распределенных величин с именем RN1; его начальное значение (как и начальное значение RN2, используемого в другой функции) устанавливается картой RMULT.

После задержки транзакт попадает в блок 5, который освобождает прибор с именем COMP (блок RELEASE).

При этом автоматически запоминается информация, необходимая для последующего подсчета среднего числа транзактов, находящихся в приборе с именем COMP на всем интервале моделирования.

Блоки 6—8 (SEIZE, ADVANCE и RELEASE) моделируют нахождение ЭВМ в состоянии восстановления, при этом используется функция с именем TIME2, описывающая эмпирическую функцию распределения случайной величины — времени восстановления. Прибор, имитирующий нахождение ЭВМ в состоянии e_0 , имеет имя SERV.

Выйдя из блока RELEASE, освобождающего прибор с именем SERV, транзакт попадает в блок 9 (TABULATE). В момент прохода транзактом этого блока определяется сумма двух случайных величин: времен пребывания ЭВМ в состояниях e_0 и e_1 . Она вычисляется как разность значений текущего модельного времени и времени, запомненного в первом параметре транзакта в момент прохождения блока MARK. Накопление информации о сумме этих случайных величин происходит в виде гистограммы, которую задает блок TABULATE с именем XTIME (предполагается, что гистограмма будет иметь 20 разрядов, причем каждый разряд имеет значение, равное 400 временным единицам. За одну временную единицу принята 1 мин реального времени работы ЭВМ).

Блок TABULATE транзакт проходит без задержки, после чего поступает в блок 10 SPLIT. Этот блок порождает новый транзакт, который направляется вновь в блок MARK с меткой INPUT. Прежний транзакт направляется в блок 11 TERMINATE, в котором и уничтожается. Тем самым отмечается окончание цикла перехода ЭВМ из состояния e_0 в e_1 и обратно. В момент уничтожения транзакта из системного счетчика вычитается 1; как только значение счетчика станет равным 0, моделирование прекратится и на печать будут выданы обработанные результаты моделирования в виде таблиц, снабженных соответствующими заголовками (рис. 2.18). Начальное значение системного счетчика определяется картой START. Она указывает параметр, равный 1000. Это означает, что случайные реализации процесса эксплуатационного обслуживания будут воспроизводиться 1000 раз, что вполне достаточно для получения эмпирической функции распределения суммарного времени нахождения ЭВМ в работоспособном состоянии и состоянии восстановления.

Нетрудно заметить, что оценка средних коэффициентов использования приборов с именем SERV и COMP есть не

FACILITY	AVERAGE UTILIZATION	NUMBER ENTRIES	AVERAGE TIME/TRAN
COMP	.891	1000	3677.508
SERV	.108	1000	448.930

TABLE X TIME ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION
1000	4166.457	1786.000

UPPER LIMIT	OBSERVED FREQUENCY	PER CENT OF TOTAL	CUMULATIVE PERCENTAGE
0	0	.00	.0
400	6	.59	.5
800	24	2.59	3.1
1200	33	3.29	6.4
1600	66	6.59	13.0
2000	70	6.99	20.0
2400	40	3.99	24.0
2800	31	3.09	27.1
3200	31	3.09	30.2
3600	32	3.19	33.4
4000	28	2.79	36.2
4400	64	6.39	42.6
4800	113	11.29	53.9
5200	163	16.29	70.2
5600	136	13.59	83.8
6000	40	3.99	87.8
6400	40	3.99	91.8
6800	32	3.19	95.0
7200	32	3.19	98.2
OVERFLOW	17	1.69	100.0
AVERAGE VALUE OF OVERFLOW		7426.64	

END

Рис. 2.18. Результаты имитационного моделирования (пример)

что иное, как оценка вероятности простоя и коэффициента готовности ЭВМ соответственно.

Аналогично приведенной могут быть построены GPSS-модели более сложных процессов эксплуатационного обслуживания ЭВМ.

Глава третья

НАДЕЖНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1. ОСНОВНЫЕ ПОНЯТИЯ

Решение любой задачи, выполнение любой функции, возложенной на ЭВМ, возможно только при соответствующем взаимодействии и функционировании аппаратурных и

SUM OF ARGUMENTS
4126460.000

NON-WEIGHTED

CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
100.0	-.000	-2.310
99.3	.096	-2.086
96.7	.193	-1.862
93.5	.290	-1.638
86.8	.387	-1.414
79.9	.484	-1.190
75.9	.581	-.966
72.8	.678	-.742
69.7	.775	-.518
66.5	.872	-.294
63.7	.969	-.070
57.3	1.066	.153
46.0	1.163	.377
29.7	1.260	.601
16.1	1.357	.825
12.1	1.454	1.049
8.1	1.550	1.272
4.9	1.647	1.496
1.7	1.744	1.720
.0		

программных средств вычислительной машины. Поэтому при анализе надежности выполнения ЭВМ заданных функций следует рассматривать ЭВМ как единый комплекс ее аппаратурных и программных средств и учитывать, что надежность работы ЭВМ зависит не только от надежности аппаратуры, но и от надежности программного обеспечения.

По аналогии с терминологией, принятой для надежности технических средств, под *надежностью программного обеспечения* (ПО) будем понимать свойство выполнять заданные функции, сохранять свои характеристики в установленных пределах при определенных условиях эксплуатации.

Надежность программного обеспечения (или просто программы) определяется его (ее) безотказностью и восста-

навливаемостью. Безотказность программы или программного обеспечения есть ее (его) свойство сохранять работоспособность при использовании в процессе обработки информации на ЭВМ.

Безотказность программного обеспечения можно оценить вероятностью его работы без отказов при определенных условиях внешней среды в течение заданного периода наблюдения. В данном определении под отказом программы или системы программного обеспечения понимается недопустимое отклонение характеристик процесса функционирования программы от требуемых. Определенные условия внешней среды понимаются как совокупность входных данных и состояния вычислительной системы. Заданный период наблюдений соответствует, как правило, необходимому для выполнения решаемой на машине задачи.

Безотказность программного средства можно также характеризовать средним временем между возникновениями отказов в функционировании программы. При этом предполагается, что аппаратура ЭВМ находится полностью в работоспособном состоянии.

С точки зрения надежности принципиальное отличие программного обеспечения от аппаратуры состоит в том, что программы не изнашиваются и, следовательно, их выход из строя из-за поломки невозможен. Поэтому характеристики функционирования программного обеспечения зависят только от его качества, предопределяемого процессом разработки.

Безотказность программного обеспечения определяется его корректностью (правильностью) и, следовательно, целиком зависит от наличия в нем ошибок, внесенных на этапах его создания, в то время как безотказность аппаратуры определяется в основном случайными отказами, зависящими от изменений параметров аппаратуры, происходящих во время эксплуатации.

Надежность, или лучше сказать безотказность, аппаратуры и программного обеспечения существенно по-разному зависит от входных данных и времени функционирования системы. Обработываемые данные, как правило, не влияют на отказы аппаратуры. Процесс выхода из строя отдельных элементов аппаратуры не зависит от поступающих входных данных. В то же время проявление ошибок программного обеспечения связано с тем, что в некоторые моменты времени на обработку поступают ранее не встречавшиеся совокупности данных, которые программа не в состоянии корректно обработать. Таким образом, входные

данные в значительной мере влияют на функционирование программного обеспечения.

Безотказность аппаратуры по существу зависит от времени ее функционирования. В начале периода функционирования на этапе обнаружения и исправления ошибок проектирования и производственных дефектов интенсивность отказов аппаратуры уменьшается со временем. Затем в течение большей части срока службы она остается примерно постоянной. В конце срока службы интенсивность потока отказов значительно увеличивается вследствие износа аппаратуры. Зависимость надежности программного обеспечения от времени кажущаяся, косвенная. В действительности частота проявления ошибок определяется только входными данными.

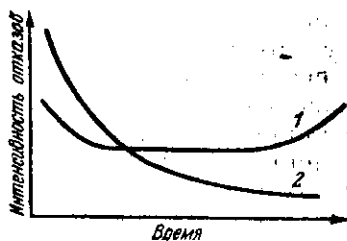


Рис. 3.1. Зависимость интенсивности отказов от времени эксплуатации аппаратурных (1) и программных (2) средств

На рис. 3.1 показаны типичные зависимости, отражающие изменение во времени надежности аппаратуры (кривая 1) и программы (кривая 2). Увеличение надежности программы является следствием того, что в процессе эксплуатации обнаруживаются и устраняются скрытые ошибки программы.

Важной характеристикой надежности программного обеспечения является его *восстанавливаемость*, которая определяется затратами времени и труда на устранение отказа из-за проявившейся ошибки в программе и его последствий. Восстановление после отказа в программе может заключаться в корректировке и восстановлении текста программы, исправлении данных, внесении изменений в организацию вычислительного процесса, что часто оказывается необходимым при работе ЭВМ в реальном масштабе времени. Восстанавливаемость программного обеспечения может быть оценена средней продолжительностью устранения ошибки в программе и восстановления ее работоспособности.

Восстанавливаемость программного обеспечения зависит от многих факторов, в том числе от сложности структуры комплекса программ, структурированности самих программ, алгоритмического языка, на котором разрабатывалась про-

грамма, стиля программирования, качества документации на программу и т. п.

Можно также говорить об *устойчивости функционирования программного обеспечения*, понимая под этим его способность ограничивать последствия собственных ошибок и неблагоприятных воздействий внешней среды (неисправности аппаратуры, некорректность входных данных, ошибки оператора и др.) или противостоять им.

Устойчивость программного обеспечения может быть повышена путем разных форм структурной, информационной и временной избыточности, позволяющих иметь дублирующие модули программ, альтернативные программы для решения одних и тех же задач, осуществлять контроль за процессом исполнения программ (контроль за заикливанием, возникновением самоблокировок, перегрузками по пропускной способности) и т. п.

В дальнейшем в настоящей главе будем рассматривать вопросы, относящиеся главным образом к безотказности программ.

3.2. ПРИЧИНЫ ОТКАЗОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Основными причинами, непосредственно вызывающими нарушения нормального функционирования программы, являются:

- ошибки, скрытые в самой программе;
- искажения входной информации, подлежащей обработке;
- неверные действия пользователя;
- неисправности аппаратуры установки, на которой реализуется вычислительный процесс.

Скрытые ошибки программы. Специфика создания сложных программных средств состоит в том, что в процессе их отладки практически невозможно обнаружить и ликвидировать все ошибки. В результате в программах остается некоторое количество скрытых ошибок. Они могут вызвать неверное функционирование программ при определенных сочетаниях входных данных. Наличие скрытых ошибок программного обеспечения является главным фактором нарушения нормальных условий его функционирования.

Можно выделить следующие основные классы ошибок в программах [66].

Ошибки вычислений. Ошибки данного класса содержатся в закодированных математических выражениях или в получаемых с их помощью результатах. Приме-

рами ошибок, относящихся к данному классу, являются неверное преобразование типов переменных, неверный знак операции, ошибка в выражении индекса, переполнение или потеря значимости при вычислениях.

Логические ошибки являются причиной искажения алгоритма решения задачи. Такого рода ошибки возникают в связи с неверной передачей управления, неверным заданием диапазона изменения параметра цикла, неверным условием и т. д.

Ошибки ввода-вывода, связанные с такими действиями, как управление вводом-выводом, формирование выходных записей, определение размеров записей и т. д. Примерами ошибок ввода-вывода являются неправильный формат ввода (вывода), ошибка в задании числа формируемых строк (страниц) при печати, отсутствие признака конца файла и т. д.

Ошибки манипулирования данными. Примерами таких ошибок являются неверно определенное число элементов данных; неверные начальные значения, присвоенные данным; неверно указанные длина операнда, имя переменной и т. д.

Ошибки совместимости связаны с отсутствием совместимости с операционной системой или другими прикладными программами, используемыми в данной программе.

Ошибки сопряжений вызывают неверное взаимодействие программы с другими программами (подпрограммами), с системными программами, устройствами ЭВМ, входными данными и т. д. В качестве примеров ошибок сопряжения могут быть названы несовместимость аргументов и параметров подпрограммы, отсутствие в системе необходимой подпрограммы, нарушение синхронизации при асинхронном выполнении программ и т. д.

Искажение информации, подлежащей обработке, вызывает нарушение функционирования программного обеспечения, когда входные данные не попадают в область допустимых значений переменных программы. В этом случае между исходной информацией и характеристиками программы возникает несоответствие.

Причинами искажения вводимой информации могут быть, например, следующие [38]:

- искажение данных на первичных носителях информации;

- сбои и отказы в аппаратуре ввода данных с первичных носителей информации;

шумы и сбои в каналах связи при передаче сообщений по линиям связи;

сбои и отказы в аппаратуре передачи или приема информации;

потери или искажения сообщений в буферных накопителях вычислительной системы;

ошибки в документации, используемой для подготовки вводимых данных;

ошибки пользователей при подготовке исходной информации.

Неверные действия пользователя, приводящие к отказу в процессе функционирования ПО, связаны прежде всего с неправильной интерпретацией сообщений, с неправильными действиями пользователя в процессе диалога с ЭВМ и т. д.

Отказы ПО, обусловленные ошибками пользователя, называются *ошибками использования*. Часто эти ошибки являются следствием некачественной программной документации (неверное описание возможностей программы, режимов работы, форматов входной и выходной информации, диагностических сообщений и т. д.).

Неисправность аппаратуры. Неисправности, возникающие при работе аппаратуры, используемой для реализации вычислительного процесса, оказывают определенное влияние на характеристику надежности ПО. Появление самоустраняющегося отказа или сбоя в работе аппаратуры приводит к нарушению нормального хода вычислительного процесса и во многих случаях — к искажению данных и текстов программ в основной и внешней памяти.

3.3. ПОСЛЕДСТВИЯ И ПРИЗНАКИ ПОЯВЛЕНИЯ ОШИБОК В ПРОГРАММЕ

Следствием ошибки в программе является ее отказ, заключающийся в отклонении от выполнения программой заданных функций. В зависимости от степени серьезности последствий ошибок (отказов) в программе эти отклонения можно разделить следующим образом:

полное прекращение выполнения функций на длительное или неопределенное время;

кратковременное нарушение хода вычислительного процесса.

Степень серьезности последствий ошибок в программе может быть оценена соотношением между длительностью восстановительных работ, которые необходимо произвести

после отказа в программе, и динамическими характеристиками объектов, использующих результаты работы программных средств. К таким характеристикам объектов относятся, например, инерционность объектов, выступающих в качестве источников и потребителей информации; заданная частота решения задач обработки информации; заданное время реакции вычислительной системы на запросы пользователей и др.

Наиболее типичными симптомами ошибок в программе являются [38]:

- преждевременное (аварийное) окончание выполнения программы;

- недопустимое увеличение времени выполнения программы;

- зацикливание ЭВМ на выполнении некоторой последовательности команд одной из программ;

- полная потеря или значительное искажение накопленных данных, необходимых для успешного выполнения решаемых задач;

- нарушение последовательности вызова отдельных программ, в результате чего происходит пропуск необходимых программ, либо непредусмотренное обращение к программам;

- искажение отдельных элементов данных (входных, выходных, промежуточных) в результате обработки искаженной исходной информации.

Аварийное завершение прикладных программ, как правило, легко идентифицируется, так как операционные системы обеспечивают возможность выдачи сообщений, содержащих соответствующий аварийный код (ABEND). Типичными причинами появления кодов аварийного завершения являются ошибки при выполнении макрокоманды; неверное использование методов доступа; нарушение защиты памяти; нехватка ресурсов памяти; неверное использование макрокоманды; возникновение программных прерываний, для которых не указан обработчик и др.

При появлении подобных ошибок после анализа аварийного кода имеется принципиальная возможность немедленного повторения запуска прикладной программы. Для увеличения эффективности восстановительных процедур необходимо:

- предусмотреть в программах специальные средства диагностики кодов аварийных завершений, в том числе кодов, формируемых самими пользователями;

- ввести в программы контрольные точки;

обеспечить возможности рестарта программ с контрольных точек.

Проявление ошибок в системном программном обеспечении имеет ряд специфических особенностей. Их анализ дан в § 16.6, в котором рассматриваются вопросы эксплуатации операционных систем.

3.4. АНАЛИТИЧЕСКИЕ МОДЕЛИ НАДЕЖНОСТИ ПРОГРАММ

Аналитические модели надежности дают возможность исследовать закономерности проявления ошибок в программах, а также прогнозировать надежность программ при разработке и эксплуатации. Модели надежности программ строятся на предположении, что проявление ошибки является случайным событием и поэтому имеет вероятностный характер. Такие модели предназначены для оценки показателей надежности программ и программных комплексов в процессе тестирования: количества ошибок, оставшихся невыявленными; времени, необходимого для выявления очередной ошибки в процессе эксплуатации программы; времени, необходимого для выявления всех ошибок с заданной вероятностью, и т. д. Модели дают возможность принять обоснованное решение о времени прекращения отладочных работ.

При построении моделей используются следующие характеристики надежности программы.

Функция надежности $P(t)$, определяемая как вероятность того, что ошибки программы не проявятся на интервале времени от 0 до t , т. е. время ее безотказной работы будет больше t .

Функция ненадежности $Q(t)$ — вероятность того, что в течение времени t произойдет отказ программы как результат проявления действия ошибки в программе. Таким образом,

$$Q(t) = 1 - P(t).$$

Интенсивность отказов $\lambda(t)$ — условная плотность вероятности времени до возникновения отказа программы при условии, что до момента t отказа не было. Как было показано в § 2.2,

$$\lambda(t) = - \frac{dP(t)}{dt} / P(t) = \left| \frac{dQ(t)}{dt} \right| / P(t).$$

Средняя наработка на отказ $\bar{T}_0$ — математическое ожидание временного интервала между последовательными отказами.

В настоящее время основными типами применяемых моделей надежности программ являются модели, основанные на предположении о дискретном изменении характеристик надежности программ в моменты устранения ошибок, и модели, основанные на экспоненциальном характере изменения числа ошибок в зависимости от времени тестирования и функционирования программы.

Модель надежности программ с дискретно-понижающейся частотой (интенсивностью) проявления ошибок [93.] В этой модели предполагается, что интенсивность обнаружения ошибок описывается кусочно-постоянной функцией, пропорциональной числу неустраненных ошибок.

Другими словами, предполагается, что интенсивность отказов $\lambda(t)$ постоянна до обнаружения и исправления ошибки, после чего она опять становится константой, но с другим, меньшим значением. При этом предполагается, что между $\lambda(t)$ и числом оставшихся в программе ошибок существует прямая зависимость:

$$\lambda(t) = k(M - i) = \lambda_i, \quad (3.1)$$

где M — неизвестное первоначальное число ошибок; i — число обнаруженных ошибок, зависящее от времени t ; K — некоторая константа (рис. 3.2).

Плотность распределения времени обнаружения i -й ошибки t_i задается соотношением

$$f(t_i) = \lambda_i e^{-\lambda_i t_i}.$$

Значение неизвестных параметров K и M может быть оценено на основании последовательности наблюдений интервалов между моментами обнаружения ошибок по методу максимального правдоподобия [66]. При этом для нахождения оценок параметров K и M необходимо решить следующие уравнения:

$$\sum_{i=1}^m (\bar{M} - 1)^{-1} = m/(\bar{M} - \theta_m);$$

$$\bar{k} = (m/A)/(\bar{N} - \theta_m),$$

$$\text{где } \theta_m = B/Am; \quad A = \sum_{i=1}^m t_i; \quad B = \sum_{i=1}^m it_i; \quad \bar{k} \text{ и } \bar{M}$$

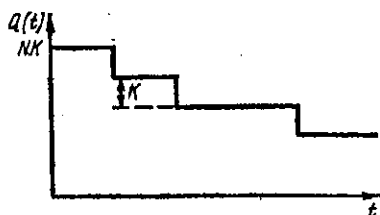


Рис. 3.2. Зависимость отказов программы от времени работы

— оценки соответственно k и M ; m — число устраненных ошибок в момент оценки надежности программ.

Рассмотренная модель надежности программ является достаточно грубой. На практике часто не соблюдаются условия, на которых она построена. Нередко при устранении ошибки вносятся новые ошибки. Во многих случаях не соблюдается также основное предположение, что при всяком устранении ошибки интенсивность отказов уменьшается на одну и ту же величину K . Не всегда удается определить и устранить причину отказа, и программу часто продолжают использовать, так как при других исходных данных ошибка, вызвавшая отказ, может себя и не проявить.

Модель надежности программ с дискретным увеличением времени наработки на отказ. В [17] предложена модель надежности программ, построенная на предположении, что устранение ошибки в программе приводит к увеличению времени наработки на отказ на одну и ту же случайную величину.

Предполагается, что время между двумя последовательными отказами $T^{(i)}$ является случайной величиной, которую можно представить в виде суммы двух случайных величин:

$$T^{(i)} = T^{(i-1)} + \Delta T^{(i)}, \quad (3.2)$$

где случайные величины $\Delta T^{(i)}$ независимы и имеют одинаковые математические ожидания $M[\Delta T]$ и средние квадратические отклонения $\sigma_{\Delta T}$.

Из (3.2) следует, что m -й отказ программы произойдет через время

$$T_m = \sum_{i=0}^m \left[T^{(0)} + \sum_{j=1}^i \Delta T^{(j)} \right].$$

Предполагается также, что $T^{(0)} \ll \sum_{j=1}^i \Delta T^{(j)}$. Основанием

для такого предположения является то, что отказы программы в начальном периоде эксплуатации возникают часто. В этом случае можно считать, что

$$T^{(0)} \approx \Delta T^{(0)}.$$

При этих предположениях средняя наработка между $(m-1)$ -м и m -м отказами программы равна

$$\bar{T}_m = M(T^{(m)}) = m M[\Delta T], \quad (3.3)$$

а средняя наработка до возникновения m -го отказа определяется выражением

$$\bar{T}_m = M(T_m) = \frac{m(m+1)}{2} M[\Delta T]. \quad (3.4)$$

Оценки величин $M[\Delta T]$, $\sigma_{\Delta T}$ могут быть получены по данным об отказах программы в течение периода наблюдения t_n следующим образом:

$$\bar{M}[\Delta T] = \frac{1}{m_n} \sum_{i=1}^{m_n} [t^{(i)} - t^{(i-1)}] = \frac{1}{m_n} \sum_{i=1}^{m_n} \Delta t^{(i)};$$

$$\sigma_{\Delta t}^2 = \frac{1}{m_n - 1} \sum_{i=1}^{m_n} (\Delta t^{(i)} - M[\Delta T])^2,$$

где m_n — число отказов программы за период $(0, t_n)$; t_i — момент возникновения i -го отказа программы.

Функция надежности определяется в зависимости от числа возникших отказов m :

$$P_m(t) = \frac{1}{2} - \Phi\left(\frac{t - mM[\Delta T]}{\sigma_{\Delta T}\sqrt{m}}\right), \quad (3.5)$$

где $\Phi(x)$ — функция Лапласа.

Экспоненциальная модель надежности программ [50]. Модель основана на предположении об экспоненциальном характере изменения во времени числа ошибок в программе.

В этой модели прогнозируется надежность программы на основе данных, полученных во время тестирования. В модели вводится суммарное время функционирования τ , которое отсчитывается от момента начала тестирования программы (с устранением обнаруженных ошибок) до контрольного момента, когда производится оценка надежности.

Предполагается, что все ошибки в программе независимы и проявляются в случайные моменты времени с постоянной средней интенсивностью в течение всего времени выполнения программы. Это означает, что число ошибок, имеющих в программе в данный момент, имеет пуассоновское распределение, а временной интервал между двумя отказами распределен по экспоненциальному закону, параметр которого изменяется после исправления ошибки.

Следует отметить, что характер закономерностей в этой модели остается таким же, как и в рассмотренных ранее,

а именно, интенсивность отказов предполагается пропорциональной числу оставшихся ошибок. Основное отличие данной модели от предыдущих состоит в том, что интенсивность отказов предполагается непрерывной функцией. Это упрощает математическое описание модели, а модель приобретает дополнительную гибкость.

Пусть M — число ошибок, имеющих в программе перед фазой тестирования (M рассматривается как некоторая константа); $m(\tau)$ — конечное число исправленных ошибок; $m_0(\tau)$ — число оставшихся ошибок. Тогда

$$m_0(\tau) = M - m(\tau). \quad (3.6)$$

При принятых предположениях интенсивность отказов пропорциональна $m_0(\tau)$, т. е.

$$\lambda(\tau) = K m_0(\tau), \quad (3.7)$$

где K — коэффициент пропорциональности, учитывающий реальное быстродействие ЭВМ и число команд в программе.

Введем дополнительное предположение, что в процессе корректировки новые ошибки не порождаются, т. е. что интенсивность исправления ошибок dm/dt будет равна интенсивности их обнаружения, т. е.

$$dm/d\tau = \lambda(\tau). \quad (3.8)$$

Решая совместно (3.7) и (3.8), получаем

$$(dm/d\tau) + Km = KM. \quad (3.9)$$

Перед началом работы ЭВМ ($t=0$) ни одна ошибка исправлена не была ($\tau=0$), поэтому решением уравнения (3.9) является

$$m = M[1 - \exp(-K\tau)]. \quad (3.10)$$

Будем характеризовать надежность программы после тестирования в течение времени τ средним временем наработки на отказ:

$$\bar{T}_0 = 1/\lambda(\tau). \quad (3.11)$$

Следовательно,

$$\bar{T}_0 = \frac{1}{KM} \exp(K\tau). \quad (3.12)$$

Введем $\bar{T}'_0$ — исходное значение среднего времени наработки на отказ перед тестированием. Тогда

$$\bar{T}'_0 = 1/KM. \quad (3.13)$$

В результате имеем

$$\bar{T}_0 = \bar{T}'_0 \exp(\tau/MT'_0). \quad (3.14)$$

Очевидно, что среднее время наработки на отказ увеличивается по мере выявления и исправления ошибок.

Рассмотрим более общий случай, когда в процессе корректировки могут появляться новые ошибки. Пусть B — коэффициент уменьшения ошибок, определяемый как отношение интенсивности уменьшения ошибок к интенсивности их проявления, т. е. к интенсивности отказов.

Пусть n — число обнаруженных отказов, а N_0 — число отказов, которые должны произойти, чтобы можно было выявить и устранить m соответствующих ошибок, т. е.

$$n = m/B, \quad N_0 = M/B.$$

В этом случае среднее время наработки на отказ $\bar{T}_0$ и число обнаруженных отказов n определяются следующими соотношениями:

$$\bar{T}_0 = \frac{1}{BK M_0} \exp(BK\tau); \quad (3.15)$$

$$n = M_0[1 - \exp(-BK\tau)]. \quad (3.16)$$

Тогда для значения среднего времени наработки на отказ перед тестированием справедливо

$$\bar{T}'_0 = 1/BKN_0. \quad (3.17)$$

В результате получаем

$$\bar{T}_0 = \bar{T}'_0 \exp(\tau/N_0 \bar{T}'_0); \quad (3.18)$$

$$n = N_0[1 - \exp(-\tau/N_0 \bar{T}'_0)]. \quad (3.19)$$

Для практического использования представляет интерес число ошибок Δn , которое должно быть обнаружено и исправлено для того, чтобы добиться увеличения среднего времени наработки на отказ от $\bar{T}_{01}$ до $\bar{T}_{02}$. Этот показатель может быть получен из следующих соотношений:

$$n = N_0(1 - \bar{T}'_0/\bar{T}_0); \quad (3.20)$$

$$\Delta n = n_2 - n_1. \quad (3.21)$$

Следовательно,

$$\Delta N = N_0 \bar{T}'_0 [1/\bar{T}_{01} - 1/\bar{T}_{02}]. \quad (3.22)$$

Дополнительное время работы, необходимое для обеспечения увеличения среднего времени наработки на отказ с $\bar{T}_{01}$ до $\bar{T}_{02}$, определяется как

$$\Delta \bar{T} = M_0 \bar{T}'_0 \ln \left(\frac{\bar{T}_{02}}{\bar{T}_{01}} \right). \quad (3.23)$$

Рассмотрим, каким образом можно оценить основные параметры модели. Коэффициент B может быть определен исходя из статистических данных по числу ошибок, порожденных во время обнаружения других ошибок. Можно считать, что для систем программного обеспечения общего назначения B изменяется в диапазоне 0,91—0,95 [88].

Для оценки M и M_0 можно рассчитать среднюю интенсивность ошибок (число ошибок на одну команду) в начале различных фаз тестирования. По оценкам, данным в [40, 66, 81], в начале процесса тестирования для программ, написанных на языке Ассемблера, число ошибок на 1000 команд варьируется от 4 до 8.

Рассмотренная модель может применяться для определения времени испытаний программ с целью достижения заданного уровня надежности, а также для оценки числа оставшихся в программе ошибок.

Модель надежности больших программных комплексов

Для прогноза надежности больших программных комплексов может быть использована марковская модель [86]. Надежность всего программного комплекса определяется как функция надежности ее составных частей. Подобная оценка значительно облегчается, если программа строится по модульному принципу.

Мерой надежности программного модуля будем считать вероятность того, что модуль выполняет возложенную на него функцию корректно, т. е. выдает корректные выходные данные и корректно передает управление следующему модулю. Каждому набору входных данных, поступающих на вход программы, соответствует некоторая последовательность выполняемых модулей. Следовательно, надежность программного комплекса будет зависеть от последовательности выполняемых модулей и надежности каждого из этих модулей.

Будем считать, что надежности модулей — величины независимые. Общий результат работы программы будет некорректным, если хотя бы в одном из выполненных в данном прогоне программы модуле была ошибка. Кроме того,

будем считать, что надежность каждого модуля может быть некоторым образом определена.

Предположим также, что управление модулями программы может быть представлено как марковский процесс, т. е. выбор следующего для выполнения модуля зависит только от модуля, выполняемого в данный момент, и не зависит от предыстории. Будем также предполагать, что вероятности передачи управления между модулями — величины постоянные и полностью характеризуют способ использования программы пользователем.

Представим структуру управления программы в виде направленного графа, в котором каждая вершина N_i ($i = \overline{1, n}$) соответствует некоторому программному модулю, а дуга (N_i, N_j) — возможной передаче управления от N_i к N_j . Каждой дуге (N_i, N_j) соотнесем величину вероятности перехода из вершины N_i по дуге (N_i, N_j) , а каждой вершине N_i — величину R_i — вероятность безотказной работы соответствующего модуля. Можно считать, что граф программы имеет единственную входную N_1 , выходную N_n и две дополнительные вершины, соответствующие корректному C и ошибочному F выходным результатам.

Будем считать, что при появлении ошибки осуществляется переход в состояние F по дуге (N_i, F) с вероятностью $1 - R_i$ независимо от правильности последующей обработки. Если модуль N_i выдает правильный результат, то осуществляется переход к выполнению модуля N_j с вероятностью $R_i P_{ij}$. Переход из выходного состояния N_n в состояние C соответствует корректному завершению программы и происходит с вероятностью R_n .

Таким образом, матрица переходных вероятностей P элементы которой P_{ij} соответствуют переходу марковского процесса из состояния i в состояние j , имеет следующий вид:

$$P = \begin{matrix} & \begin{matrix} N_1 & N_2 & \dots & N_j & \dots & N_n & C & F \end{matrix} \\ \begin{matrix} N_1 \\ \vdots \\ N_i \\ \vdots \\ N_{n-1} \\ N_n \\ C \\ F \end{matrix} & \left| \begin{array}{ccccccc} 0 & R_1 P_{12} & \dots & R_1 P_{1j} & \dots & R_1 P_{1n} & 0 & 1 - R_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & R_i P_{i2} & \dots & R_i P_{ij} & \dots & R_i P_{in} & 0 & 1 - R_i \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & R_{n-1} P_{(n-1)2} & \dots & R_{n-1} P_{(n-1)j} & \dots & R_{n-1} P_{(n-1)n} & 0 & 1 - R_{n-1} \\ 0 & 0 & \dots & 0 & \dots & 0 & R_n & 1 - R_n \\ 0 & 0 & \dots & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 & 0 & 1 \end{array} \right| \end{matrix}$$

Надежность всего программного комплекса может быть вычислена как вероятность достижения конечного состояния S из начального состояния графа. Для этого рассмотрим матрицу Q , которая получена из матрицы P , после вычеркивания столбцов и строк, соответствующих конечным состояниям S и F :

$$Q = \begin{matrix} & \begin{matrix} N_1 & N_2 & \dots & N_j & \dots & N_n \end{matrix} \\ \begin{matrix} N_1 \\ \vdots \\ N_i \\ \vdots \\ N_{n-1} \\ N_n \end{matrix} & \left| \begin{array}{cccccc} 0 & R_1 P_{12} & \dots & R_1 P_{1j} & \dots & R_1 P_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & R_i P_{i2} & \dots & R_i P_{ij} & \dots & R_i P_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & R_{n-1} P_{(n-1)2} & \dots & R_{n-1} P_{(n-1)j} & \dots & R_{n-1} P_{(n-1)n} \\ 0 & 0 & \dots & 0 & \dots & 0 \end{array} \right| \end{matrix} \quad (3.25)$$

Для каждого целого $k > 0$ определим P^k как k -ю степень P . Очевидно, $P^k(i, j)$ — это вероятность того, что марковский процесс за k шагов перейдет из состояния i в состояние j . Тогда матрица

$$T = I + P + P^2 + \dots = \sum_{k=0}^{\infty} P^k \quad (3.26)$$

определяет вероятности перехода из одного состояния в другое за произвольное число шагов.

Пусть S — квадратная матрица размерности n такая, что

$$S = I + Q + Q^2 + \dots = \sum_{k=0}^{\infty} Q^k \quad (3.27)$$

Положим $W = I - Q$, тогда согласно [4] имеем

$$S = W^{-1} = (I - Q)^{-1}, \quad (3.28)$$

откуда надежность программного комплекса

$$R = S(1, n) R_n. \quad (3.29)$$

На рис. 3.3 в качестве примера изображен граф, отражающий структуру программного комплекса, содержащего семь модулей, где N_1 — входной модуль, а N_7 — выходной. Надежность каждого из модулей N_i программы задается константой R_i .

Пусть, например,

$$R_1 = 0,999; \quad R_2 = 0,970; \quad R_3 = 0,980; \quad R_4 = 0,995; \\ R_5 = 0,985; \quad R_6 = 0,950; \quad R_7 = 0,980.$$

Пусть также известны вероятности P_{ij} передачи управления между модулями N_i и N_j :

$$P_{12} = 0,6; \quad P_{13} = 0,4; \quad P_{23} = 0,7; \quad P_{24} = 0,3; \\ P_{36} = 1,0; \quad P_{45} = 0,4; \quad P_{46} = 0,6; \quad P_{52} = 0,2; \\ P_{56} = 0,2; \quad P_{57} = 0,6; \quad P_{67} = 1,0.$$

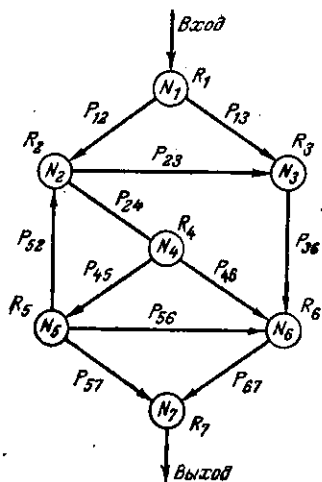


Рис. 3.3. Пример графа программного комплекса

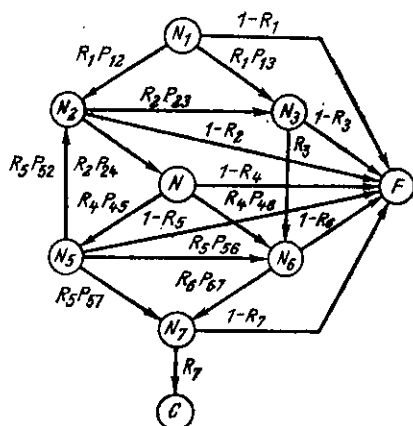


Рис. 3.4. Марковская модель надежности программного комплекса

Граф модели надежности программного комплекса представлен на рис. 3.4. Матрица P переходных вероятностей выглядит следующим образом:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,001 & 0 & 0,599 & 0,400 & 0 & 0 & 0 & 0 \\ 0 & 0,02 & 0 & 0 & 0,685 & 0,295 & 0 & 0 & 0 \\ 0 & 0,02 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0,005 & 0 & 0 & 0 & 0 & 0,398 & 0,597 & 0 \\ 0 & 0,015 & 0 & 0,197 & 0 & 0 & 0 & 0,197 & 0,591 \\ 0 & 0,05 & 0 & 0 & 0 & 0 & 0 & 0 & 0,950 \\ 0,980 & 0,02 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Матрица Q имеет вид

$$Q = \begin{bmatrix} 0 & 0,599 & 0,400 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,685 & 0,295 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0,398 & 0,597 & 0 \\ 0 & 0,197 & 0 & 0 & 0 & 0,197 & 0,591 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,950 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$S(1,7) = [I - Q]^{-1}(1,7) = 0,938;$$

$$R = S(1,7) R_7 = 0,938 \cdot 0,980 = 0,919.$$

Таким образом, надежность программного комплекса в целом оценивается значением 0,919.

Глава четвертая

СИСТЕМОТЕХНИЧЕСКИЕ АСПЕКТЫ ЭКСПЛУАТАЦИОННОГО ОБСЛУЖИВАНИЯ ЭВМ

4.1. ИЕРАРХИЯ ПРОЦЕССОВ ОБСЛУЖИВАНИЯ.

ФОРМЫ ОРГАНИЗАЦИИ ЭКСПЛУАТАЦИОННОГО ОБСЛУЖИВАНИЯ

Широкий масштаб внедрения электронной вычислительной техники в различные области народного хозяйства, науки и культуры, большое число различных ЭВМ, эксплуатируемых в составе ВЦ, различных автоматизированных систем управления и обработки данных, систем автоматизации проектирования, автоматизации эксперимента и испытаний сложных объектов, систем автоматизации управления технологическими процессами (АСУ ТП) делает исключительно важной проблему эффективного эксплуатационного обслуживания технических и программных средств ЭВМ.

Большая стоимость оборудования ЭВМ, их высокая производительность обуславливают значительные экономические потери при простое ЭВМ из-за неисправностей или плохо организованной эксплуатации.

Сложность обслуживания ЭВМ связана с большим числом и разнообразием входящих в их состав компонентов и устройств, сложностью логической организации машин, большим объемом технической документации.

Сложность обслуживания усугубляется тем, что вычислительная система представляет собой единый комплекс

взаимодействующих компонентов разной природы: людей — операторов и пользователей, аппаратных, микропрограммных и программных (операционные системы) средств.

Устройства системы работают параллельно во времени, причем часть асинхронно относительно друг друга. Имеющаяся в настоящее время тенденция к использованию таких прогрессивных режимов работы ЭВМ, как режимы коллективного пользования, дополнительно усложняют обслуживание системы.

Несмотря на все это обслуживание технических и программных средств ЭВМ должно быть своевременным и качественным.

Сложность ЭВМ как объекта обслуживания, быстрая смена моделей ЭВМ и версий программного обеспечения, непрерывное появление в составе ЭВМ новых типов устройств и новых пакетов прикладных программ в трудно преодолеваемых условиях запаздывания и недостаточной полноты эксплуатационной документации предъявляют высокие требования к квалификации, самостоятельности и творческим способностям персонала, обслуживающего ЭВМ.

Новые проблемы в обслуживании средств вычислительной техники возникли в связи с широким распространением и рассредоточенностью микропроцессорной и микромашинной техники.

Если не принять специальных мер, организационных и технических, то значительный уровень затрат на обслуживание ЭВМ — трудовых и материальных (запасные части, помещения и оборудование для ремонта) — при большом числе эксплуатируемых ЭВМ существенно снизит общую эффективность использования электронной вычислительной техники.

Эксплуатация ЭВМ предполагает реализацию следующей иерархически структурированной совокупности процессов обслуживания (рис. 4.1).

Подготовительный этап включает в себя выбор (исходя из потребностей организации-пользователя) конфигурации ЭВМ (состава технических средств) и состава системы программного обеспечения (типа операционной системы и т. п.), проектирование ВЦ, размещение в нем оборудования систем обеспечения определенных требований к среде, в которой будет работать аппаратура ЭВМ. К этому же этапу относится установка ЭВМ в ВЦ, наладка, испытание и сдача ЭВМ в эксплуатацию.

Этап собственно эксплуатации предполагает следующие процессы обслуживания:

планово-профилактические работы, проводимые в определенные сроки (ежедневно, еженедельно, ежемесячно и т. д.);

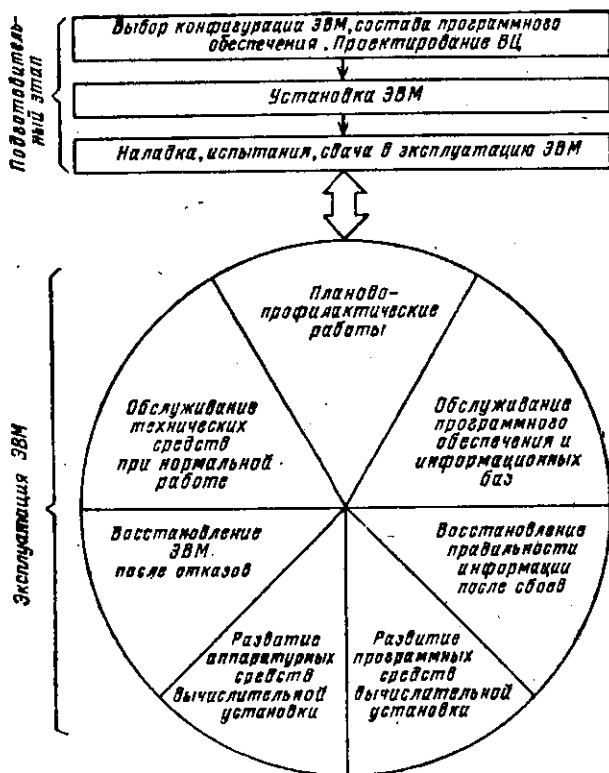


Рис. 4.1. Иерархия процессов обслуживания ЭВМ

обслуживание технических средств при нормальной работе, включая обслуживание процесса подготовки и решения задачи на ЭВМ;

обслуживание программного обеспечения и информационных баз при нормальных условиях работы;

восстановление правильности информации (программ и данных) после сбоев;

восстановление ЭВМ после отказов (ремонт);

развитие аппаратурных средств эксплуатируемой вычислительной установки — ввод в состав вычислительной установки дополнительных технических средств (дополнительные блоки памяти, модули дисковых ЗУ, новые ЗУ большей емкости, дисплей и т. п.);

развитие программных средств вычислительной установки — ввод в эксплуатацию новых версий операционной системы, пакетов программ, расширяющих функции операционной системы, новых пакетов прикладных программ.

Для уменьшения потерь времени и затрат труда на такие виды обслуживания, как планово-профилактические работы, восстановление вычислительного процесса (и правильности информации) после сбоев, восстановление работоспособности ЭВМ после отказов, машины снабжают комплексом аппаратурно-программных средств поддержки процессов обслуживания.

Однако даже при наличии таких средств быстро увеличивающееся число территориально рассредоточенных вычислительных установок (различных ВЦ, АСУ, АСУ ТП) сопровождается ростом числа специалистов — инженеров и техников, занятых эксплуатацией ЭВМ. Это заставляет настойчиво искать и внедрять наиболее эффективные формы организации эксплуатационного обслуживания ЭВМ.

Можно указать следующие формы эксплуатационного обслуживания ЭВМ: индивидуальное, централизованное, централизованное дистанционное, смешанное.

Индивидуальное (локальное, или децентрализованное) обслуживание предполагает наличие при каждой вычислительной установке (в каждом ВЦ) штата специалистов по эксплуатации технических и программных средств ЭВМ, достаточного по количеству и квалификации для выполнения всех процессов обслуживания, указанных на рис. 4.1.

Такая форма обслуживания помимо больших людских затрат, невозможности эффективно использовать персонал порождает трудности с обеспечением вычислительных установок запасными деталями, узлами, достаточно подробной эксплуатационной документацией, созданием условий для систематического обучения и повышения квалификации специалистов, эксплуатирующих ЭВМ. Сказанное является причиной перехода к другим формам обслуживания ЭВМ, в первую очередь к централизованному обслуживанию, а также к промежуточной форме — смешанному обслуживанию, когда одни виды обслуживания реализуются в форме централизованного обслуживания, а другие — в форме индивидуального.

4.2. КОМПЛЕКСНОЕ ЦЕНТРАЛИЗОВАННОЕ ОБСЛУЖИВАНИЕ ЭВМ

Прогрессивной формой организации эксплуатации ЭВМ — является комплексное централизованное обслуживание (КЦО) технических и программных средств вычислительной техники (СВТ).

Централизованное обслуживание предполагает наличие территориальных (региональных, районных, городских и т. п.) центров обслуживания, оказывающих услуги по обслуживанию и эксплуатации ЭВМ вычислительным центрам и другим организациям — владельцам машин, расположенным в территориальной зоне действия обслуживающего центра. Центры централизованного обслуживания располагают штатом квалифицированных специалистов по эксплуатации ЭВМ, необходимой документацией, запасными деталями и узлами, специализированным оборудованием.

Под комплексным обслуживанием понимают выполнение всего комплекса работ, связанных с обеспечением эксплуатации ЭВМ, включая проектирование ВЦ, установку и наладку машин, техническое обслуживание и ремонт ЭВМ, сопровождение программного обеспечения, обучение эксплуатационного персонала организаций-пользователей.

Целью создания системы КЦО является повышение эффективности использования ЭВМ в народном хозяйстве. Для выполнения этой задачи система КЦО представляет ВЦ и другим организациям — владельцам ЭВМ услуги в следующих трех направлениях: обслуживание технических средств; обслуживание средств программного обеспечения; обучение и повышение квалификации персонала пользователей в области эксплуатации технических и программных средств ЭВМ.

Более подробный перечень услуг, предоставляемых системой КЦО, включает в себя следующие [61, 67]:

- проектирование ВЦ, в том числе выбор конфигурации вычислительной установки и состава системы программного обеспечения, проектирование размещения средств вычислительной техники, средств обеспечения необходимых параметров среды в помещениях ВЦ;

- монтаж, ввод в эксплуатацию и гарантийный ремонт устройств ЭВМ;

- централизованное обслуживание СВТ, включая регламентные профилактические работы, ремонт по вызову, аварийный ремонт, централизованный ремонт блоков и устройств;

доработку устройств по документации организации-разработчика;

поставку, генерацию, ввод в эксплуатацию и сопровождение новых версий операционных систем;

поставку, ввод в эксплуатацию и сопровождение пакетов прикладных программ, банков данных и программных средств расширения функций операционных систем;

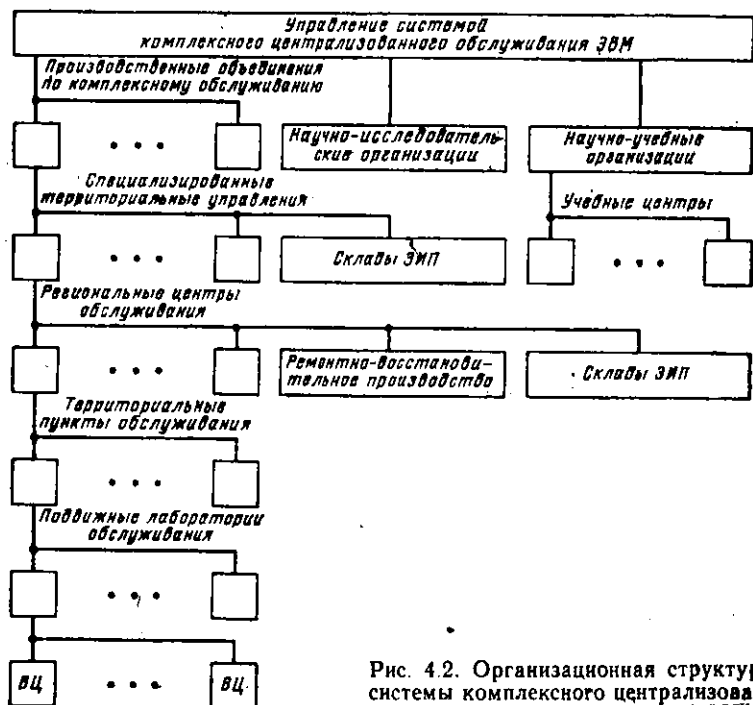


Рис. 4.2. Организационная структура системы комплексного централизованного обслуживания ЭВМ

подготовку и переподготовку в специальных учебных центрах специалистов по техническим и программным средствам ЭВМ.

Структура созданной в стране системы КЦО показана на рис. 4.2. Обслуживание ЭВМ непосредственно выполняют территориальные пункты обслуживания (ТПО), располагающие подвижными лабораториями обслуживания (ПЛО). В пределах определенных регионов ТПО объединяются в региональные центры обслуживания (РЦО), которые в свою очередь вместе с ремонтно-восстановитель-

ными заводами и складами ЗИП объединяются в специализированные территориальные управления. Следующий уровень организационной структуры образован производственными объединениями (ПО). К этому же уровню относятся специальные научно-производственные объединения, располагающие учебными центрами. Система возглавляется Всесоюзным объединением.

В настоящее время осуществляются три вида централизованного обслуживания:

полное обслуживание, включая проведение всех планово-профилактических ремонтов и ремонта по вызову. При полном обслуживании аварийный ремонт начинается не позже 3,5 ч после получения вызова. Этот вид обслуживания практикуется в отношении вычислительных установок, расположенных в пределах города, в котором находится ТПО;

частичное обслуживание с выполнением только ежемесячных планово-профилактических ремонтов, ремонта по вызову и аварийного. Практикуется в отношении вычислительных установок, находящихся на расстоянии до 100 км от ТПО;

обслуживание по вызову с выполнением аварийного ремонта в сроки, оговоренные с пользователем. Применяется для вычислительных установок, находящихся на расстоянии более 100 км от ТПО.

Система комплексного централизованного обслуживания предоставляет пользователю определенные гарантии. Так, например, гарантируются продолжительность ввода в эксплуатацию ЭВМ (не свыше 20—45 и 45—80 дней соответственно для ЭВМ средней и высокой производительности) и устранение неисправностей (за 3,5—24 ч).

Опыт внедрения централизованной организации обслуживания показал, что эта форма обслуживания приводит к сокращению численности эксплуатационного персонала ВЦ, увеличению среднего времени суточной полезной работы ЭВМ, позволяет уменьшить общие фонды запасных частей.

В настоящее время системой централизованного обслуживания охвачено свыше 50 % всех машин Единой системы.

4.3. СИСТЕМА ДИСТАНЦИОННОГО ОБСЛУЖИВАНИЯ ЭВМ

Заметим, что осуществляемый системой КЦО вид обслуживания зависит от расстояния пользователя ЭВМ от территориального центра (пункта). Такая зависимость снижает гибкость и эффективность системы обслуживания.

ния. Действительно, КЦО со структурой, подобной изображенной на рис. 4.2, является крупнейшим шагом в сторону централизации по сравнению с индивидуальным обслуживанием. Однако и при такой структуре обслуживание оказывается рассредоточенным между многочисленными территориальными пунктами обслуживания и передвижными лабораториями. В этом случае остаются определенные трудности в укомплектовании сети обслуживания высококвалифицированным персоналом, способным разрешать наиболее сложные ситуации, которые могут возникать в процессе эксплуатации ЭВМ. Остается и сложность в обеспечении сети обслуживания (ТПО и передвижных лабораторий) подробной технической документацией, схемами всех узлов и устройств.

Между тем успехи, достигнутые в области систем передачи по линиям связи дискретной информации, реализации режимов телеобработки данных на ЭВМ, подсказывают новый путь в построении централизованных систем обслуживания — дистанционное обслуживание.

Дистанционное обслуживание является системой обслуживания, в которой обслуживаемые ЭВМ связаны с территориально удаленным от них центром обслуживания линиями и аппаратурой передачи данных, с помощью которых при осуществлении процессов обслуживания из центра обслуживания поступают на ЭВМ задания на выполнение определенных программ, тестовых процедур, а в центр обслуживания выдаются реакции ЭВМ на тестовые процедуры. В центре обслуживания эти реакции анализируются или автоматически вычислительным центром комплекса, или с привлечением самого высококвалифицированного персонала и подробной технической документации. При необходимости из центра задаются дополнительные проверки. В результате центр вырабатывает поступающее в обслуживаемый ВЦ решение о замене некоторого блока, изменении регулировки и т. п. или о высылке из центра в ВЦ необходимого блока для замены.

В результате дистанционно-контролируемых периодических профилактических испытаний в центре обслуживания концентрируется в систематизированном виде информация о состоянии прикрепленной на обслуживание к центру ЭВМ. На этой основе возможно прогнозирование состояния ЭВМ и принятие мер, предупреждающих аварийные отказы.

Концентрация в одном или немногих центрах обслуживания (точнее, в их вычислительных системах) информа-

ции о поведении в эксплуатации выпущенных промышленностью машин (виды отказов, их частоты и др.) позволяет интенсифицировать процесс конструктивных доработок ЭВМ, направленный на повышение их качества и в первую очередь надежности работы.

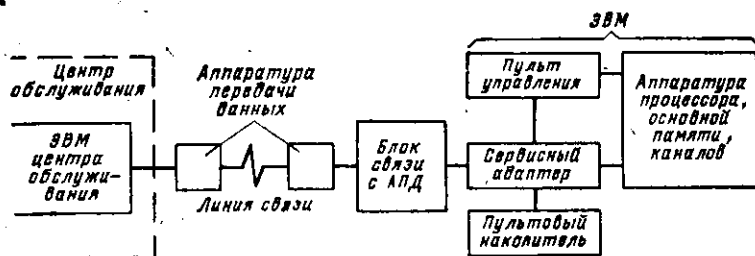


Рис. 4.3. Система дистанционного обслуживания ЭВМ

Техническая сторона реализации системы дистанционного обслуживания поясняется схемой, изображенной на рис. 4.3. Построение подобной системы облегчается, если ЭВМ (как показано на рисунке и как это имеет место, например, в ЕС-1045) имеет логически и конструктивно обособленное оборудование для реализации процедур диагностики неисправностей. На рис. 4.3 это оборудование представлено в виде сервисного адаптера (узкоспециализированного процессора) и пультового дискового или ленточного накопителя. Сервисный адаптер связывает пультовый накопитель, хранящий диагностические тесты, с пультом управления и аппаратурой ЭВМ. При индивидуальном местном обслуживании сервисный адаптер управляет процессом диагностирования, в том числе вводом в аппаратуру ЭВМ с пультового накопителя диагностических тестов, съемом с пульта управления реакций ЭВМ на тесты, сопоставлением их с эталонами, выдачей данных в протокол диагностирования.

При дистанционном обслуживании сервисный адаптер (а через него пультовый накопитель и пульт управления) через блок связи и аппаратуру передачи данных соединяется с ЭВМ центра обслуживания. Таким образом обеспечивается возможность дистанционно из удаленного центра обслуживания задавать режимы испытаний обслуживаемой ЭВМ, приводить в действие систему автоматической диагностики и получать в центр обслуживания реакции

ЭВМ на различные тесты для выработки решений по обеспечению работоспособного состояния ЭВМ.

Широкое развитие методов построения, технических и программных средств сетей ЭВМ, основанных на использовании систем передачи данных по линиям связи, облегчает развитие систем дистанционного обслуживания ЭВМ.

4.4. РЕЖИМЫ РАБОТЫ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И СИСТЕМ И ПРОБЛЕМЫ ИХ НАДЕЖНОСТИ И ОБСЛУЖИВАЕМОСТИ

Большая стоимость и высокая номинальная производительность современных вычислительных машин и систем делает весьма актуальной задачу их эффективного использования. Это естественным образом порождает стремление к реализации на вычислительных установках наиболее эффективных для пользователя режимов работы.

На рис. 4.4 показана иерархия режимов работы вычислительных машин. Режимы работы перечисляются в порядке возрастания эффективности использования вычислительных средств с точки зрения интересов пользователя.

Наиболее простой режим — режим индивидуального пользования, при котором машина выполняет одну программу, а пользователь (программист), работая за пультом управления машиной (или выполняющей его функции электрифицированной пишущей машинкой или дисплеем),

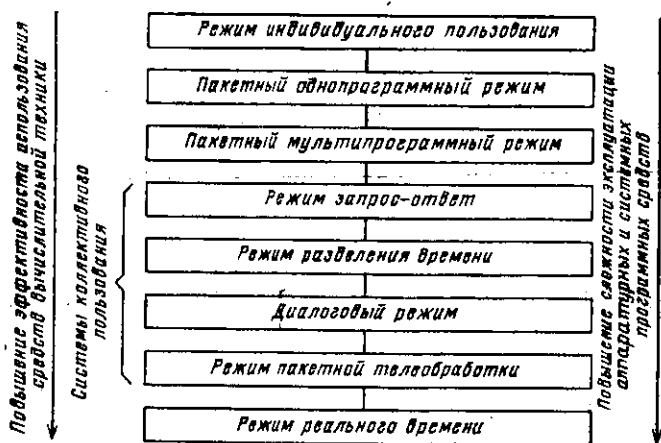


Рис. 4.4. Иерархия режимов работы вычислительной машины

осуществляет процедуры ручного управления вычислительным процессом, ведет отладку своей программы, обдумывает обнаруживающиеся ошибки, вносит поправки в программу. Режим этот очень удобен для программиста, но монополизация средств машины одним программистом и одной программой обуславливает длительные простои оборудования, когда пользователь обдумывает, какие изменения надо внести в его программу, и, кроме того, простои процессора в ожидании выполнения операции ввода-вывода.

Более производителен пакетный режим, при котором задачи нескольких пользователей собираются в «пакет», который под наблюдением оператора обрабатывается в однопользовательном, а еще лучше в мультипользовательном режиме. Здесь исключены простои машины из-за обдумывания решений пользователем. Обнаружение ошибки в программе приводит к фиксации ошибки и снятию программы с обработки. Мультипрограммирование позволяет в максимальной степени загрузить наиболее производительную часть оборудования — процессор. Если обрабатываемая программа доходит до места, когда необходим обмен информацией между основной памятью и периферийными устройствами (например, внешним запоминающим устройством на дисках или лентах), обработка данной программы на процессоре приостанавливается (выполняется для этой программы операция ввода-вывода) и производится обработка другой программы из пакета, пока она тоже не дойдет до места ввода-вывода, и т. д.

В пакетном режиме уменьшаются простои оборудования ЭВМ. Однако характерная для этого режима потеря пользователем непосредственного контакта с машиной (пользователь общается с машиной через оператора) приводит к значительному снижению производительности труда лиц, связанных с подготовкой и решением задач на ЭВМ.

Контакт пользователя с ЭВМ возобновляется на новой основе в системах коллективного пользования, когда ЭВМ обслуживает одновременно несколько десятков, сотен или тысяч пользователей, причем за счет разницы в скорости реакции машины и человека для каждого из пользователей мало заметно, что машина одновременно обслуживает других лиц. Режим коллективного пользования предполагает предоставление каждому пользователю индивидуальных аппаратных средств (терминального оборудования) для связи с ЭВМ. В системах коллектив-

ного пользования возникает очередь запросов на обработку.

Простейшим из режимов коллективного пользования является режим «запрос-ответ». Этот режим применяют, когда выставляемые пользователями запросы требуют примерно одинакового времени обработки. Машина обрабатывает запросы в порядке их поступления. Начав обработку запроса, она завершает его до конца и выдает пользователю ответ на запрос.

Режим «запрос-ответ» мало подходит для систем, в которых запросы пользователей могут значительно отличаться по требуемому времени работы машины. Поскольку режим коллективного пользования вводится для создания ощущения контакта пользователя с машиной в интересах повышения эффективности труда человека, особенно недопустимы длительные задержки обработки коротких запросов пользователей. Пользователь психологически подготовлен к сравнительно длительному ожиданию ответа на запрос, требующий, по его мнению, довольно много машинного времени.

Учет этих обстоятельств лежит в основе режима разделения времени, при котором не практикуется непрерывная обработка запроса до конца, а время работы процессора делится на отдельные небольшие интервалы — кванты, представляемые запросам из очереди в порядке, определяемом соответствующей дисциплиной обслуживания. Если выделенного кванта оказывается недостаточно для завершения запроса, то его обработка прерывается и он возвращается в очередь. Разделение времени (квантованное обслуживание) имеет целью уменьшить время ответа системы на короткие запросы.

В режиме разделения времени восстанавливается непосредственный контакт пользователя с ЭВМ и пользователю предоставляется возможность активно вести отладку программы и вмешиваться в процесс выполнения программы.

Наиболее эффективным с точки зрения производительности системы «ЭВМ — пользователи» является диалоговый режим, в котором обеспечивается тесное общение человека с машиной, позволяющее человеку и машине совместно решать задачу (так называемый интерактивный режим) и распределять функции таким образом, чтобы оптимально использовались возможности человека и машины. Интерактивный (диалоговый) режим решения задачи и отладки программ требует развитых аппаратурных

средств общения человека с машиной в виде интеллектуальных дисплейных терминалов со световым пером, графических дисплеев, расчерчивающих устройств, цветных дисплеев, и т. п. и создания на основе диалоговых языков (БЭИСИК, АПЛ и др.) специальных систем программирования и системных программных средств поддержки процессов взаимодействия человека с машиной.

Большие удобства для пользователя представляет режим телеобработки, при котором удаленный от ЭВМ пользователь может инициировать обработку своих программ на ЭВМ и получать результаты обработки, будучи соединенным с ЭВМ линиями связи и аппаратурой передачи данных.

Дальнейшим развитием систем телеобработки являются сети ЭВМ — совокупность распределенных по значительной территории вычислительных машин, связанных между собой аппаратурой и линиями передачи данных. Такая система позволяет пользователю привлечь для решения своих задач аппаратные, программные и информационные ресурсы не только своего, но и других вычислительных центров сети ЭВМ.

Перечень режимов завершает режим реального времени, при котором ЭВМ связана с помощью специального оборудования с некоторым технологическим процессом и осуществляет обработку поступающей от процесса информации в темпе развития самого процесса, с тем чтобы результаты обработки информации на ЭВМ могли быть использованы для воздействия на процесс в нужном направлении.

Режимы разделения времени и диалоговый могут сопровождаться фоновой обработкой данных в пакетном режиме.

Переход к более эффективным режимам связан с усложнением аппаратных средств, операционной системы и других системных программ машины, усложнением процедур распределения ресурсов между пользовательскими программами и процедур взаимодействия аппаратных и программных средств между собой и пользователем. Значительно увеличивается объем информации, которая должна быть в любой момент доступна машине.

В результате значительно осложняется обеспечение надежности, обслуживание и восстановление систем после сбоев и отказов. Увеличивается число ошибок, порождаемых несовершенством системных программных средств, в

частности из-за трудности учета нюансов в процедурах взаимодействия аппаратурных и программных средств.

При этом увеличивается доля наиболее неприятных ошибок в работе машины, приводящих к заикливанию или зависанию, после которых требуется перезагрузка операционной системы. Для анализа таких ошибок необходимы опытные специалисты в области как программных, так и аппаратурных средств, поэтому повышаются требования к квалификации эксплуатационного персонала.

К сожалению, на практике при проведении работ по развитию вычислительных установок и по переходу к более прогрессивным режимам работы часто игнорируется положение о том, что такой переход требует значительного возрастания надежности функционирования аппаратурно-программного комплекса системы. В противном случае вместо повышения может получиться даже уменьшение эффективности вычислительной установки.

В пакетном режиме пользователи разобщены с вычислительной системой и не знают о происходящих нарушениях ее функционирования, которые могут только отодвинуть (относительно регламентированного) срок получения из ВЦ результатов обработки программ.

В режимах коллективного пользования отказ в работе системы нарушает трудовой процесс сразу большого числа пользователей, которые к тому же чаще всего остаются не осведомленными о причинах происшедшего и о сроках восстановления, что помимо всего психологически обоснованно вызывает у пользователей раздражение и недоверие к системе.

В настоящее время вопросы связи режимов работы вычислительной машины с надежностью функционирования вычислительных установок и особенностями их обслуживания недостаточно исследованы.

4.5. СИСТЕМНЫЙ ПОДХОД ПРИ РАЗРАБОТКЕ КОНЦЕПЦИИ И АППАРАТУРНО-ПРОГРАММНЫХ СРЕДСТВ ОБСЛУЖИВАНИЯ ЭВМ

По мере развития электронной вычислительной техники обостряется проблема эксплуатационного обслуживания ЭВМ. Этому способствует непрерывное усложнение логической организации ЭВМ, повышение степени интеграции электронных схем, высокая плотность монтажа электронного оборудования, наличие в эксплуатации боль-

шого числа машин, рассредоточенных по территории всей страны.

В настоящее время приспособленность к обслуживанию (*обслуживаемость*) следует рассматривать как одну из основных характеристик машины, непосредственно влияющих на эксплуатационную надежность ЭВМ. Обслуживаемость ЭВМ, как отмечалось ранее, тем выше, чем меньше требования к квалификации и затратам труда эксплуатирующего машину персонала.

Ремонтопригодность (среднее время продолжительности ремонта) является важным, но не единственным показателем обслуживаемости. К другим показателям обслуживаемости следует отнести затраты времени и труда на профилактические испытания, восстановление достоверности информации после сбоев.

Разработка концепции и отдельных вопросов обслуживания должна входить составной и неотъемлемой частью в общий процесс проектирования ЭВМ. Создание концепции обслуживания требует системного подхода, учитывающего во взаимодействии организационные, технические, программные и, конечно, экономические стороны этого процесса. Системный подход должен обосновать, исходя из заданных эксплуатационных характеристик ЭВМ и минимизации затрат на эксплуатацию, решения в отношении организации системы обслуживания, архитектурных особенностей ЭВМ, состава и функций аппаратурно-программных средств поддержки эксплуатации.

При разработке концепции обслуживания следует учитывать, что в настоящее время и в дальнейшем будет практиковаться смешанное централизованное и индивидуальное (децентрализованное) обслуживание. В этом случае и создаваемая технология и в особенности средства поддержки обслуживания должны учитывать потребности как централизованного, так и индивидуального обслуживания.

В настоящее время общепринятым является представление о том, что все сколько-нибудь сложные ЭВМ должны снабжаться комплексом аппаратурно-программных средств поддержки эксплуатации, органически включенных в архитектуру ЭВМ.

Чтобы определить, какие архитектурные свойства и аппаратурно-программные средства должны иметь ЭВМ для обеспечения высокой эффективности эксплуатации машины, следует проанализировать приведенное в гл. 1 выражение (1.9) для комплексного коэффициента эксплуата-

ционной надежности (комплексного коэффициента использования).

Знаменатель в (1.9) есть полное время наблюдений за эксплуатацией машины, равное сумме общей продолжительности работоспособного состояния ЭВМ $\sum_{i=1}^n t_i$ и суммарной продолжительности восстановлений после отказов $\sum_{j=1}^n \tau_{в.о.}$.

В числителе из общего времени работоспособного состояния вычитаются суммарное время, затраченное на контроль правильности работы машины τ_k , суммарные потери времени на восстановление достоверности информа-

ций после сбоев $\sum_{r=1}^m \tau_{в.с.}$ и затраты времени на профилакти-

ческие испытания $\sum_{s=1}^k \tau_{п.ф.}$. Выражение (1.9) определяет

долю полезного времени работы ЭВМ.

Пользователь должен быть уверен в правильности результатов вычислений и обработки данных на ЭВМ. Если при проектировании не принять специальных мер для обеспечения автоматического контроля правильности работы машины, то осуществлять такой контроль придется пользователю и он будет им производиться путем двойного счета или выполнения контрольных вариантов расчета, что приведет к потере фактической производительности машины вследствие увеличения члена τ_k в числителе формулы (1.9). Вместе с тем сбой или отказ в машине приводит к искажению информации (данных или программы), и если не принять специальных мер, такое искажение может привести к распространению ошибки и в итоге к искажению программ и данных. В результате существенно увеличивается время восстановления достоверности информации, что необходимо для продолжения решения задачи после сбоя.

Для освобождения пользователя от забот по контролю правильности работы ЭВМ, недопущения распространений последствий сбоев (отказов) на программы и данные ЭВМ следует снабжать *системой автоматического контроля*, воспринимающей ошибку ЭВМ практически в момент ее возникновения и препятствующей дальнейшему выполнению программы.

Для сокращения потерь времени на восстановление после сбоя работы ЭВМ (восстановление достоверности информации) следует иметь *автоматическую систему восстановления вычислительного процесса*. На эту систему может быть возложено автоматическое повторение неправильно выполненной операции (в разных случаях повторению может подлежать микрокоманда, команда или участок программы) и при безошибочном исходе повторения — восстановление вычислительного процесса.

Системы автоматического контроля и восстановления обеспечивают высокую контролепригодность ЭВМ.

Из (1.9) следует, что для повышения комплексного коэффициента использования следует, с одной стороны, добиваться увеличения среднего времени наработки машины на отказ, а с другой стороны, повышать ремонтпригодность ЭВМ (уменьшать среднюю продолжительность восстановления после отказа) и снижать время простоя в ожидании начала ремонта. Принимаемые для этого технические и организационные мероприятия должны быть обоснованы экономически.

Так, например, значительное уменьшение времени простоя в ожидании ремонта может быть достигнуто при индивидуальном (децентрализованном) обслуживании, если все рабочие смены ЭВМ укомплектовать ремонтными бригадами специалистов и обеспечить наличие полного ЗИП. Однако такой подход во многих случаях экономически не оправдан, так как экономический эффект от уменьшения времени простоя может оказаться меньше расходов, связанных с содержанием при ЭВМ большого штата специалистов по эксплуатации машины и полного ЗИП. Решение этого и других подобных вопросов требует системного подхода, учитывающего взаимодействие технических, организационных и экономических сторон эксплуатации машин.

Процесс восстановления работоспособности при отказе состоит из следующих процедур:

- 1) диагностирования неисправности — поиска неисправности с точностью до функционального устройства или блока, группы съемных элементов;
- 2) локализации неисправности — определения неисправного съемного элемента;
- 3) замены или ремонта неисправного элемента;
- 4) проверки работоспособности машины после ремонта или замены неисправного элемента.

Основное время в процессе восстановления работоспособности занимает поиск неисправности. Эти процедуры в

среднем занимают около 60 % общего времени восстановления и требуют высокой квалификации обслуживающего персонала. Поэтому важнейшим средством повышения ремонтпригодности является оснащение ЭВМ средствами *автоматического диагностирования*.

Наличие системы автоматического диагностирования не только облегчает и ускоряет этап поиска неисправности, но и сокращает время и труд, затрачиваемые на этапе проверки работоспособности машины после ремонта (замены) неисправного элемента.

Определение стратегии профилактических работ (периодичности и глубины профилактик) также требует системного подхода. Сокращение промежутков времени между профилактиками, увеличение объема профилактических испытаний естественно будет способствовать уменьшению вероятности появления отказов и сбоев в период между профилактиками. Однако проведение профилактических испытаний связано с непроизводительными затратами рабочего времени ЭВМ, а также с затратами труда на эти испытания.

Для повышения проверкопригодности (уменьшения затрат времени и труда на профилактические испытания) ЭВМ оборудуют *системой автоматической программно-управляемой профилактики*, которая обеспечивает автоматическое изменение уровней вторичных питающих напряжений, частоты и формы управляющих сигналов и автоматическую регистрацию результатов профилактических испытаний, в том числе выдачу сообщений об элементах и узлах, параметры которых находятся вблизи критических значений.

Для повышения эксплуатационной надежности вычислительной установки, принятия обоснованных и своевременных решений о производстве доработок, замене элементов и узлов, в которых развиваются постепенные отказы, важное значение имеет накопление и обработка информации об имевших место отказах и сбоях и о местах их возникновения. Эта задача решается автоматической *аппаратурно-программной системой регистрации и обработки данных об отказах и сбоях*.

Встроенные в ЭВМ системы автоматического контроля, диагностирования и профилактики используются также при заводской и пусковой наладках, восстановлении после ремонта, в результате чего резко уменьшается время, затрачиваемое на эти процедуры.

Перечисленные выше автоматические системы, поддерживающие процессы обслуживания, существенно сокращают затраты на них времени и труда. Учитывая сложность и компактность аппаратуры современных ЭВМ, правильнее будет сказать, что указанные выше автоматические системы делают обслуживание ЭВМ осуществимым. Без этих систем эксплуатация современных ЭВМ была бы практически невозможной.

Значение вопросов обслуживания ЭВМ при их эксплуатации настолько возросло для современных ЭВМ средней и большой производительности, что стало ясно, что разработка концепции обслуживания ЭВМ должна являться органической составной частью начального этапа проектирования ЭВМ, когда определяются основные особенности архитектуры ЭВМ. Исходными данными для разработки концепции обслуживания являются эксплуатационные характеристики ЭВМ, такие как средние времена наработки на отказ и сбой, средние времена восстановления, коэффициенты использования и эксплуатационной готовности.

Концепция обслуживания включает в себя представление о форме организации обслуживания (централизованное, индивидуальное, смешанное), составе и характеристиках аппаратурно-программных средств поддержки обслуживания, таких как системы автоматического контроля, диагностирования и профилактики, системы регистрации и накопления информации об особых событиях (сбоях, отказах), результатах профилактики и в целом поведении машины в процессе эксплуатации. Концепция обслуживания определяет также функции обслуживающего персонала при нормальной эксплуатации, профилактических испытаниях, планово-предупредительных ремонтах, взаимодействие персонала с указанными выше системами поддержки обслуживания и взаимодействие этих поддерживающих систем между собой.

В области методологии проектирования ЭВМ современная концепция обслуживания требует, чтобы аппаратурно-программные средства поддержки обслуживания (ранее упоминавшиеся системы автоматического контроля, восстановления вычислительного процесса, диагностирования профилактических испытаний, учета и регистрации событий при эксплуатации ЭВМ, а также рассматриваемые в последующих главах системы реконфигурации, параллельного тестирования периферийных устройств) проектировались параллельно с разработкой аппаратурных и системных программных средств ЭВМ.

На рис. 4.5 показано взаимодействие основных аппаратурно-программных средств, предназначенных для повышения эксплуатационной надежности ЭВМ и поддержки эксплуатационного обслуживания машины. Эти средства обеспечивают обнаружение, локализацию и устранение неисправностей в ЭВМ.

Система автоматического контроля ЭВМ обнаруживает ошибки с вероятностью p_1 . Если общее число ошибок за

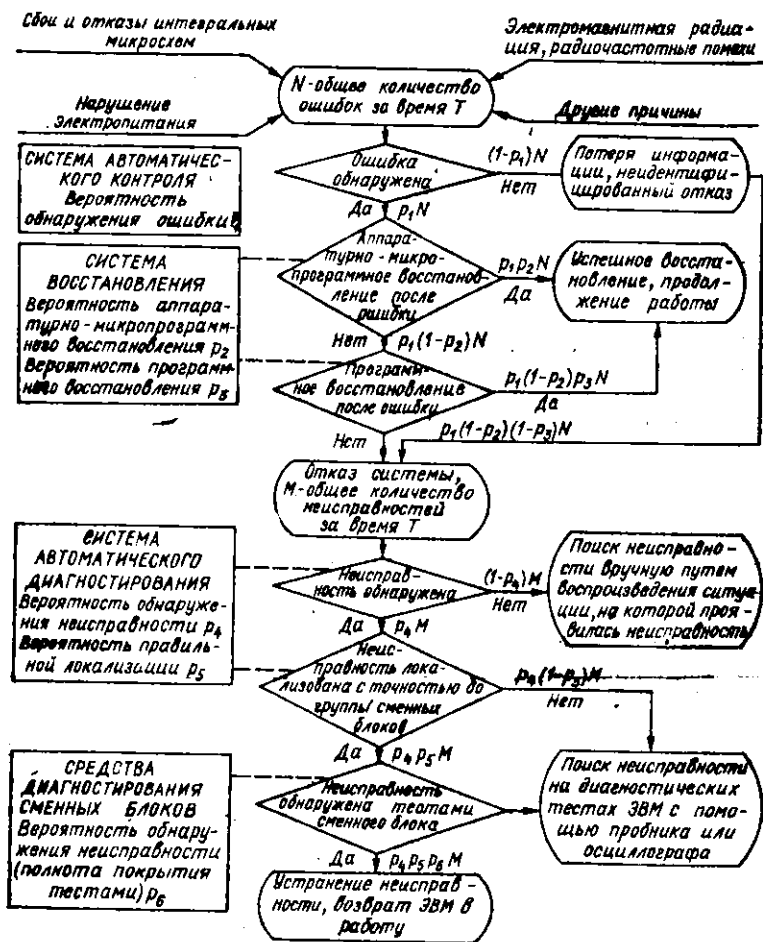


Рис. 4.5. Взаимодействие аппаратурно-программных средств поддержки технического обслуживания

время T равно N , то система автоматического контроля обнаруживает $p_1 N$ ошибок. Ошибки, необнаруженные системой контроля, в количестве $(1-p_1)N$ вызывают потерю информации и неидентифицированные отказы.

Система восстановления ЭВМ обеспечивает восстановление вычислительного процесса после ошибок; содержит средства аппаратурно-микропрограммного и программного восстановления и характеризуется, соответственно, вероятностью аппаратурно-микропрограммного p_2 и программного p_3 восстановления вычислительного процесса после ошибок.

Система восстановления обрабатывает ошибки, обнаруженные системой контроля, и восстанавливает вычисления после $p_1 N [p_2 + (1-p_2)p_3]$ ошибок из общего их числа N .

Остальные ошибки, соответствующие отказам в аппаратуре машины, локализуются с помощью системы автоматического диагностирования.

Система автоматического диагностирования служит для локализации неисправностей и характеризуется вероятностью обнаружения неисправностей p_4 , вероятностью правильной локализации p_5 , точностью (разрешающей способностью) и продолжительностью диагностирования.

В том случае, когда система автоматического диагностирования не обнаруживает неисправность, последняя локализуется вручную, что приводит к значительному увеличению времени восстановления ЭВМ. Если общее число неисправностей (отказов) за время T равно M , то число необнаруженных неисправностей равно $(1-p_4)M$. Время восстановления ЭВМ также существенно увеличивается и в том случае, когда неисправность неверно локализуется системой автоматического диагностирования. Число таких неисправностей составляет $p_4(1-p_5)M$.

Обычно система автоматического диагностирования локализует неисправности (их число равно $p_4 p_5 M$) с точностью до группы сменных блоков.

Дальнейшая локализация неисправности до элемента сменного блока выполняется на ручных или автоматизированных стендах проверки сменных блоков.

Средства диагностирования сменных блоков обеспечивают локализацию неисправностей до элемента сменного блока и характеризуются вероятностью обнаружения неисправности p_6 и продолжительностью диагностирования.

Вероятность обнаружения p_6 определяется качеством тестов, т. е. полнотой охвата проверяемых схем тестами.

Хотя она и зависит от качества системы синтеза тестов сменных блоков, однако существуют ограничения, не позволяющие получить 100 %-ное «покрытие» тестами сложных сменных блоков, содержащих схемы с памятью и обратными связями. В связи с этим большое значение имеет проектирование *тестопригодных* сменных блоков, позволяющих получить охват тестами электронных компонентов блока, близкий к 100 %, путем введения дополнительных контрольных точек и других схемных решений.

4.6. ТРЕБОВАНИЯ К ЭКСПЛУАТАЦИОННОЙ ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ

Практически не удается построить системы диагностирования, которые определяли бы все возможные неисправности и локализовали их местоположение настолько точно, что обслуживающему персоналу оставалось бы только механическое выполнение примитивных операций по замене неисправного элемента. К тому же существуют неисправности, например замыкания и обрывы проводников, исчезновение отдельных питающих напряжений, которые не удастся диагностировать. В подобных случаях, но часто и при отыскании (локализации) конкретного места неисправности по информации, выдаваемой системой диагностики, обслуживающий персонал должен выполнять довольно сложную работу, требующую высокой квалификации. Здесь многое зависит от качества технической документации на ЭВМ и ее приспособленности для ведения работ по восстановлению работоспособности машины после отказов.

При поставке ЭВМ пользователю передается эксплуатационная техническая документация на ЭВМ и системное программное обеспечение. Объем технической документации, на основе которой ведется изготовление и отладка ЭВМ, огромен. Большой объем имеет и описательная часть документации (технические описания). Технические описания, получаемые пользователем, в основном содержат описания структур и процессов функционирования машины и ее отдельных частей.

В меньшей степени получаемые пользователем технические описания ориентированы на техническое обслуживание машин. Более того, в настоящее время недостаточно изучен вопрос, каким требованиям должна удовлетворять эксплуатационная документация для облегчения обслуживания и в особенности ремонта машины.

Поэтому ограничимся перечислением лишь некоторых требований к текстовой документации (техническим описаниям) [72]:

структурные схемы должны показывать работу структурных элементов и схем управления в реальных режимах работы;

структурные схемы должны показывать направления передачи информации в различных режимах работы;

описания алгоритмов работы должны быть подробны и обеспечивать связь с символической микропрограммой и электрической схемой;

символический язык микропрограммирования должен использовать обозначения элементарных операций в соответствии со структурной схемой;

должны иметься отдельные структурные схемы системы контроля с обозначением адресов схемных элементов;

функциональные схемы съемных блоков должны содержать наименования входных, выходных сигналов, регистров, триггеров, счетчиков и другой аппаратуры. Желательно указывать на функциональных схемах источники входных и приемники выходных сигналов;

инструкции по эксплуатации ЭВМ и периферийных устройств должны содержать конструктивные адреса взаимозаменяемых сменных блоков, что позволяет путем соответствующей перестановки быстро подтвердить правильность локализации неисправности.

В периферийных устройствах основная доля неисправностей приходится на электромеханические узлы. Поэтому инструкции по эксплуатации ПУ должны содержать диаграммы поиска неисправностей в электромеханических узлах, а также разнообразный графический материал, иллюстрирующий процедуры съема, ремонта и установки механических узлов.

Для тех устройств ЭВМ, которые имеют средства автоматического диагностирования, поиск неисправности ведется с помощью тестов и диагностических словарей. Словари, в общем случае, содержат коды неисправностей и соответствующие им списки блоков, подозреваемых на неисправность. Основными требованиями, предъявляемыми к диагностическим словарям, является высокая достоверность и точность локализации неисправностей.

АВТОМАТИЧЕСКИЙ КОНТРОЛЬ ПРАВИЛЬНОСТИ ФУНКЦИОНИРОВАНИЯ ЭВМ В ПРОЦЕССЕ НОРМАЛЬНОЙ ЭКСПЛУАТАЦИИ

5.1. ФУНКЦИИ И ХАРАКТЕРИСТИКИ СИСТЕМЫ КОНТРОЛЯ

В предыдущей главе было показано, что для уменьшения потерь от сбоев и отказов, порождающих ошибки, надо предотвратить распространение ошибки в вычислительном процессе, так как в противном случае существенно усложняются и удлиняются процедуры проверки правильности работы программы, определения и устранения искажений в программе, данных и промежуточных результатах.

Для этого необходимо обнаруживать появление ошибки в выполняемых машиной преобразованиях информации возможно ближе к моменту ее возникновения. С этой целью надо иметь систему автоматического контроля правильности работы ЭВМ, которая при появлении ошибки в работе машины немедленно приостанавливает вычисления и производит диагностирование характера ошибки, с тем чтобы при сбое автоматически восстанавливались достоверность информации и выполнение программы и при этом был минимален повторяемый участок программы, а при отказе обслуживающий персонал был извещен о необходимости ремонта машины.

Наличие такой системы освобождает пользователя от забот о контроле правильности выполняемых машиной операций, что существенно при работе ЭВМ в системах реального времени, управляющих технологическими процессами. В противном случае пользователь вынужден включить в свои программы процедуры двойного счета, просчет контрольных вариантов, что усложняет программирование задач пользователя и приводит к непроизводительным затратам машинного времени.

Для повышения комплексного коэффициента эксплуатационной надежности необходимо повышать обслуживаемость машины и добиваться уменьшения потерь времени на устранение отказов (повышение ремонтпригодности) и проведение профилактических работ. Эти потери времени в таких сложных объектах, как ЭВМ, в первую очередь связаны с поиском места неисправности. Важнейшим средством уменьшения указанных потерь и повышения обслужи-

ваемости ЭВМ является система автоматического диагностирования, позволяющая локализовать неисправность.

Система диагностирования использует имеющуюся в машине систему контроля, и поэтому трудно провести между ними четкую границу.

Система контроля и диагностирования современных ЭВМ представляет собой сочетание аппаратурных, микропрограммных и программных средств.

Обнаружение ошибок должно производиться в машине непрерывно и, следовательно, не должно вызывать заметного снижения ее быстродействия. Поэтому эта функция возлагается обычно на быстродействующие аппаратурные средства контроля, которые позволяют почти полностью совместить во времени выполнение основных и контрольных операций.

Необходимость коррекции ошибок и диагностирования неисправностей при современном уровне надежности ЭВМ возникает достаточно редко. Поэтому целесообразно использовать для выполнения этих функций главным образом микропрограммные и программные средства в виде корректирующих и диагностических микропрограмм и программ. Однако чтобы эти программы и микропрограммы не были чрезмерно сложны, предусматриваются определенные аппаратурные средства, которые предоставляют в распоряжение микропрограмм и программ необходимую информацию о характере ошибки.

Электронно-вычислительную машину можно рассматривать как преобразователь информации, а реализуемый в ней вычислительный процесс — как процесс преобразования информации [21]. С этой точки зрения любые выполняемые ЭВМ операции могут быть сведены к следующим трем классам: передача информации, логические преобразования, арифметические преобразования.

При передаче информации кодовое (машинное) слово передается в пространстве (из одного узла машины в другой) или во времени (запоминание и хранение слова в запоминающем устройстве). Передача информации и хранение, являются взаимно-однозначными преобразованиями, при которых входное слово, поступающее на устройство или схему передачи (хранения) информации, и выходное слово на выходе устройства (схемы) совпадают.

Логическое преобразование в общем случае состоит в формировании по некоторым правилам из k входных двоичных слов длины n одного выходного слова

той же длины. При этом двоичный символ в i -м разряде выходного слова зависит только от значений символов в i -разрядах и не зависит от значений символов в других разрядах входных слов.

В ЭВМ обычно выполняются логические преобразования (операции) при $k=2$ (сложение по модулю два, логическое ИЛИ, логическое И и др.) и при $k=1$ (инверсия).

В арифметических преобразованиях (операциях) из двух входных слов, задающих числовые операнды, вырабатывается в соответствии с алгоритмами арифметических операций результирующее выходное слово, причем значение двоичного символа в i -м разряде выходного слова определенным образом зависит от значений символов как в i -м, так и в других разрядах входных слов.

Из сказанного следует, что для построения автоматического контроля правильности выполняемых ЭВМ преобразований информации достаточно иметь схемы контроля правильности передачи информации и правильности логических и арифметических операций.

В ряде случаев при реализации систем автоматического контроля правильности работы ЭВМ не ограничиваются только контролем достоверности выполняемых машиной операций, а применяют также в том или ином объеме средства автоматической коррекции обнаруживаемых ошибок.

В основе построения автоматического контроля правильности работы ЭВМ (и коррекции ошибок) лежит принцип избыточности, предполагающий использование той или иной избыточности (временной, информационной, аппаратурной, алгоритмической) по сравнению с устройствами, работающими без контроля и коррекции ошибок.

Временная избыточность предполагает дополнительные затраты времени на выполнение контрольных операций. Примером является решение задачи методом двойного счета со сравнением получаемых результатов.

Информационная избыточность проявляется в использовании для представления команд и данных в ЭВМ кодов с дополнительными разрядами, используемыми в процедурах контроля и коррекции ошибок.

Аппаратурная избыточность состоит в применении дополнительной аппаратуры для реализации контроля и коррекции ошибок, например двух арифметическо-логических устройств, выполняющих параллельно во времени одни и те же операции со сравнением получаемых результатов.

Алгоритмическая избыточность предполагает выполнение решения задачи по разным алгоритмам (программам) с проверкой получаемых результатов на совпадение.

На практике системы автоматического контроля правильности работы ЭВМ строятся в основном на использовании информационной избыточности в сочетании с элементами избыточностей других типов.

Использование схем контроля правильности работы ЭВМ порождает проблему контроля правильности работы самих схем контроля. В последнее время эту проблему пытаются решать путем построения таких схем контроля, которые обладают свойством самоконтроля правильности своего процесса функционирования.

В современных ЭВМ общего назначения средства автоматического контроля обеспечивают контроль правильности функционирования до 90—95 % оборудования машины. При этом за рамками автоматического аппаратного контроля оказываются некоторые цепи управления, приема и передачи информации, синхронизации и сигнализации. Попытки дальнейшего увеличения доли контролируемого аппаратурными средствами оборудования наталкиваются на практически неоправданное возрастание объема аппаратуры автоматического контроля. Не охватываемые аппаратурным контролем цепи контролируются с помощью специальных контролирующих (тестовых) программ при профилактических испытаниях или в форме фоновой задачи при обработке пользовательских программ.

Важной характеристикой системы автоматического контроля является ее время реакции на ошибку, измеряемое числом тактов синхронизации между моментом возникновения ошибки и формированием сигнала прерывания от схем контроля (блокированием тактовых синхросигналов).

Казалось бы, что в идеальном случае следовало контролировать выполнение каждой микрооперации и переходить к выполнению следующей микрокоманды только после получения сигнала о правильности выполнения предыдущей микрокоманды. Однако при таком подходе производительность ЭВМ существенно уменьшается, причем без достаточных оснований, так как ошибки в работе ЭВМ возникают сравнительно редко, особенно если их соотнести с числом команд и микрокоманд, выполняемых в интервале между ошибками машины. На практике процедуры контроля реализуются во времени с небольшим отставанием параллельно с развертыванием процесса выполнения

команд программы. В результате реакция на возникновение ошибки задерживается на один-два такта синхронизации.

Основными характеристиками системы автоматического контроля правильности функционирования ЭВМ являются: доля оборудования ЭВМ, охваченного системой контроля; вероятность обнаружения системой контроля ошибок в функционировании ЭВМ; степень детализации, с которой система контроля указывает место возникновения ошибки¹; отношение количества оборудования системы контроля к общему объему оборудования ЭВМ; время реакции системы контроля на ошибку.

5.2. КОНТРОЛЬ ПЕРЕДАЧИ ИНФОРМАЦИИ

При контроле передачи информации наибольшее распространение получили методы информационной избыточности, использующие коды с обнаружением и коррекцией ошибок.

Если длина кода n разрядов, то таким двоичным кодом можно представить максимум 2^n различных слов. Если все разряды слова служат для представления информации, код называется простым (неизбыточным). Коды, в которых только часть кодовых слов используется для представления информации, называются избыточными. Часть слов в избыточных кодах является запрещенными, и их появление при передаче информации свидетельствует о наличии ошибки. Принадлежность слова к разрешенным или запрещенным словам определяется правилами кодирования, и для различных кодов эти правила различны.

Коды разделяются на равномерные и неравномерные. В равномерных кодах все слова содержат одинаковое число разрядов. В неравномерных кодах число разрядов в словах может быть различным. В вычислительных машинах применяются преимущественно равномерные коды.

Равномерные избыточные коды делятся на делимые и неразделимые. Делимые коды всегда содержат постоянное число информационных (представляющих передаваемую информацию) и избыточных разрядов, причем избыточные разряды занимают одни и те же позиции в кодовом слове. В неразделимых кодах разряды кодового

¹ Детальное распознавание места неисправности является задачей системы диагностирования (см. гл. 8).

слова невозможно разделить на информационные и избыточные.

Способность кода обнаруживать или исправлять ошибки определяется так называемым минимальным кодовым расстоянием. *Кодовым расстоянием* между двумя словами называется число разрядов, в которых символы слов не совпадают. Если длина слова n , то кодовое расстояние может принимать значения от 1 до n . *Минимальным кодовым расстоянием* данного кода называется минимальное расстояние между двумя любыми словами в этом коде. Если имеется хотя бы одна пара слов, отличающихся друг от друга только в одном разряде, то минимальное расстояние данного кода равно 1.

Простой (неизбыточный) код имеет минимальное расстояние $d_{min}=1$. Для разделимых избыточных кодов $d_{min}>1$. Если $d_{min}\geq 2$, то любые два слова в данном коде отличаются не менее чем в двух разрядах, следовательно, любая одиночная ошибка приведет к появлению запрещенного слова и может быть обнаружена. Если $d_{min}\geq 3$, то любая одиночная ошибка создает запрещенное слово, отличающееся от правильного в одном разряде, а от любого другого разрешенного слова — не менее чем в двух разрядах. Заменяя запрещенное слово ближайшим к нему (в смысле кодового расстояния) разрешенным словом, можно исправить одиночную ошибку.

В общем случае, чтобы избыточный код позволял обнаруживать ошибки кратностью r , должно выполняться условие

$$d_{min} \geq r + 1. \quad (5.1)$$

Действительно, одновременная ошибка в r разрядах слова создает новое слово, отстоящее от первого на расстоянии r . Чтобы оно не совпало с каким-либо другим разрешенным словом, минимальное расстояние между двумя разрешенными словами должно быть хотя бы на единицу больше, чем r .

Для исправления r -кратной ошибки необходимо, чтобы новое слово, полученное в результате ошибки, не только не совпадало с каким-либо разрешенным словом, но и оставалось ближе к правильному слову, чем к любому другому разрешенному слову. От правильного слова новое отстоит на расстоянии r . Следовательно, от любого другого разрешенного слова оно должно отстоять не менее чем на $r+1$, а минимальное кодовое расстояние должно быть не меньше суммы этих величин:

$$d_{min} \geq 2r + 1. \quad (5.2)$$

Код с проверкой четности образуется путем добавления к группе информационных разрядов, представляющих собой простой (неизбыточный) код, одного избыточного (контрольного) разряда.

При формировании кода слова в контрольный разряд записывается 0 или 1 таким образом, чтобы сумма единиц в слове, включая избыточный разряд, была четной (при контроле по четности) или нечетной (при контроле по нечетности). В дальнейшем при всех передачах (включая запись в память и считывание) слово передается вместе со своим контрольным разрядом. Если при передаче информации приемное устройство обнаруживает, что в принятом слове значение контрольного разряда не соответствует четности суммы единиц слова, то это воспринимается как признак ошибки.

Минимальное расстояние кода $d_{min}=2$, поэтому код с проверкой четности (нечетности) обнаруживает все одиночные ошибки, а кроме того, все случаи нечетного числа ошибок (3, 5 и т. д.). При одновременном возникновении двух или любого другого четного числа ошибок код с проверкой четности (нечетности) не обнаруживает ошибок.

При контроле по нечетности контролируется полное пропадание информации, так как кодовое слово, состоящее из 0, относится к запрещенным.

Код с проверкой четности имеет небольшую избыточность и не требует больших затрат оборудования на реализацию контроля. Этот код широко применяется в вычислительных машинах для контроля передач информации между регистрами и для контроля считываемой информации в оперативной памяти.

При построении схем определения четности суммы единиц слова используют двухразрядные логические элементы с парафазным выходом, подобные изображенным на рис. 5.1, а и б. Показанные схемы выполняют операцию сложения по модулю 2 (условное обозначение $M2$) для двоичных переменных x и y . На рис. 5.1, в показана схема определения признака четности байта.

Каждый информационный символ должен быть задан прямым и инверсным кодом. Структура схемы проверки четности является многоступенчатой, т. е. слово делится на несколько групп разрядов, в каждой из которых проверка четности производится прямым способом (первая ступень), далее производится проверка четности для групп второй ступени, образованных из групп первой ступени, четности которых в этом случае рассматриваются как

обычные двоичные разряды, и т. д. до окончательной проверки четности суммы единиц всего слова. В последней ступени четность байта сравнивается со значением контрольного разряда (КР).

Легко установить связь кодирования при контроле по четности с выполнением сложения по модулю 2. Если число единиц в слове должно быть четным, то в контрольный

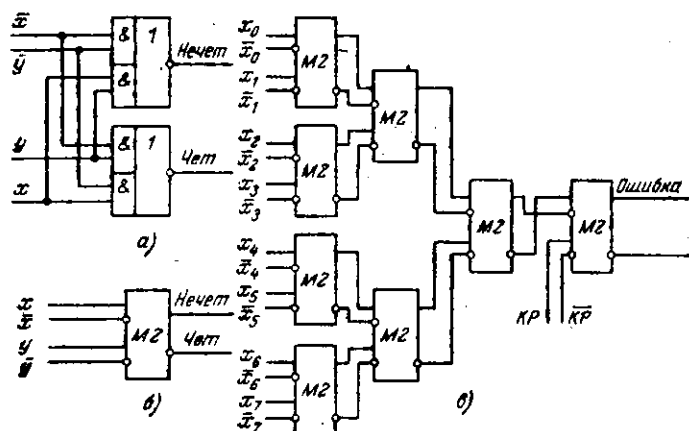


Рис. 5.1. Схема определения четности

разряд записывается прямой код суммы по модулю 2 всех информационных разрядов слова. При контроле на нечетность в контрольный разряд заносится обратный код указанной суммы.

Контроль по совпадению. При передачах можно использовать и другой вид контроля — контроль по совпадению. После передачи информации из одного регистра в другой правильность передачи можно проверить путем поразрядного сравнения содержимого всех разрядов регистров.

При контроле по совпадению не требуется формирования каких-либо дополнительных контрольных разрядов, следовательно, этот метод основывается не на информационной, а на схемной избыточности.

Один из вариантов схемы контроля передач по совпадению показан на рис. 5.2. После передачи информации из регистра А в регистр В (или из В в А) через время, несколько большее времени установления переходных процессов в триггерах регистров, на выходе схемы появляется

сигнал 1 при несовпадении содержимого A и B и 0 в противном случае. При контроле передачи по совпадению обнаруживаются ошибки любой кратности.

Затраты оборудования при контроле по совпадению меньше, чем при контроле по четности. Контроль по совпадению является быстродействующим, так как используется одна ступень формирования. В схеме проверки четно-

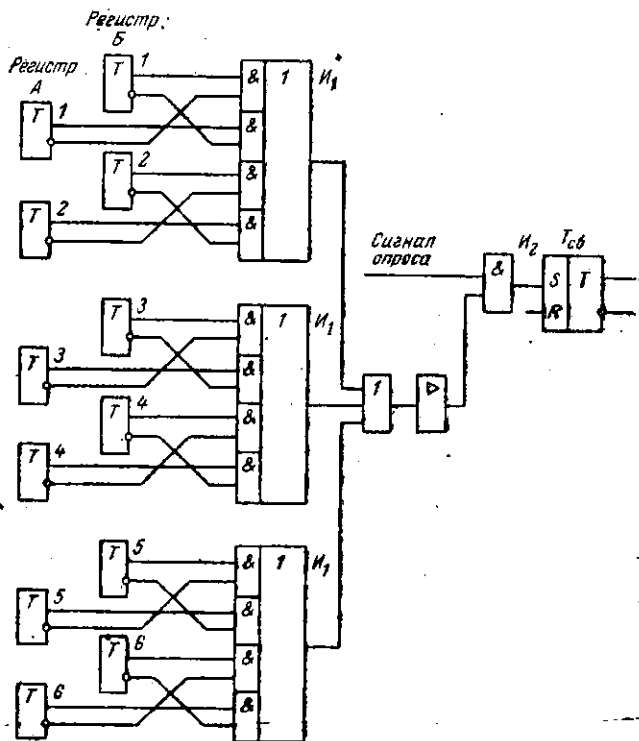


Рис. 5.2. Схема контроля по совпадению

сти многоразрядного числа, как правило, больше одной ступени.

Однако контроль по совпадению обладает существенным недостатком. Этот метод позволяет проверить правильность передачи числа в регистр и отсутствие сбоев при его хранении только до тех пор, пока не изменит своего состояния регистр, из которого передавалась информация. При контроле по четности проверяется не только пра-

тельность передачи, но и отсутствие сбоев при хранении числа в регистре (памяти) в течение сколь угодно долгого времени.

Корректирующие коды. Код Хэмминга. В ЭВМ, например в оперативной памяти, в запоминающих устройствах на магнитных лентах и дисках применяются корректирующие коды, позволяющие не только обнаруживать, но и исправлять ошибки. К таким кодам относятся код Хэмминга, групповые корректирующие коды, циклические коды и др.

Код Хэмминга строится таким образом, что к имеющимся информационным разрядам слова добавляется определенное число контрольных разрядов, которые формируются перед передачей информации путем подсчета четности суммы единиц для определенных групп информационных разрядов. Контрольная аппаратура на приемном конце образует из принятых информационных и контрольных разрядов путем аналогичных подсчетов четности корректирующее число, которое равно 0 при отсутствии ошибки либо указывает место ошибки, например двоичный порядковый номер ошибочного разряда в слове. Ошибочный разряд автоматически корректируется путем изменения его состояния на противоположное.

Рассмотрим более подробно процесс кодирования с использованием кода Хэмминга с коррекцией одиночной ошибки (минимальное кодовое расстояние $d_{\min}=3$). Если кодовое слово не содержит ошибок, то корректирующее число должно быть равно 0. При наличии ошибки это число должно содержать номер ошибочного разряда. Если в младшем разряде корректирующего числа появится 1, то это означает ошибку в одном из тех разрядов слова, порядковые номера которых имеют 1 в младшем разряде (разрядов с нечетными номерами). Введем первый контрольный разряд, которому присвоим нечетный порядковый номер и который установим при кодировании таким образом, чтобы сумма единиц всех разрядов с нечетными порядковыми номерами была равна 0. Эта операция может быть записана в виде

$$E_1 = x_1 \oplus x_3 \oplus x_5 \oplus \dots = 0,$$

где x_1, x_3 и т. д. — двоичные символы, размещенные в разрядах с порядковыми номерами 1, 3 и т. д.

Появление единицы во втором разряде (справа) корректирующего числа означает ошибку в тех разрядах слова, порядковые номера которых (2, 3, 6, 7, 10, 11, 14, 15 и

т. д.) имеют единицу во втором разряде справа. Поэтому вторая операция кодирования, позволяющая найти второй контрольный разряд, которому должен быть присвоен какой-либо порядковый номер из группы 2, 3, 6, 7, 10, 11 и т. д., имеет вид

$$E_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus \dots = 0.$$

Рассуждая аналогичным образом, можно определить все другие контрольные разряды путем выполнения операций

$$E_3 = x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_{12} \oplus x_{13} \oplus x_{14} \oplus x_{15} \oplus \dots = 0;$$

$$E_4 = x_8 \oplus x_9 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{13} \oplus x_{14} \oplus x_{15} \oplus \dots = 0$$

и т. д.

После приема кодового слова (совместно со сформированными контрольными разрядами) выполняются те же операции подсчета, которые были описаны выше, а образующееся число

$$E_k E_{k-1} \dots E_3 E_2 E_1$$

считается корректирующим.

При отсутствии ошибок $E_k E_{k-1} \dots E_2 E_1 = 0$, при наличии ошибок не равными нулю будут те суммы E_i , в образовании которых участвовал ошибочный разряд; корректирующее число при этом будет равно порядковому номеру ошибочного разряда.

Выбор места для контрольных разрядов производится таким образом, чтобы контрольные разряды участвовали только в одной операции подсчета четности. Это упрощает процесс кодирования. Рассмотрение выражений для E_1 , E_2 , E_3 и т. д. показывает, что такими позициями являются разряды с номерами, являющимися целыми степенями двойки: 1, 2, 4, 8, 16 и т. д.

Требуемое число контрольных разрядов (разрядность корректирующего числа) определяется из следующих соображений. Пусть кодовое слово длиной n разрядов имеет m информационных и $k = n - m$ контрольных разрядов. Корректирующее число длиной k разрядов описывает 2^k состояний, соответствующих отсутствию ошибки и появлению ошибки в i -м разряде. Таким образом, должно соблюдаться соотношение

$$2^k \geq n + 1 \quad (5.3)$$

или

$$2^k - k - 1 \geq m. \quad (5.4)$$

Из этого неравенства следует, например, что пять контрольных разрядов позволяют передавать в коде Хэмминга от 11 до 26 информационных разрядов и т. д. Таким образом, избыточность кода Хэмминга значительно выше избыточности кода с проверкой четности.

Контроль по коду Хэмминга реализуется с помощью набора схем подсчета четности (см. рис. 5.1), которые при кодировании определяют контрольные разряды, а при декодировании формируют корректирующее число.

5.3. ЦИКЛИЧЕСКИЕ КОДЫ

Циклические коды основаны на представлении передаваемых данных в виде полинома и используются в основном при последовательной передаче данных между ЭВМ и внешними запоминающими устройствами, а также при передаче данных по каналу связи.

Контроль передачи данных с помощью циклических (полиномиальных) кодов основан на следующей закономерности: если информационный полином умножить на некоторый так называемый порождающий полином, получившийся в результате этого кодовый полином передать приемнику информации и выполнить в нем обратное действие, т. е. деление полинома принятого сообщения на порождающий полином, то ненулевой остаток будет означать, что произошла ошибка. Нулевой остаток будет означать, что ошибки нет (или она не обнаружена).

Для пояснения процедуры формирования циклического кода и его декодирования введем следующие обозначения:

$G(x)$ — информационный полином, соответствующий передаваемой информации длиной m бит. Он имеет степень меньше $m-1$;

$P(x)$ — порождающий полином степени k , определяющий число контрольных бит, а также обнаруживающую и корректирующую способность циклического кода;

$F(x)$ — кодовый полином, соответствующий передаваемому циклическому коду. Это полином степени $m+k$, делящийся без остатка на порождающий полином степени k .

Образование кодового полинома $F(x)$ путем умножения информационного полинома $G(x)$ на порождающий полином $P(x)$ неудобно тем, что получающийся при этом код не будет систематическим (контрольные биты не будут выделены). Вследствие этого применяется другой способ формирования кодового полинома, при котором старшие коэффициенты образуют информационные знаки, а младшие —

контрольные. Информационный полином $G(x)$ степени $m-1$, который необходимо закодировать, умножается на x^k , что соответствует сдвигу на k разрядов влево. Полученный таким образом полином $x^k G(x)$ делится на $P(x)$ для определения остатка $R(x)$:

$$x^k G(x)/P(x) = Q(x) \oplus R(x)/P(x). \quad (5.5)$$

Из этого выражения следует

$$\begin{aligned} x^k G(x) &= Q(x) P(x) \oplus R(x); \\ F(x) &= Q(x) P(x) = x^k G(x) \oplus R(x). \end{aligned} \quad (5.6)$$

Полином $F(x)$ делится на $P(x)$ без остатка и поэтому является кодовым полиномом.

Остаток $R(x)$ имеет степень меньше k , и $x^k G(x)$ имеет нулевые коэффициенты в k младших членах. Таким образом, m старших коэффициентов кода $F(x)$ равны коэффициентам информационного полинома $G(x)$ и представляют собой кодируемое сообщение, а младшие k коэффициентов кодового полинома $F(x)$ представляют собой коэффициенты остатка $R(x)$, т. е. контрольные биты. Из вышеприведенного следует, что кодовый полином можно получить путем сдвига информационного полинома на k бит, деления его на порождающий полином $P(x)$ и записи остатка в младшие k бит кодового полинома. Процедура кодирования, соответствующего этому алгоритму, реализуется с помощью сдвигового регистра с обратными связями, соответствующими виду порождающего полинома $P(x)$.

Если полином, полученный при передаче сообщения, содержит ошибки, то он может быть представлен в виде

$$H(x) = F(x) \oplus E(x). \quad (5.7)$$

Это выражение означает, что если полином принятого сообщения $H(x)$ не делится на $P(x)$, то при передаче информации произошла ошибка $E(x)$.

Выбор порождающего полинома $P(x)$ зависит от характера ошибок при передаче. Можно записать следующие правила выбора порождающего полинома для разных ошибок:

ошибки в одном бите обнаруживаются, если порождающий полином содержит более одного члена;

ошибки в двух битах обнаруживаются, если порождающий полином содержит три члена;

ошибки в нечетном числе бит обнаруживаются, если порождающий полином содержит множитель $(x+1)$;

пакеты ошибок длиной менее $k+1$ бита обнаруживаются, если порождающий полином содержит множитель $x+1$ и один множитель с тремя членами и более ($k+1$ — число бит порождающего полинома). Этот же порождающий полином обнаруживает с вероятностью $1-(1/2)^{k-1}$ пакеты ошибок длиной $(k+1)$ бит и с вероятностью $1-(1/2)^k$ пакеты ошибок длиной более $k+1$ бит [43].

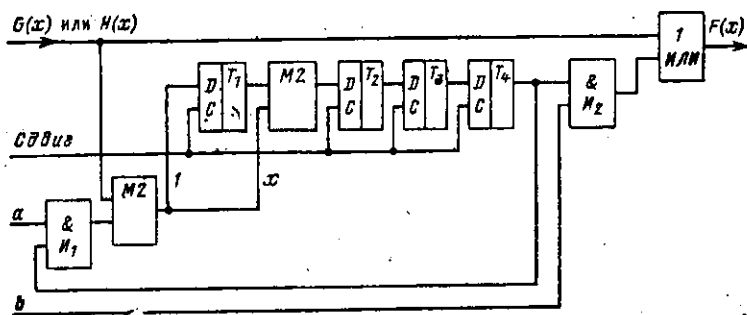


Рис. 5.3. Схема формирования и декодирования циклического кода

На рис. 5.3 показана схема сдвигового регистра с обратными связями и схемами сложения по модулю 2, реализующими схему деления на порождающий полином.

Важным преимуществом таких схем является возможность их применения для контроля передаваемой информации как при приеме, так и при генерации контрольных бит при передаче, что позволяет обойтись одной такой схемой в приемной и передающей аппаратуре.

Рассмотрим в качестве примера схему, реализующую деление на порождающий полином $P(x) = x^4 + x + 1$. Предположим, что передаются данные вида 11010011, которые представляются в виде полинома $G(x) = x^7 + x^6 + x^4 + x + 1$. Как было сказано, циклический код передаваемых данных образуется путем умножения информационного полинома $G(x)$ на x^4 : $x^4 G(x) = x^4(x^7 + x^6 + x^4 + x + 1)$, что эквивалентно сдвигу данных, соответствующих информационному полиному $G(x)$, на 4 бита в сдвиговом регистре. Как только первая единица (коэффициент при старшем члене полинома) сдвинется до конца регистра, т.е. появится на выходе триггера T_4 при очередном сдвиге, будет выполнено вычитание делителя. Проследим за процессом деления $x^4 G(x)$ на $P(x)$ и образования остатка:

$$\begin{array}{r}
 x^4 G(x) = 110100110000 \\
 P(x) = 10011 \\
 \hline
 110100110000 \\
 10011 \\
 \hline
 10010 \\
 10011 \\
 \hline
 11100 \\
 10011 \\
 \hline
 11110 \\
 10011 \\
 \hline
 11010 \\
 10011 \\
 \hline
 1001 \text{ — остаток, т. е. } R(x).
 \end{array}$$

В течение первых восьми тактов сдвига на входе a схемы I_1 имеется разрешение. Данные, соответствующие информационному полиному, одновременно с передачей на выход схемы **ИЛИ** поступают также и в сдвиговый регистр. К началу девятого такта разрешение со входа a снимается, а на вход b схемы I_2 подается разрешение, в результате чего на выход схемы поступает остаток $R(x)$, накопившийся в сдвиговом регистре. Таким образом, на выходе схемы **ИЛИ** образуется последовательность бит, соответствующая кодовому полиному $F(x) = 110100111001$.

В том случае, когда описанная схема используется для декодирования принятой информации (обнаружение ошибок), на ее вход поступает последовательность полинома $H(x)$. Если после $m+k$ сдвигов сдвиговый регистр находится в нулевом состоянии, это означает, что передача выполнена без ошибок.

Циклические (полиномиальные) коды получили широкое распространение благодаря высокой обнаруживающей способности и относительной простоте схем кодирования и декодирования.

5.4. КОНТРОЛЬ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

Арифметические операции, как правило, можно представить в виде последовательности следующих элементарных операций: передача слова и операции преобразования содержимого триггерных регистров — сдвиг, взятие обратного кода и сложение.

Операция сдвига информации в регистре представляет собой по существу передачу информации из i -х разрядов регистра в $(i+m)$ -е или $(i-m)$ -е разряды в зависимости от направления сдвига (m — число разрядов, на которое производится сдвиг). Поэтому для контроля операции сдвига можно использовать те же методы, что и для контроля пе-

редачи информации, например контроль четности суммы единиц.

Регистр, в котором производится сдвиг, должен иметь дополнительный контрольный разряд, устанавливаемый перед сдвигом в такое состояние, чтобы сумма единиц регистра вместе с контрольным разрядом была, например, четной. Кроме схемы определения общей четности содержимого регистра необходимы схемы, устанавливающие четность разности между числом единиц, выдвигаемых из регистра, и числом единиц, вдвигаемых в регистр. Если в освобождающиеся при сдвиге разряды вдвигается 0, то достаточно иметь схему определения четности суммы единиц выдвигаемых разрядов. Одновременно со сдвигом выполняется контрольная операция, заключающаяся в том, что состояние контрольного разряда меняется на противоположное при нечетности суммы единиц выдвигаемых разрядов. Тогда при правильном выполнении сдвига общая четность суммы единиц в регистре после сдвига не меняется.

Операция взятия обратного кода может быть также проконтролирована путем использования кодов с проверкой четности. Если число информационных разрядов в слове четно, то число единиц в слове четно при четном числе нулей и нечетно при нечетном числе нулей. В этом случае после образования обратного кода четность числа единиц в слове сохраняется и правильность выполнения операции можно определить, проверив сохранение четности или нечетности суммы единиц в слове, включая контрольный разряд.

Если число информационных разрядов в слове нечетно, то четному числу единиц в слове соответствует нечетное число нулей, а нечетному числу единиц — четное число нулей. В этом случае после образования обратного кода четность числа единиц изменится на обратную и для проверки правильности выполнения операции необходимо взять обратный код от содержимого контрольного разряда и проверить сохранение четности или нечетности суммы единиц в слове с учетом инвертированного контрольного разряда.

Из множества разработанных способов контроля операции сложения рассмотрим способ, использующий проверку четности.

При сложении чисел a и b разряды суммы S образуются в соответствии с выражениями

$$\left. \begin{aligned} S_1 &= a_1 \oplus b_1 \oplus P_1; \\ S_2 &= a_2 \oplus b_2 \oplus P_2; \\ &\dots \dots \dots \\ S_n &= a_n \oplus b_n \oplus P_n, \end{aligned} \right\} \quad (5.8)$$

где S_i, a_i, b_i, P_i ($i=1, 2, \dots, n$) — соответственно значения разрядов суммы, слагаемых и переноса, поступающего в i -й разряд. Знак $\oplus$ означает сложение по модулю 2; n — число разрядов слагаемых и суммы.

Сложив все n приведенных выше равенств по модулю 2, получим

$$S_1 \oplus S_2 \oplus \dots \oplus S_n = (a_1 \oplus a_2 \oplus \dots \oplus a_n) \oplus \\ \oplus (b_1 \oplus b_2 \oplus \dots \oplus b_n) \oplus (P_1 \oplus P_2 \oplus \dots \oplus P_n). \quad (5.9)$$

Поскольку сумма по модулю 2 всех разрядов слова выражает четность суммы единиц слова, последнее уравнение можно переписать в виде

$$\text{четность } S = \text{четность } a \oplus \text{четность } b \oplus \text{четность } P. \quad (5.10)$$

Таким образом, при правильном образовании суммы четность суммы ее единиц должна совпасть с четностью, определяемой выражением (5.10). Однако, как указывалось ранее, контроль по четности не обнаруживает четного числа ошибок. Сбой в схеме образования цифры разряда суммы дает одиночную ошибку, и она будет обнаружена. Если сбой произойдет в схеме формирования переноса, то он может привести к распространению ошибки по многим разрядам суммы. В связи с этим для полноты контроля необходимо проверять правильность образования переносов P_i . Такой контроль может быть организован с помощью схемы, которая проверяет, что в каждом разряде существует либо перенос в прямом коде, либо инверсия переноса и не существует одновременно и то, и другое.

Другой способ контроля заключается в дублировании схем формирования переносов и сравнении переносов основной и дублирующей схем.

Упрощенная схема контроля сумматора приведена на рис. 5.4. Оба слагаемых поступают на схемы формирования переносов. Сформированные переносы проверяются на совпадение с помощью схемы, подобной изображенной на рис. 5.2. Для переносов и отдельно для суммы формируются контрольные разряды четности. Затем схема проверки четности проверяет выполнение условия (5.10).

Контроль выполнения арифметических операций (сложения, вычитания, умножения) можно осуществлять с помощью контрольных кодов, представляющих собой остатки от деления чисел на некоторый модуль R (контроль по модулю R).

Если в качестве контролируемого кода используется остаток по модулю R , то в качестве контрольной операции над остатками может быть выбрана та же арифметическая операция, которая производится над числами. Это следует из того, что для сложения, вычитания и умножения действительно соотношение

$$R(A * B) = R[R(A) * R(B)], \quad (5.11)$$

где $R(X)$ — остаток числа X по модулю R ; $*$ — знак арифметических операций сложения, вычитания или умножения.

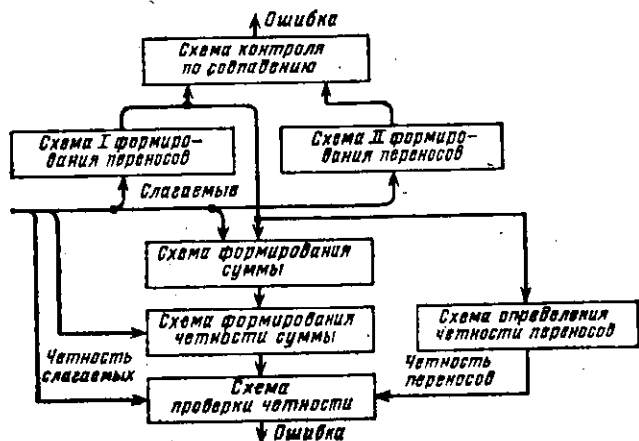


Рис. 5.4. Контроль сложения с использованием проверки по четности

Контроль арифметических операций по модулю организуется следующим образом. Каждому числу, участвующему в арифметической операции, ставится в соответствие контрольный код — остаток по модулю R . Одновременно с выполнением основной операции над числами та же операция производится над их контрольными кодами, и контрольный код результата основной операции сравнивается с результатом операции над контрольными кодами исходных чисел. При несовпадении фиксируется ошибка.

Чем меньше R , тем меньше разрядность контрольного кода и проще дополнительная аппаратура. Для двоичных чисел контроль по модулю возможен при $R=3$, поэтому в ЭВМ часто используют контроль по модулю 3.

Покажем, что при контроле по модулю 3 обнаруживаются любые одиночные ошибки. Одиночная ошибка в ка-

ком-либо разряде двоичного числа соответствует изменению числа на $\pm 2^i$. Для обнаружения ошибки необходимо, чтобы контрольные коды чисел a и $a \pm 2^i$ не совпадали, т.е. $R(a) \neq R(a \pm 2^i) = R(a) \pm R(2^i)$ или $R(2^i) \neq 0$.

Но 2^i не делится на 3 без остатка, следовательно, требуемое условие выполняется. Кроме одиночных ошибок при контроле по модулю 3 обнаруживается часть двойных ошибок, а именно те, при которых правильный и ошибочный результаты имеют несовпадающие остатки от деления на 3.

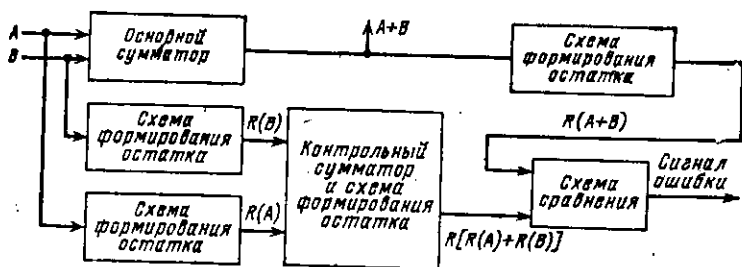


Рис. 5.5. Структурная схема сумматора с контролем по модулю 3

На рис. 5.5 изображена структурная схема сумматора с контролем по модулю 3. Отметим, что контрольный сумматор значительно проще основного, так как содержит только два двоичных разряда. При реализации контроля важное значение имеет построение схем формирования остатков при минимальных затратах оборудования. Очевидно, что нахождение остатка путем прямого деления двоичного числа на 3 — путь неприемлемый. Однако кодирование по модулю 3 обладает свойствами, позволяющими строить достаточно простые комбинационные схемы формирования остатка по модулю 3.

Может быть применен контроль по модулю с большим основанием, чем 3. При этом увеличивается число кратных ошибок, которые могут обнаруживаться системой контроля, однако возрастает сложность кодирующей и декодирующей аппаратуры.

5.5. КОНТРОЛЬ КОМБИНАЦИОННЫХ СХЕМ

Основными способами контроля комбинационных схем (КС) являются контроль путем дублирования, путем проверки выходных сигналов и с помощью образования контрольных бит.

Контроль путем дублирования (рис. 5.6) является наиболее универсальным и основан на побитном сравнении выходных сигналов основной и дублирующей комбинацион-

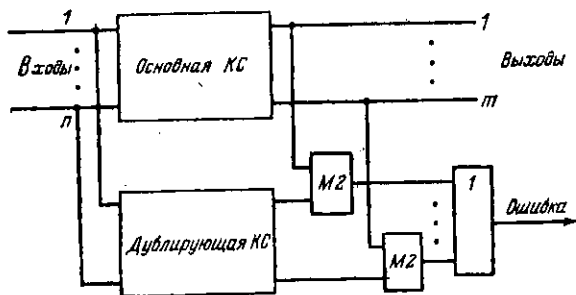


Рис. 5.6. Схема контроля путем дублирования

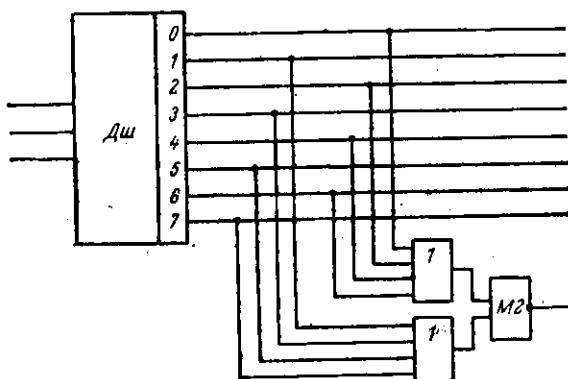


Рис. 5.7. Схема контроля дешифратора

ных схем с последующим объединением выходных сигналов схем сравнения в один общий сигнал ошибки.

Контроль путем проверки выходных сигналов используется для тех комбинационных схем, в которых ошибка может быть обнаружена по выходному сигналу. К таким комбинационным схемам относятся дешифраторы. Приведенная на рис. 5.7 схема иллюстрирует контроль трехвыходового дешифратора. Для контроля выходы дешифратора (Дш) делятся на две группы — четную и нечетную. Сигналы с четных выходов дешифратора подаются на входы первой

схемы ИЛИ, а с нечетных выходов — на входы второй. При одновременном отсутствии или присутствии сигналов на выходах обеих схем ИЛИ схема по модулю 2 вырабатывает сигнал ошибки.

Подобные схемы контроля обнаруживают все неисправности кроме тех, которые приводят к появлению сигналов на двух или более выходах дешифратора, относящихся к одной и той же группе (четной или нечетной).

Контроль с помощью образования контрольных бит используется в тех случаях, когда по входным сигналам схемы можно предсказать контрольные биты выходных сигналов.

5.6. САМОПРОВЕРЯЕМЫЕ СХЕМЫ КОНТРОЛЯ

Неисправность в самих схемах контроля может остаться необнаруженной, если не предпринять дополнительных мер по ее обнаружению.

Работа ЭВМ с неисправной схемой контроля приводит к ухудшению достоверности функционирования в результате того, что ошибки, возникающие из-за сбоев, не будут обнаружены и исправлены.

Для обнаружения неисправностей схем контроля применяют: самопроверяемые схемы контроля; диагностические средства проверки схем контроля.

Схема контроля (СК) называется *самопроверяемой*, если она обнаруживает не только неисправности контролируемого устройства (КУ), но также и свои собственные.

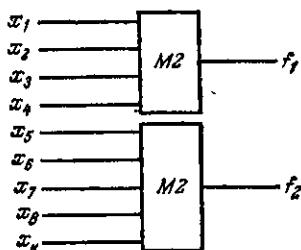
Схема контроля с двумя выходами f_1, f_2 называется *полностью самопроверяемой*, если она обладает следующими свойствами: а) самотестируемости — все неисправности СК из заданного класса проявляются на выходах f_1, f_2 в виде пар сигналов со значениями 00 или 11 хотя бы при одном из наборов входных сигналов из множества наборов рабочих входных сигналов при исправном КУ; б) защищенности от неисправностей — каждая неисправность СК из заданного класса проявляется на выходах f_1, f_2 только в виде пары сигналов со значениями 00 или 11 [58].

Одним из требований при построении самопроверяемой схемы контроля (ССК) является требование раздельной реализации функций f_1 и f_2 , которая гарантирует, что любая одиночная константная неисправность на входах и выходах ССК не приведет к инвертированию сигналов на обоих выходах схемы.

Значения сигналов 01, 10 на выходах f_1, f_2 свидетельствуют о правильном функционировании контролируемого устройства и ССК, а значения 00, 11 — о наличии одиночной неисправности в устройстве или ССК.

Вопросы построения самопроверяемых схем контроля подробно изложены в [56].

В качестве примера ССК рассмотрим схему контроля по четности байтного регистра (восемь информационных и один контрольный бит), изображенную на рис. 5.8. Эта ССК состоит из двух отдельных схем контроля. Первая реализует функцию



$$f_1 = x_1 \oplus x_2 \oplus x_3 \oplus x_4,$$

Рис. 5.8. Самопроверяемая схема контроля по четности

а вторая — функцию

$$f_2 = x_5 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{\text{к}}.$$

Поскольку любая входная информация имеет нечетное число единиц, одна из групп будет содержать их четное число, а другая — нечетное. Таким образом, выходной сигнал ССК всегда будет принимать значения 01 или 10.

При появлении одиночной ошибки на входе число единиц станет четным и обе группы будут иметь один и тот же выходной сигнал со значениями 00 или 11.

Поскольку входная информация обычно обеспечивает все множество возможных кодов, на обе схемы контроля поступает полное множество тестовых наборов. Это обязательно приведет к обнаружению ошибки в самой схеме контроля при одном из наборов входных сигналов.

Другим примером самопроверяемой схемы контроля является схема контроля дешифратора, приведенная на рис. 5.9. При отсутствии неисправностей в дешифраторе и в самой схеме контроля выходные сигналы ССК принимают значения 10 и 01. В противном случае выходные сигналы ССК принимают значения 00 или 11, что свидетельствует о неисправности дешифратора или схемы контроля. Для объединения сигналов с выходов схем контроля с целью формирования общего сигнала ошибки применяют самопроверяемые схемы сжатия. Схема сжатия, приведенная на рис. 5.10, обеспечивает объединение сигналов с выходов двух ССК в один общий сигнал. Если на одном из входов

схемы сжатия (СЖ) сигналы f_1, f_2 принимают значения 00 или 11, то сигнал на ее выходе также будет свидетельствовать о наличии ошибки.

Диагностические средства проверки схем контроля. Диагностические средства проверки схем контроля предназна-

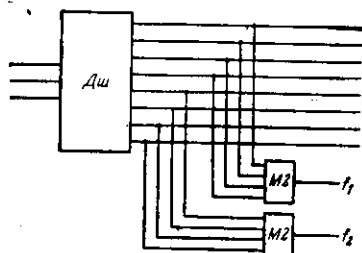


Рис. 5.9. Самопроверяемая схема контроля дешифратора

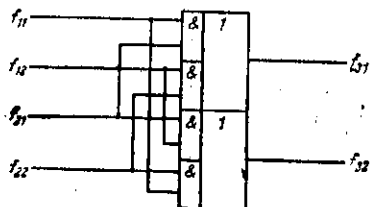


Рис. 5.10. Схема сжатия

чены для периодической проверки работоспособности этих схем. Существуют следующие способы проверки схем контроля:

- введение информации с неправильной четностью в информационные тракты ЭВМ;

- имитация ошибок в схемах контроля четности;

- имитация ошибок в схемах контроля путем дублирования;

- имитация одиночных и двойных ошибок в оперативной памяти;

- имитация ошибки в общем регистре ошибок.

Проверка схем контроля в режиме диагностирования выполняется с отключенной системой контроля ЭВМ.

5.7. ОПРЕДЕЛЕНИЕ ЭФФЕКТИВНОСТИ СИСТЕМЫ АВТОМАТИЧЕСКОГО КОНТРОЛЯ ЭВМ

Эффективность системы автоматического контроля определяется долей обнаруживаемых ошибок. Для оценки эффективности используются методы аналитический, вероятностного моделирования, экспериментальный.

Аналитический метод основан на определении доли обнаруживаемых ошибок для каждого функционального узла ЭВМ по формуле

$$\gamma_i = \lambda_{oi} / \lambda_i, \quad (5.12)$$

где γ_j — доля обнаруживаемых ошибок j -го функционального узла; λ_j — интенсивность неисправностей j -го функционального узла; λ_{0j} — интенсивность неисправностей контролируемой части функционального узла.

Доля обнаруживаемых ошибок для всей ЭВМ определяется по формуле

$$\gamma = \sum_1^k \lambda_{0j} / \sum_1^k \lambda_j. \quad (5.13)$$

Метод вероятностного моделирования основан на логическом моделировании работы функциональных узлов. С этой целью используется модель, отображающая логику работы элементов, структуру и алгоритм работы функциональной части в исправном и неисправном состояниях. Имитация случайных неисправностей при моделировании обеспечивается путем выбора неисправного элемента на основе случайных равновероятно распределенных чисел с учетом веса интенсивности неисправностей всех элементов функциональной части.

Для моделирования применяется комплекс моделирующих программ.

Наиболее приемлемым является экспериментальный метод, основанный на введении в ЭВМ искусственных неисправностей и проверке реакции системы автоматического контроля на эти неисправности [55].

Число вводимых при проведении эксперимента неисправностей определяется с помощью приведенных на рис. 5.11, а—д графиков зависимости числа вводимых неисправностей N от числа пропущенных средствами аппаратного контроля ошибок $N_{пр}$ и требуемой достоверности оценки A процента обнаруживаемых ошибок γ .

Для определения числа вводимых неисправностей через точку на оси ординат, соответствующую доверительной вероятности $A=0,9$, проводится прямая, параллельная оси абсцисс. Проекция на ось абсцисс точек пересечения указанной прямой с кривыми, соответствующими разному числу пропущенных средствами аппаратного контроля ошибок, дает необходимое число вводимых неисправностей $N_0—N_4$ соответственно для 0—4 ошибок, пропущенных средствами контроля.

Обычно при использовании экспериментального метода определения доли обнаруживаемых ошибок сначала аналитическим методом определяется расчетное значение γ . По полученному значению γ выбирается ближайший

соответствующий график из графиков, приведенных на рис. 5.11.

С помощью выбранного графика определяется, как было указано выше, число неисправностей N_0 и N_4 , искусственно вводимых в ЭВМ при проведении эксперимента.

Для введения искусственных неисправностей подготавливается их перечень. Число вводимых неисправностей должно соответствовать интенсивности неисправностей функциональных узлов. Дальнейшая процедура выполняется следующим образом:

а) вводятся N_0 неисправностей. Если система контроля обнаружила все введенные неисправности, то выполняется п. «б», если не все, то п. «в»;

б) определяется очередной график (рис. 5.11), построенный для значения γ больше выбранного на 5 %. По этому графику определяется число вводимых неисправностей для нулевого числа обнаруженных неисправностей. Дополнительно в ЭВМ вводится такое число неисправностей, чтобы их общее число соответствовало числу, полученному из графика. Если система контроля обнаружила все неисправности и число вводимых неисправностей определялось по графику, соответствовавшему значению γ , меньшему 95 %, то повторяется выполнение данного пункта.

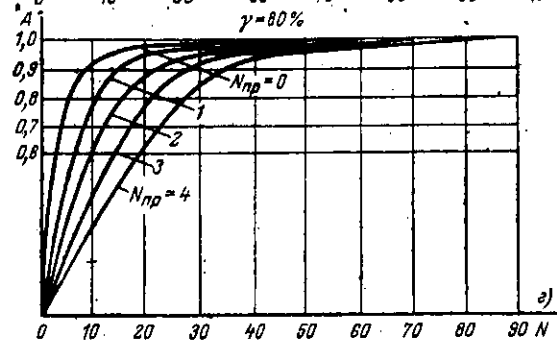
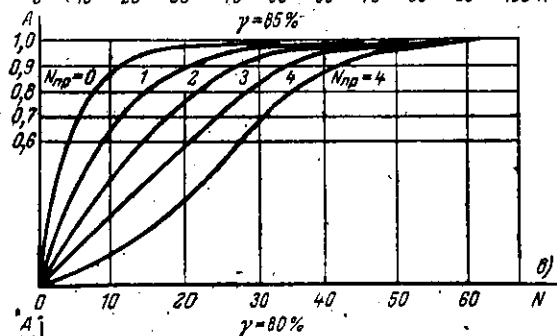
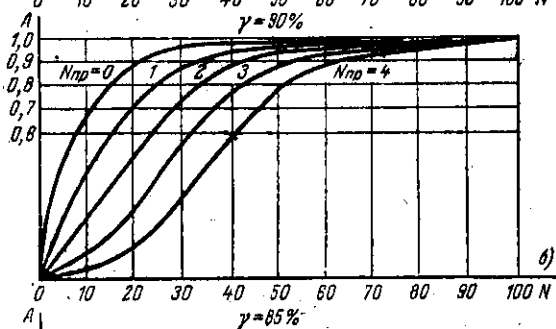
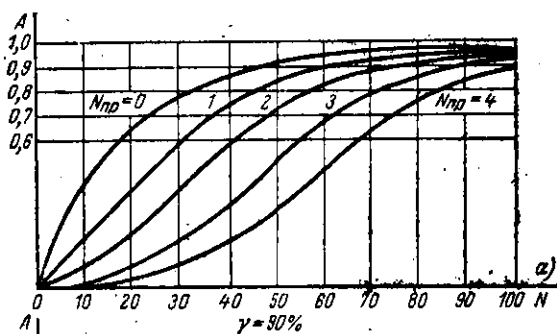
При отсутствии ошибок, обнаруженных системой контроля, при числе введенных неисправностей, соответствующем графику для $\gamma = 95$ %, считается, что реальный процент обнаруживаемых ошибок средствами аппаратного контроля выше 95 %.

Если при очередном увеличении числа вводимых неисправностей система контроля пропустила одну ошибку, эксперимент прекращается и процент обнаруживаемых ошибок считается равным значению γ для графика, соответствующего предыдущему числу N_0 вводимых неисправностей;

в) если система контроля пропустила ошибки при введении N_0 неисправностей, соответствующих расчетному значению γ , число вводимых неисправностей увеличивается до $N_1 - N_4$.

Если при введении $N_1 - N_4$ неисправностей система контроля пропустила 1—4 неисправности соответственно, считается, что реальное значение равно расчетному;

г) если система контроля пропустила более четырех ошибок, то определяется график (рис. 5.11), построенный для значения γ меньше выбранного на 5 %. По этому графику определяется число неисправностей, соответствующих четырем пропущенным ошибкам. Если полученное число



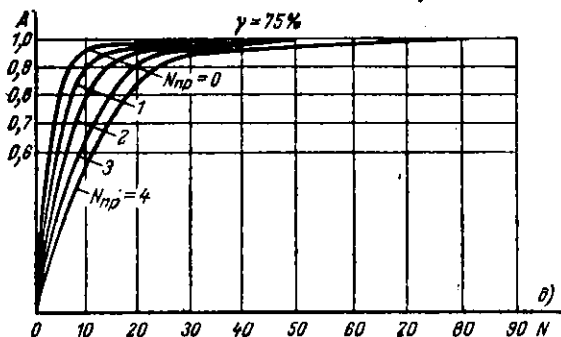


Рис. 5.11. Характеристики системы контроля

неисправностей меньше числа введенных неисправностей для четырех пропущенных ошибок или равно ему, то считается, что обнаруживающая способность средств аппаратного контроля равна значению γ , для которого составлен график. В противном случае выполнение данного пункта повторяется до тех пор, пока будет выбран график, построенный для $\gamma=75\%$. Если и для этого графика не будет выполняться указанное выше условие, то считается, что обнаруживающая способность системы контроля ниже 75% .

Иногда для проведения эксперимента используют специальные средства физической имитации неисправностей, позволяющие вносить неисправность в указанное место под управлением специальной программы. Результаты имитации неисправностей позволяют оценить качество разработанной системы контроля и принять меры по ее улучшению.

3.8. СИСТЕМА АВТОМАТИЧЕСКОГО КОНТРОЛЯ ЭВМ

Рассмотрим систему автоматического контроля вычислительной машины на примере ЭВМ ЕС-1045.

Электронно-вычислительная машина ЕС-1045 является одной из моделей ЕС ЭВМ второй очереди. Ее центральная часть состоит из следующих основных устройств: процессора, устройства управления памятью, каналов и основной оперативной памяти.

Процессор состоит из следующих блоков: микропрограммного управления, выборки команд, операций, измерения времени, регистрации программных событий, акселе-

ратора, прямого управления, пульта управления и восстановления и диагностики.

Акселератор представляет собой специальное арифметическое устройство, ускоряющее выполнение некоторых команд ЭВМ.

Устройство управления памятью состоит из буферной и индексной памяти, блока управления обменом информацией, буферной памяти команд, блока динамического преобразования адресов, адаптера оперативной памяти, памяти ключей защиты (рис. 5.12).

Средства контроля перечисленных выше блоков и устройств образуют систему автоматического контроля ЭВМ ЕС-1045.

Система автоматического контроля ЭВМ ЕС-1045 обеспечивает обнаружение:

- двойных (и исправление одиночных) ошибок оперативной памяти благодаря использованию модифицированного кода Хэмминга;

- ошибок в сумматорах и схемах управления с помощью самопроверяемого дублирования;

- ошибок в передачах и внутренних запоминающих устройствах путем контроля на нечетность;

- ошибок в дешифраторах путем контроля числа активных выходов;

- ошибок счетчиков путем контроля на нечетность с использованием схем предсказания нечетности;

- ошибок в самих схемах контроля благодаря их самоконтролируемости;

- ошибок в акселераторе путем контроля по модулю 9 и 15;

- нарушений временных соотношений с помощью специальных счетчиков времени;

- временных ошибок в каналах и неправильных сигналов в интерфейсе ввода-вывода с микропрограммной установкой триггеров соответствующих ошибок.

Поскольку основным методом контроля в ЭВМ является метод контроля на нечетность, непрерывность контроля в местах перехода от одного вида контроля к другому обеспечивается путем добавления бита нечетности.

Сигналы ошибок от схем контроля процессора, каналов, блока управления памятью (БУП), акселератора поступают в сводный регистр ошибок (РгОш). В регистре ошибок фиксируются также и имитированные ошибки, предназначенные для проверки схем обработки ошибок.

Общая схема сбора сигналов ошибок в РгОш показана

на рис. 5.13. Общий регистр ошибок состоит из двух ступеней. Содержимое первой ступени РгОш обновляется в каждом машинном такте, а во второй ступени происходит накопление сигналов ошибок в промежутках между сбросами РгОш. Выходы второй ступени РгОш соединены со схемой сигнализации и регистрации (на рис. 5.13 для простоты регистр ошибок представлен в укрупненном виде). Общие

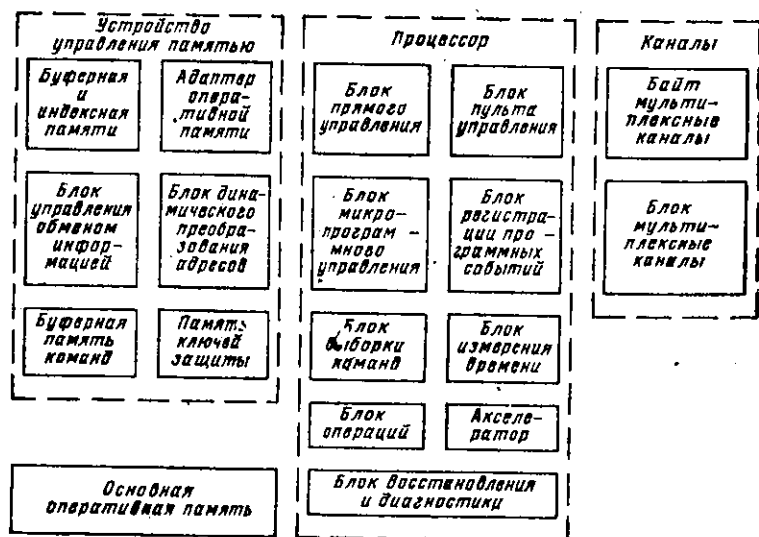
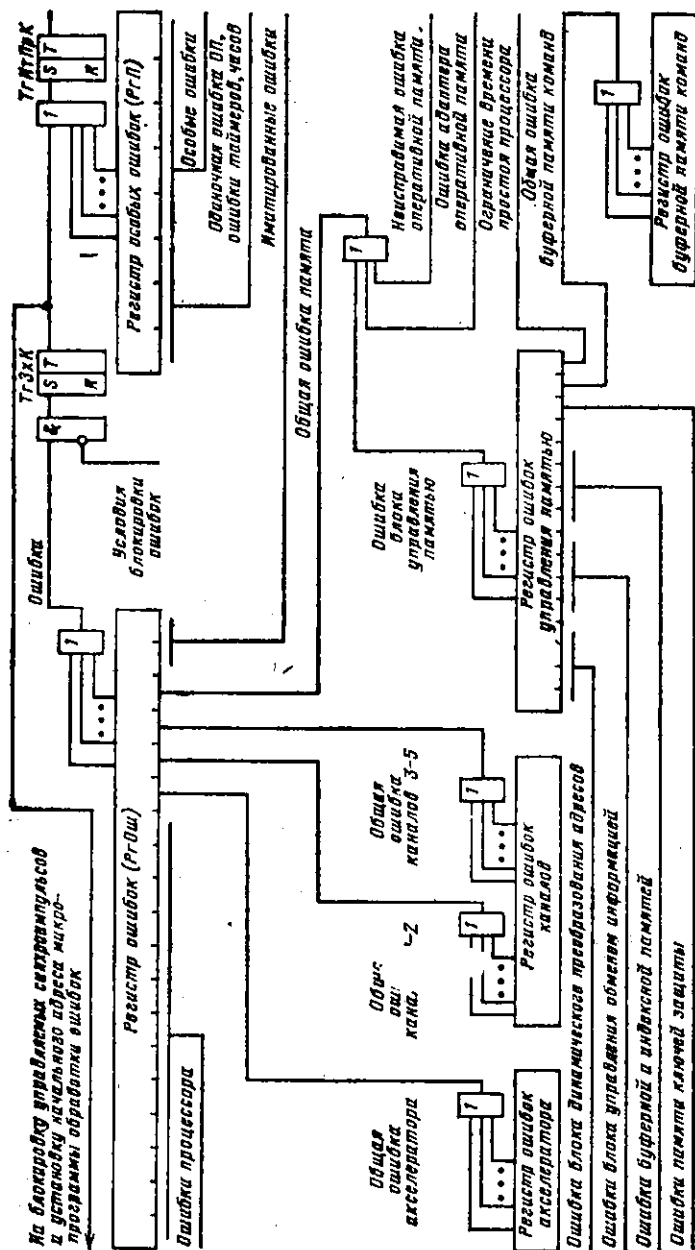


Рис. 5.12. Состав центральной части ЭВМ ЕС-1045

сигналы ошибок с первой ступени РгОш поступают в схемы обработки ошибок, на установку триггера захвата от схем контроля (при отсутствии условий блокировки ошибок) и триггера неотработанного прерывания от схем контроля.

Триггер захвата от схем контроля (ТгЗхК) обеспечивает переход к микропрограмме обработки ошибок путем установки в регистр адреса управляющей памяти фиксированного начального адреса микропрограммы обработки ошибок. Единичное состояние ТгЗхК вызывает также блокировку управляемых синхронимпульсов ЭВМ на время регистрации состояния ЭВМ в служебной памяти.

Триггер неотработанного прерывания от схем контроля (ТгНтПрК) вызывает ветвление микропрограммы выборки



команд для перехода к прерыванию от схем контроля (при наличии соответствующих условий).

Триггер неотработанного прерывания от схем контроля устанавливается при установке ТгЗхК и наличии одной из следующих ошибок, зафиксированных в регистре особых ошибок: одиночной ошибки оперативной памяти, ошибки таймеров, ошибки часов.

Регистр особых ошибок, называемый регистром повторения РгП, служит также для запоминания таких ошибок, как постоянная ошибка (возникает при восьмикратной неудачной попытке повторения операции процессора), неисправимая ошибка оперативной памяти, постоянная ошибка при восьмикратной попытке записи в оперативную память (ОП), неисправимая ошибка в буферной памяти (БП) и индексной памяти (ИП).

Машинные ошибки, зафиксированные в РгОш и регистре особых ошибок, при наличии соответствующих условий приводят к прерыванию от схем контроля. Прерывание от схем контроля является средством, с помощью которого программе сообщается о сбоях и отказах в аппаратуре, а также представляется информация о месте и степени повреждения.

Ошибки аппаратуры создают условия прерывания от схем контроля, которые могут быть подавляемыми и неотложными. Прерывание от схем контроля, вызванное подавляемым условием, не прекращает выполнения текущей команды. Такое прерывание выполняется после того, как нормально закончится выполнение текущей команды, и после обработки программных прерываний и прерываний по обращению к супервизору. Прерывание от схем контроля, вызванное неотложными условиями, прекращает выполнение текущей команды и отменяет другие прерывания.

Рассмотрим принципы контроля некоторых блоков и устройств ЭВМ ЕС-1045 [80].

Контроль оперативной памяти

В ЭВМ ЕС-1045, также как и в других моделях ЕС ЭВМ второй очереди, используется модифицированный код Хемминга, который обеспечивает исправление всех одиночных ошибок и обнаружение двойных ошибок. Ошибки в двух битах и более называются неисправимыми в отличие от одиночных, называемых исправимыми.

Использование восьми контрольных бит для 64-битной информации обеспечивает исправление одиночных ошибок

и обнаружение всех двойных. Кроме того, появляется возможность получения контрольных бит нечетности байт непосредственно из схемы генерации битов кода Хэмминга. Это минимизирует аппаратные средства и позволяет простым способом переходить от кода Хэмминга к контрольным битам нечетности. Последнее обстоятельство имеет очень важное значение, так как информация для записи в ОП поступает с битами нечетности. При записи в ОП биты нечетности отбрасываются и на их место записываются биты кода Хэмминга. При считывании из ОП отбрасываются биты кода Хэмминга, на место которых записываются биты нечетности байты, и в таком виде информация поступает в процессор на обработку.

В ЭВМ ЕС-1045 оперативная память имеет ширину выборки 144 бита, из которых 128 информационных и 16 контрольных. Обмен информацией с оперативной памятью, обнаружение и исправление ошибок выполняется адаптером оперативной памяти (АОП). Информационный тракт связи АОП с ОП разделен на две идентичные половины по 64 информационных и 8 контрольных бит в каждой. Для каждого 64-битного тракта имеются свои схемы кодирования и декодирования кода Хэмминга.

На рис. 5.14 показана структурная схема АОП. При выполнении операции записи информация принимается из процессора (или других устройств) и запоминается на регистре данных записи (РгДЗп). На РгДЗп возможно накопление до четырех слов, т.е. 144 бит с последующей одновременной записью их в ОП. Информация из РгДЗп через коммутатор выдачи данных записи (КВДЗп) и коммутатор приема АОП (КПАОП) поступает в регистр данных АОП (РгДАОП). Регистр данных АОП состоит из двух 72-битных частей, РгДАОП верхнего двойного слова (РгДАОПВС) и РгДАОП нижнего двойного слова (РгДАОПНС). В РгДАОП информация поступает с контрольными битами нечетности. С выходов РгДАОП информация через ассемблирующие коммутаторы верхнего и нижнего слова (АКВС, АКНС) поступает в генераторы кода Хэмминга (генераторы С-бит) верхнего и нижнего слова Г2ВС и Г2НС.

Биты контроля на нечетность обозначаются $K_0—K_7$, а биты кода Хэмминга $C_0—C_7$ ($C_0, C_1, C_2, C_4, C_8, C_{16}, C_{32}, C_7$). На структурной схеме (рис. 5.14) 8 контрольных бит нечетности обозначаются 8 К.

Формирование бит кода Хэмминга в генераторах Г2 происходит в соответствии с рис. 5.15. Биты кода Хэммин-

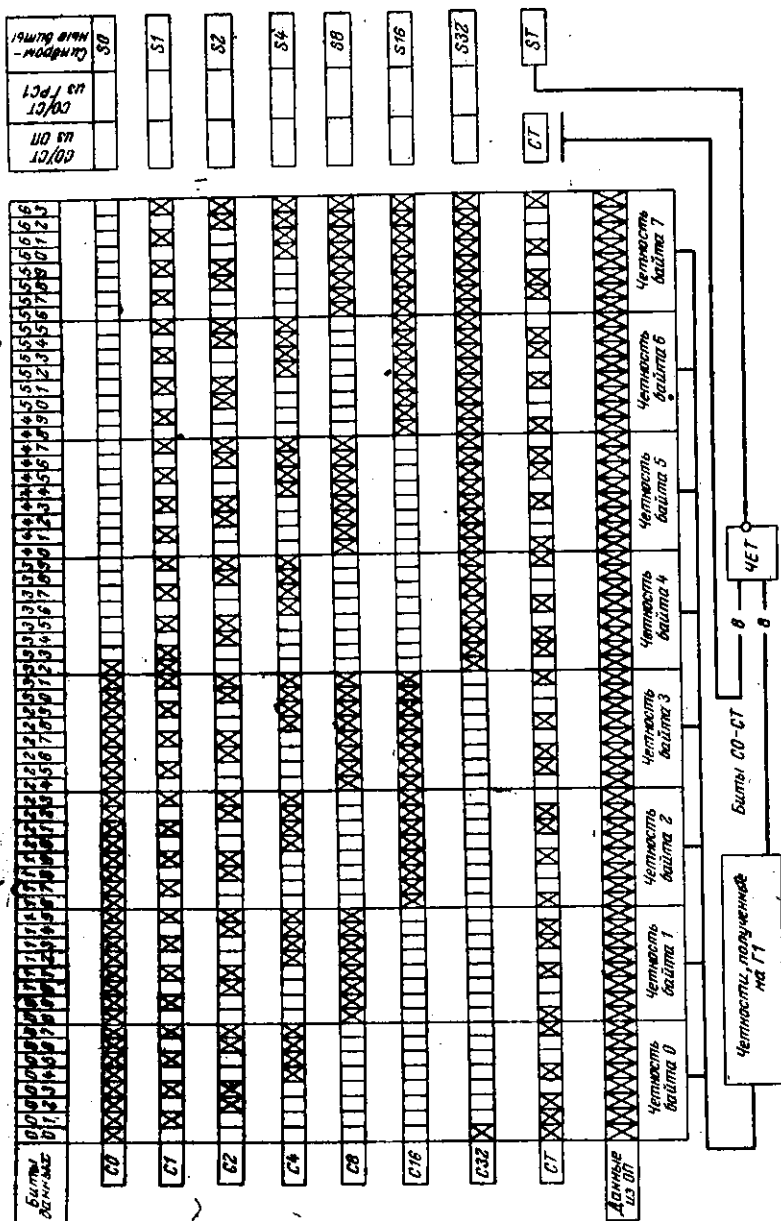


Рис. 5.15. Схема формирования кода Хэмминга и синдромов ошибки

га определяются как дополнение до нечетности суммы по модулю 2 бит информации, отмеченных крестиками в соответствующей строке таблицы на рис. 5.15. Например, если сумма по модулю 2 бит информации, отмеченных крестиками в строке *СО*, равна нулю, то бит *СО* кода Хэмминга получает значение 1 и наоборот.

В результате преобразования 8 бит нечетности двойного слова заменяются на биты *СО—СТ* кода Хэмминга. При этом биты кода Хэмминга *СО—СТ* двойного слова с целью сохранения байтной структуры потоков информации располагаются в позициях бит нечетности. Биты кода Хэмминга, сформированные в Г2, поступают в соответствующий коммутатор АК (см. рис. 5.14), с выхода которого в результате открывания передатчиков информация поступает на запись в ОП по двунаправленной магистрали.

При выполнении операции чтения из ОП считываются четыре слова (128 информационных и 16 контрольных бит). Эта информация поступает по двунаправленной магистрали в регистр данных ОП (РгДОП), также состоящий из двух половин — РгДОП верхнего слова (РгДОПВС) и РгДОП нижнего слова (РгДОПНС). На выходе каждой половины РгДОП располагаются генераторы кода Хэмминга Г1НС и Г1ВС. Информационные биты из РгДОП поступают в генераторы Г1 и на корректоры, а биты кода Хэмминга (*С*-биты) — на схему сравнения и корректор. В генераторе Г1 формируются также биты нечетности байт информации, которые при выдаче информации из АОП заменяют биты кода Хэмминга.

Полученные в генераторе Г1 биты *СО—СТ* сравниваются с соответствующими битами, считанными из ОП, в блоке обнаружения и декодирования верхнего слова (БОДВС) и нижнего слова (БОДНС). При несовпадении этих бит фиксируются ошибка и биты *СО—СТ* синдрома, определяющего вид ошибки и номер ошибочного байта/бита при одиночной ошибке. Совпадение *С*-бит устанавливает синдромный бит в 0, а несовпадение — в 1. Формирование синдромного бита *ST* показано на рис. 5.15. На первые восемь входов схемы определения четности подаются биты четности, полученные на Г1, на следующие восемь входов поступают *С*-биты, считанные из ОП. При нечетном числе единиц на входе этой схемы выходной сигнал устанавливает в единицу бит синдрома *ST*. Полученные биты синдромов *SO—ST* позволяют по табл. 5.1 и 5.2 определить наличие, тип и место ошибки [59]. Двойное слово, верхнее или нижнее, в котором произошла ошибка, определяется по

28-му биту регистра адреса ОП. Если он равен 0, то ошибка имела место в нижнем слове, т.е. в битах 0—63. Если он равен 1, то ошибка имела место в верхнем слове, т.е. в битах 64—127.

Таблица 5.1

ST	Биты S0—S32	Тип ошибки
0	Все нули	Нет ошибки
1	Все нули	Ошибка бита ST
1	Один бит или более равен 1	Исправимая ошибка (ошибочный бит определяется по табл. 5.2)
0	Один бит или более равен 1	Неисправимая ошибка

Адрес ОП, в котором обнаружена ошибка, и синдромы S0—ST, характеризующие эту ошибку, запоминаются соответственно на регистре хранения адреса ОП (РгХАОП) и регистре синдромов (РгСИНД) для последующей записи их состояния в журнал ошибок операционной системы. Состояние РгХАОП и РгСИНД опрашивается специальной операцией АОП, называемой «чтение кода коррекции ошибок» (ККО).

Обнаруженная в соответствии с табл. 5.2 одиночная ошибка корректируется в корректоре путем инверсии ошибочного бита. Вместе с тем в корректоре производится изменение бит нечетности того байта, в котором произведена коррекция.

Откорректированные данные из корректора через КВАОП, коммутаторы данных чтения КДЧт1 и КДЧт2 поступают в регистр данных чтения (РгДЧт), а из него — в схемы процессора.

Для проверки исправности аппаратуры кода Хэмминга информация передается также в АК, с тем чтобы на Г2 выработать дубликат бит нечетности K0—K7 и сравнить их с битами нечетности, сформированными в Г1. При несопадении этих бит формируется сигнал ошибки аппаратуры контроля по коду Хэмминга (АККХ), поступающий на вход схемы общего сигнала ошибки АОП.

Одиночная ошибка ОП вызывает установку бита регистра особых случаев и через него ТгНтПрК, т.е. вызывает подавляемое условие прерывания от схем контроля. Случаи одиночных ошибок обрабатываются операционной системой (ОС) для накопления статистики.

Таблица 5.2

Синдромы				S1	0	1	0	1	0	1	0	1	1
				S2	0	0	1	1	0	0	1	1	2
				S4	0	0	0	0	1	1	1	1	4

Веса синдромных бит¹

S32	S16	S8	S0	Бит	0	1	2	3	4	5	6	7
				Байт								
0	0	0	0		СТ	C1	C2		C4			
0	0	0	1	0	CO	1	2	3	4	5	6	7
0	0	1	0		C8							
0	0	1	1	1	8	9	10	11	12	13	14	15
0	1	0	0		C16							
0	1	0	1	2	16	17	18	19	20	21	22	23
0	1	1	0									
0	1	1	1	3	24	25	26	27	28	29	30	31
1	0	0	0	4	C32	33	34	35	36	37	38	39
1	0	0	1		0	32						
1	0	1	0	5	40	41	42	43	44	45	46	47
1	0	1	1									
1	1	0	0	6	48	49	50	51	52	53	54	55
1	1	0	1									
1	1	1	0	7	56	57	58	59	60	61	62	63
1	1	1	1									
4	2	1	0									

Веса синдромных бит²

Примечания: 1. Сумма весов единичных синдромных бит показывает номер ошибочного бита.

2. Сумма весов единичных синдромных бит показывает номер байта, содержащего ошибку.

Пример (к табл. 5.2)

Синдромы ошибки

S0	S1	S2	S4	S8	S16	S32
0	0	0	1	0	0	1

Ошибка в бите
36 (байт 4)

Неисправимая ошибка ОП вызывает установку бита ошибки памяти в РгОш, бита неисправимой ошибки в регистре особых ошибок и неотложное условие прерывания от схем контроля. Для проверки работоспособности тракта процессор — ОП, а также самой аппаратуры контроля по коду Хэмминга в ЭВМ ЕС-1045 имеется специальный режим записи и чтения ОП с заблокированной аппаратурой кода Хэмминга. Этот режим используется при диагностировании и позволяет:

проверить работу оперативной памяти без коррекции одиночных ошибок, т. е. используя только контроль на четность;

имитировать одиночную и двойную ошибки в оперативной памяти путем записи информации в ОП с битами нечетности, т. е. с заблокированной аппаратурой контроля по коду Хэмминга и чтением ее в нормальном режиме, т. е. с работающей аппаратурой контроля по коду Хэмминга. Этот режим позволяет имитировать одиночные и двойные ошибки и проверять работу схем фиксации и коррекции этих ошибок. Рассмотрим в качестве примера имитацию одиночной ошибки в ОП. Запись в ОП 8 байт информации: 40 00 00 00 00 00 00 00 (шестнадцатичный код) по адресу, кратному восьми, с заблокированным кодом Хэмминга приводит к записи бит нечетности $K0—K7$ на место бит $C0—C7$ в ОП. Таким образом, в ОП запишется информация с битами $C0—C7=01111111$, так как $K0=0$, $K1=K2=K3=K4=K5=K6=K7=1$. Считывание информации из ОП с включенной аппаратурой контроля по коду Хэмминга приведет к тому, что генератор кода Хэмминга (Г) сформирует по информации следующие биты кода Хэмминга:

$$C0 = C1 = 0; \quad C2 = C4 = C8 = C16 = C32 = C7 = 1.$$

В этом легко убедиться, пользуясь табл. 5.2. Сравнение бит $C0—C7$, считанных из ОП и сформированных генератором кода Хэмминга, приводит к формированию синдромов:

$C0$	$C1$	$C2$	$C4$	$C8$	$C16$	$C32$	$C7$	
0	1	1	1	1	1	1	1	Из ОП
0	0	1	1	1	1	1	1	Из генератора кода Хэмминга

Синдромы 0 1 0 0 0 0 0 0

Таким образом, $S0=0$, $S1=1$, $S2=0$, $S4=0$, $S8=0$, $S16=0$, $S32=0$, $S7=0$. По табл. 5.1 и 5.2 определяем, что этому синдрому соответствует исправимая ошибка в бите

С1. Эта ошибка будет скорректирована в корректорах аппаратуры контроля по коду Хэмминга, и в процессор поступит информация с откорректированной нечетностью байт.

Признак одиночной ошибки будет также зафиксирован в регистре особых ошибок (РгП, см. рис. 5.13).

Контроль устройства управления оперативной памяти

Как видно из рис. 5.13, общая ошибка блока управления памятью формируется схемами ИЛИ на выходе регистра ошибок управления памятью, на вход которого поступают сигналы ошибок от схем контроля на нечетность, расположенных в блоках динамического преобразования адресов, управления обменом информацией, буферной и индексной памятью, памяти ключей защиты и буферной памяти команд.

Общий сигнал ошибки блока управления памятью вместе с сигналами ошибки адаптера оперативной памяти и неисправимой ошибки формирует общий сигнал ошибки памяти.

Сигналы неисправимой и одиночной ошибок памяти формируются аппаратурой кода Хэмминга.

Ошибка адаптера оперативной памяти формируется схемами контроля на нечетность адаптера. Если ошибка АОП возникает во время операции записи, то ее выдача в схему формирования общей ошибки памяти блокируется на время восьмикратной попытки перезаписи. Если одна из попыток записи была успешной, т. е. ошибок АОП не возникло, то выдача сигнала «ошибка АОП» разрешается, и этот сигнал поступает в схему формирования общей ошибки памяти. Это вызовет останов ЭВМ для регистрации сбойной ситуации и повторение выполнявшейся команды.

Если все попытки перезаписи были безуспешны, т. е. ошибка АОП возникала при каждой попытке записи, то кроме сигнала ошибки АОП формируется также сигнал «постоянная ошибка при восьмикратной попытке записи в ОП», который поступает в регистр особых ошибок. При этом выполнение команды прекращается и выполняется неотложное прерывание от схем контроля.

Контроль процессора

Основным методом контроля в процессоре является контроль на нечетность. Путем контроля на нечетность проверяются все передачи между регистрами. В наиболее ответственных местах информационных трактов, где неисправ-

ность самих схем контроля может привести к пропаданию сигнала ошибки или невозможности восстановления вычислительного процесса при сбоях, используются самопроверяемые схемы контроля. В ЭВМ ЕС-1045 самопроверяемые схемы контроля на нечетность реализованы, как показано на рис. 5.16. Выходные сигналы f_1 и f_2 самопроверяемой схемы контроля на нечетность соответствуют: первый — выходному сигналу схемы сложения по модулю 2 всех инфор-

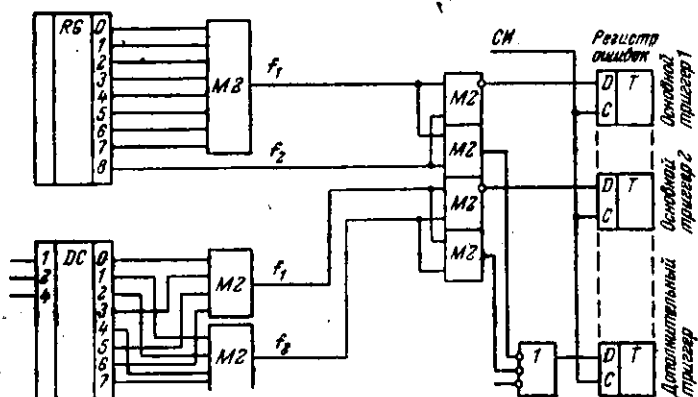


Рис. 5.16. Самопроверяемые схемы контроля на нечетность и контроля дешифратора

мационных бит байтного регистра, второй — выходному сигналу контрольного бита нечетности. Сигналы f_1 и f_2 поступают от контролируемой схемы в ТЭЗ, где расположен регистр ошибок (РгОш). В ТЭЗ РгОш сигналы f_1 и f_2 поступают на две идентичные схемы сложения по модулю 2, которые формируют сигналы ошибки. Выход первой схемы по модулю 2 соединен со входом основного триггера ошибки, а выход второй — через схему ИЛИ со входом дополнительного триггера ошибки.

Благодаря такой организации самопроверяемых схем контроля на нечетность достигается:

экономичная реализация самопроверяемой схемы контроля на нечетность;

исключение возможности пропадания сигнала ошибки на всем пути его распространения вплоть до выхода РгОш.

Для экономии аппаратуры дополнительный триггер ошибки в РгОш разделяется между несколькими основными триггерами РгОш, т.е. для нескольких основных триг-

геров в РгОш имеется один дополнительный триггер со схемой ИЛИ на входе.

На рис. 5.16 показана также реализация самопроверяемой схемы контроля дешифратора. На входы первой и второй схем сложения по модулю 2 поступают выходные сигналы дешифратора, появляющиеся при четном и нечетном числах единиц на входе соответственно.

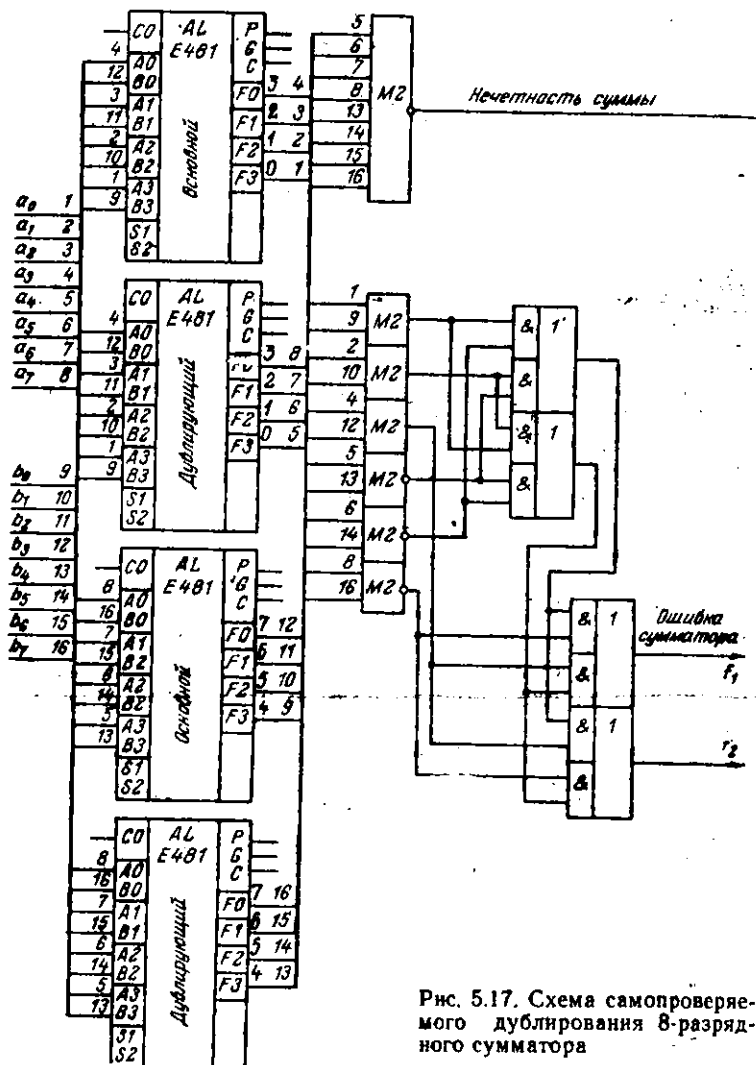


Рис. 5.17. Схема самопроверяемого дублирования 8-разрядного сумматора

Передача сигналов f_1 и f_2 с выхода дешифратора в РгОш аналогична описанной ранее схеме контроля на нечетность.

Для контроля основного (4-байтного) и 1-байтного сумматоров используется метод самопроверяемого дублирования. Сумматоры процессора ЭВМ ЕС-1045 построены на интегральных микросхемах (ИС), представляющих собой 4-разрядный арифметическо-логический блок. Логическая схема 8-разрядного сумматора приведена на рис. 5.17. Сумматор состоит из двух основных 4-разрядных ИС арифметическо-логического устройства (АЛУ) и двух дублирующих. Входные операнды поступают одновременно на входы основного и дублирующего сумматоров. Входные операнды проверяются схемами контроля на нечетность на входах сумматора. Сигналы с выходов разрядов 0 и 4, 1 и 5, 3 и 7 основного сумматора поступают на входы первой, второй и третьей схем сложения по модулю 2 соответственно. Сигналы с аналогичных выходов дублирующего сумматора поступают на входы четвертой, пятой и шестой схем сложения по модулю 2. Выходные сигналы первой и четвертой, второй и пятой, третьей и шестой схем сложения по модулю 2 образуют пары выходных сигналов схемы самопроверяемого дублирования и поступают на схемы сжатия, на выходе которых формируется общий сигнал ошибки сумматора. Этот сигнал поступает на схемы сложения по модулю 2, стоящие на входе РгОш, и далее на основной и дополнительные триггеры ошибки сумматора.

Суммарный сигнал поступает в регистр результата. Контрольный бит нечетности суммы формируется схемой сложения по модулю 2, стоящей на выходах дублирующего сумматора. Регистр результата контролируется на нечетность самопроверяемой схемой контроля. Такая организация контроля сумматора позволяет ограничиться сравнением на схемах сложения по модулю 2 не всех бит конечной суммы¹.

В состав процессора ЭВМ ЕС-1045 входит акселератор— ускоритель выполнения некоторых операций, основанный на табличном способе получения результата. Акселератор реализован на интегральных микросхемах ППЗУ, в которых хранится таблица результатов. Результирующий сигнал получается на выходе ППЗУ в зависимости от входных сигналов, представляющих собой биты операндов. Переда-

¹ Описанные выше реализации самопроверяемых схем контроля на нечетность и контроля сумматора методом самопроверяемого дублирования были предложены К. А. Кучукян [37].

чи и операции сдвига в акселераторе контролируются на нечетность, десятичное умножение и преобразование в десятичную систему — по модулю 9, а двоичное умножение и преобразование в двоичную систему — по модулю 15. Контроль операций умножения выполняется путем умножения остатков и сравнения с остатком произведения. Умножение остатков и вычисление остатка произведения выполняется табличным способом с помощью ИС ППЗУ.

Дополнительным средством контроля процессора является ограничитель времени простоя процессора (ОВПП), предотвращающий заикливание микропрограмм процессора. В качестве счетчика времени простоя используется аппаратура часов. Сигнал превышения допустимого времени простоя вырабатывается при работе процессора в том случае, когда до поступления сигнала «конец команды» успевают поступить два последовательных сигнала «пуск ОВПП» с интервалом 16 с. Этот сигнал фиксируется в регистре ошибок управления памятью и далее в виде общей ошибки памяти поступает в РгОш процессора, вызывая неотложное условие прерывания от схем контроля.

Для проверки работоспособности схем реакции на машинную ошибку в РгОш предусмотрены специальные триггеры имитированных ошибок.

Эти триггеры могут устанавливаться в режиме тестирования при выполнении любой микрокоманды, адрес которой задается либо с пульта управления ЭВМ, либо из управляющего слова команды ДИАГНОСТИКА.

Контроль каналов

Ошибки каналов условно делятся на три группы:

ошибки общей аппаратуры каналов;

ошибки аппаратуры, принадлежащей конкретному каналу;

ошибки общей аппаратуры каналов, имеющей микропрограммное управление.

Эти ошибки фиксируются на триггерах ошибок каналов, условно объединенных в регистр ошибок. С выхода регистра ошибок каналов в РОш процессора поступают два сигнала: общая ошибка каналов 0—2 и общая ошибка каналов 3—5.

Возникновение ошибки в канале вызывает установку общей ошибки каналов в РгОш, а также триггера ошибки и триггера прерывания от ошибки соответствующего канала. Следствием установки РгОш является переход к микропрограммам обработки ошибок и в последующем к прерыва-

нию от схем контроля, а следствием установки триггера прерывания канала по ошибке — прерывание ввода-вывода от соответствующего канала с ограниченной регистрацией состояния канала.

Ограниченная регистрация состояния канала содержит информацию об источнике ошибок и код последовательно-сти, показывающий, на какой стадии выполнения приказа находилось периферийное устройство.

Прерывание от схем контроля служит для регистрации состояния ЭВМ в момент ошибки в системном журнале ошибок ОС, а прерывание ввода-вывода — для регистрации состояния канала в момент ошибки и повторения прерванной операции ввода-вывода.

Триггер ошибки и прерывания по ошибке канала устанавливается и в том случае, когда машинная ошибка возникла не в канале. Необходимость в этом возникает вследствие того, что при машинной ошибке выполняется регистрация состояния ЭВМ в служебную память при остановленных синхросигналах ЭВМ. Регистрация длится несколько миллисекунд, и поскольку в момент регистрации приостанавливается синхронизация каналов, возможно возникновение ошибок в устройствах и канале, работавших в момент ошибки. Установка триггера ошибки и прерывания по ошибке в работавшем канале позволяет повторить выполнявшиеся приказы с помощью средств восстановления ОС.

Глава шестая

АВТОМАТИЧЕСКОЕ ВОССТАНОВЛЕНИЕ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА ПОСЛЕ МАШИННЫХ ОШИБОК

6.1. ТИПЫ МАШИННЫХ ОШИБОК И ИХ ВЛИЯНИЕ НА ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕСС

Из материалов гл. 4 следует, что для достижения высоких значений эксплуатационной надежности ЭВМ должна иметь средства (аппаратурные, микропрограммные, программные), позволяющие автоматически восстанавливать процесс выполнения программы после значительной части возможных ошибок в работе машины.

По степени воздействия на вычислительный процесс ошибки делятся на «мягкие» и «жесткие». *Мягкими* называются машинные ошибки, последствия которых удастся ав-

томатически устранить и восстановить выполнение программы без вмешательства оператора. Мягкие ошибки вызываются, как правило, сбоями в работе ЭВМ. Ошибка в работе ЭВМ называется жесткой, если аппаратурные и программные средства ЭВМ не в состоянии автоматически восстановить выполнение программы и для восстановления вычислительного процесса может потребоваться реконфигурация ЭВМ с ручным отключением отказавшего блока, перезагрузка операционной системы или иное вмешательство обслуживающего персонала.

Ошибку, вызванную отказом, но допускающим автоматическое восстановление путем автоматической реконфигурации, будем относить к мягким ошибкам.

При жесткой ошибке в работе ЭВМ операционная система пытается осуществить восстановление программными средствами. Существуют следующие уровни программного восстановления:

- 1) полное;
- 2) с ухудшением характеристик;
- 3) с завершением поврежденной задачи;
- 4) путем перезагрузки операционной системы и запуска заданий с контрольной точки благодаря сохранению очередей заданий;
- 5) путем ремонта.

По степени проявления ошибки делятся на явные и скрытые. *Явными* называются ошибки, обнаруживаемые сразу после их возникновения. Эти ошибки не имеют тяжелых последствий, так как обнаруживаются до их распространения. *Скрытыми* называются ошибки, которые сразу не обнаруживаются и могут вызывать искажения данных и неправильное функционирование машины.

Скрытые ошибки характеризуются *периодом проявления*, т.е. временем между появлением ошибки и ее обнаружением. После обнаружения скрытой ошибки, которая может быть выявлена пользователем или с помощью периодически выполняемых тестов, необходимо восстанавливать данные, которые могли быть искажены. Последнее имеет особенно важное значение в системах коллективного доступа к базам данных.

Предполагается, что время, необходимое для восстановления целостности системы, пропорционально периоду проявления скрытой ошибки. Это объясняется тем, что возникает необходимость проверки каждого действия системы в период проявления.

Для достижения высокой производительности машины надо стремиться к тому, чтобы возможно большая часть ошибок машины проявлялась как мягкие ошибки.

Все или часть отказов в аппаратуре машины удается перевести в псевдомягкие ошибки на основе реализации в архитектуре ЭВМ принципа «постепенной деградации». Этот принцип заключается в том, что при отказе в некоторой части оборудования система выполняет автоматическую реконфигурацию комплекса оборудования ЭВМ — логически отключает неисправное устройство, восстанавливает работоспособное состояние и продолжает нормальную работу с ухудшенными характеристиками (с меньшей производительностью и логическими возможностями) до тех пор, пока неисправность не будет устранена. Реконфигурация может состоять из таких этапов, как перемещение ОС в область памяти, где нет отказов, переотображение адресов памяти с целью исключения отказавшего модуля, исключение неисправных блоков и даже процессоров в многопроцессорных системах.

В общем случае восстановление заключается в восстановлении управления, целостности данных и упорядоченном повторном запуске системы. Характер восстановительных процедур зависит от типа вычислительной установки и режима ее работы. Так, в системах коллективного доступа к базам данных требуется гарантия сохранности баз данных. Восстановление баз данных достигается путем восстановления их исходного состояния, повторного выполнения заданий, обновления запасных копий баз данных. Подробно эта проблема рассматривается в гл. 7.

Данная глава посвящена процедурам восстановления вычислительного процесса при явных ошибках, вызванных сбоями и отказами.

При возникновении сигнала ошибки от схем контроля необходимо распознать (классифицировать) типы ошибки (сбой или отказ). Если произошел сбой, то осуществить восстановление по сбою. Если произошел отказ, то последний надо классифицировать на допускающий и не допускающий восстановление путем автоматической реконфигурации. Если реконфигурация невозможна, то система восстановления должна вырабатывать сообщение оператору о необходимости его вмешательства в процесс восстановления (например, о необходимости ремонта).

Классификация ошибки на сбой или отказ производится путем многократного повторения операции, при выполнении которой произошла ошибка.

Если одно из повторений ошибочной операции завершается правильно, ошибка классифицируется как сбой, производится регистрация информации о сбое и соответствующем состоянии ЭВМ и восстанавливается выполнение программы. Если при повторениях не удастся выполнить операцию правильно, ошибка классифицируется как отказ и после регистрации данных о состоянии машины на основе этих данных выносится решение о возможности реконфигурации для «обхода» отказа в аппаратуре.

Особых процедур требует восстановление по ошибкам в периферийных устройствах с движущимся носителем.

6.2. АВТОМАТИЧЕСКОЕ ВОССТАНОВЛЕНИЕ - ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА ПОСЛЕ СБОЕВ

Практика показывает, что большая часть ошибок при работе ЭВМ (по имеющимся данным — около 75 %) вызвана сбоями.

Сбои могут быть результатом флуктуаций напряжений источников питания, удара, внешних помех, статических зарядов, совпадения тяжелых режимов, напряженной временной диаграммы блоков ЭВМ и т. д. Последствия таких сбоев для систем, не снабженных средствами восстановления, носят часто тяжелый характер, так как вызывают переход операционной системы в состояние «тяжелого останова», когда необходима ее перезагрузка. Способность ЭВМ противостоять воздействию сбоев на процесс решения задач определяется устойчивостью ЭВМ к сбоям.

Методы автоматического восстановления вычислительного процесса после сбоев делятся на *программные, аппаратурно-микропрограммные и аппаратурные*. Выбор метода автоматического восстановления зависит от типа сбоя, который определяется местом его возникновения. Различают следующие типы сбоев, обнаруживаемых средствами аппаратурного контроля ЭВМ: сбой памяти, процессора, канала и периферийного устройства.

Для восстановления после сбоя памяти применяют аппаратурный метод восстановления: коррекцию одиночных ошибок памяти с помощью различных корректирующих кодов.

Для восстановления после сбоев процессора используются программные, аппаратурно-микропрограммные методы восстановления.

К программным методам относятся метод контрольных точек и программный метод повторения команд. При ис-

пользовании метода контрольных точек периодически запоминается информация о текущем состоянии процесса с целью продолжения работы после сбоя с той точки, в которой произошло запоминание информации.

Программный метод повторения команд обеспечивает восстановление всей необходимой информации для повторения команды, которая не была выполнена из-за сбоя, и повторное ее выполнение.

Основной проблемой при повторении команд процессора является изменение исходных данных (операндов) команды. Так, например, многие команды процессора выполняются в соответствии с формулой

операнд 1 $\odot$ операнд 2 $\Rightarrow$ операнд 1;

где $\odot$ — символ операции.

Для повторения команд при сбое необходимо сохранение операндов.

Повторение неправильно выполнившейся операции может осуществляться как микрокомандное повторение, когда архитектура машины такова, что контролируется правильность каждой микрооперации, сохраняются без искажений операнды микроопераций и в случае возникновения ошибки при выполнении микрооперации делается попытка повторить ее заданное число раз. Такой метод микрокомандного повторения реализован в управляющем вычислительном комплексе повышенной живучести [24].

Однако такой подход, вполне обоснованный для управляющих комплексов с повышенными требованиями к живучести, обнаруживает недостатки при применении его к высокопроизводительным машинам общего назначения из-за дополнительных аппаратных затрат и снижения быстродействия из-за потерь времени на передачи, связанные с сохранением операндов микроопераций.

Обычно при выполнении команды происходят события (например, изменение значения операнда в памяти), влияющие на процедуру восстановления. Такие события называются порогами. Система восстановления должна иметь средства контроля прохождения порогов.

В командах процессора имеется *порог повторения*, при прохождении которого исходные данные команды изменяются. Обычно этот порог фиксируется специальным триггером изменения исходных данных. При возникновении сбоя и регистрации состояния ЭВМ в момент сбоя по состоянию этого триггера программа восстановления может распознать, возможно повторение команды или нет. Если порог

повторения не пройден, выполняется повторение команды. В противном случае необходимо повторение программы с последней контрольной точки.

Сравнение программных и аппаратурно-микропрограммных средств восстановления показывает, что недостатками первых являются трудоемкость их разработки, непроизводительные затраты времени на организацию и хранение промежуточных данных, усложнение ОС. Аппаратурно-микропрограммная реализация функции восстановления является отражением общей тенденции аппаратурной поддержки некоторых частей ОС. Благодаря ей упрощается обслуживание ЭВМ, сокращается число остановов системы.

Аппаратурно-микропрограммные методы восстановления процессора после сбоя

Для обеспечения аппаратурно-микропрограммного повторения операции, выполнявшейся ЭВМ в момент сбоя, необходимо:

- сохранение информации об операции, выполнявшейся в ЭВМ в момент сбоя;

- устранение последствий сбоя, для того чтобы в дальнейшем они не вызвали других ошибок;

- восстановление состояний определенных индикаторов, имевших место до сбоя;

- наличие исходных данных для выполнявшейся операции;

- организация повторной попытки выполнения операции.

Последовательность действий при аппаратурно-микропрограммном восстановлении приведена на рис. 6.1. При обнаружении схемами контроля машинной ошибки производится «замораживание» состояния ЭВМ, т.е. прекращение выполнения команд на всех уровнях совмещения и регистрация состояния ЭВМ в оперативной или специальной памяти. Затем происходит восстановление адреса команды, во время выполнения которой произошел сбой и повторное ее выполнение.

При успешном восстановлении производится запись информации о сбое в системный журнал регистраций для ведения статистики и продолжается нормальная работа ЭВМ.

Обычно микропрограммы команд проверяют, пройден ли порог повторения, т.е. имело ли место изменение исходных данных. При изменении исходных данных команда выполняется по альтернативной микропрограмме [45].

Существуют следующие способы сохранения исходных данных:

а) запоминание исходных данных, которые могут быть изменены, в специальной памяти. В таком случае при возникновении сбоя измененные исходные данные восстанавливаются из специальной памяти и команда повторяется.

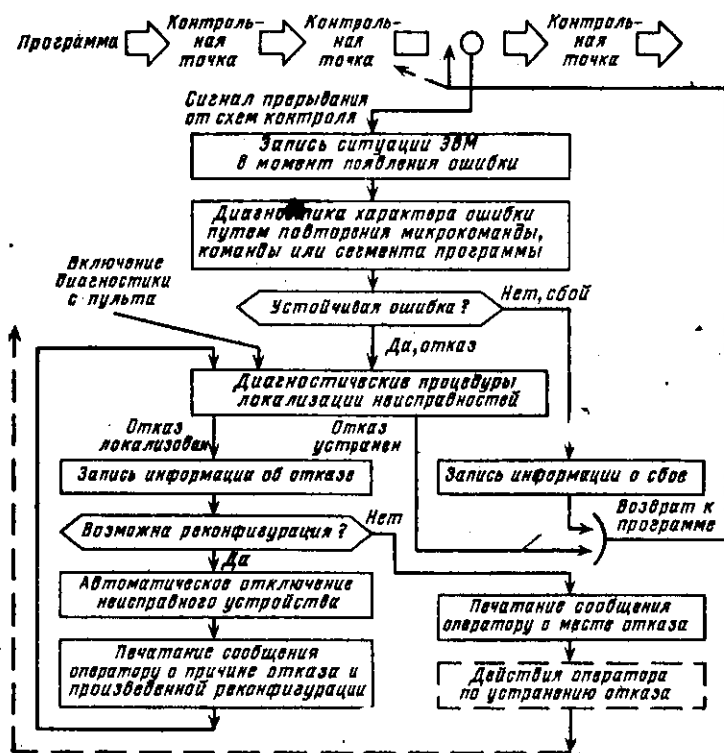


Рис. 6.1. Взаимодействие систем контроля и восстановления

Процедура выполнения таких команд в нормальном режиме и при сбое показана на рис. 6.2.

Сохранение исходных данных, особенно при длинных операндах, усложняет микропрограммы, увеличивает время выполнения команды. Способы сохранения исходных данных, позволяющие сократить объем запоминаемой информации, рассмотрены далее на примере команд процессора ЕС ЭВМ;

б) блокировка изменения первого операнда в командах формата *RR* и *RX*. В этих командах запись результата в общие регистры происходит в последней микрокоманде. Сохранность первого операнда достигается благодаря тому, что сигнал ошибки от схем контроля блокирует запись в общие регистры. Сама микрооперация записи выполняется с задержкой, достаточной для обнаружения ошибки схемами контроля процессора. Такое решение позволяет обойтись без запоминания исходных данных;

в) запоминание адреса второго операнда. Адрес второго операнда $[B2] + D2$ после вычисления запоминается в регистре хранения *X* (в рабочей области местной памяти). Запись в общие регистры в процессе выполнения команды вызывает установку триггера изменения исходных данных (ТИД). При возникновении ошибки команда повторяется согласно блок-схеме, приведенной на рис. 6.3, где дан при-

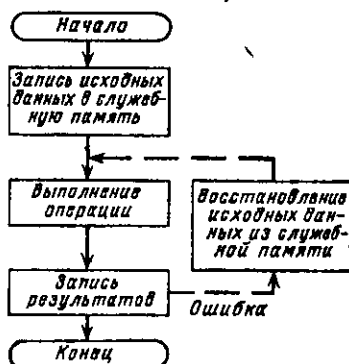


Рис. 6.2. Процедура повторения при запоминании исходных данных в служебной памяти

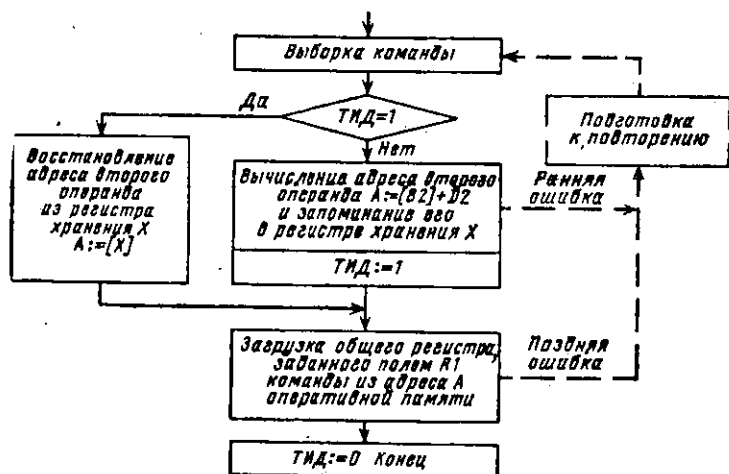


Рис. 6.3. Процедура повторения команды ЗАГРУЗКА

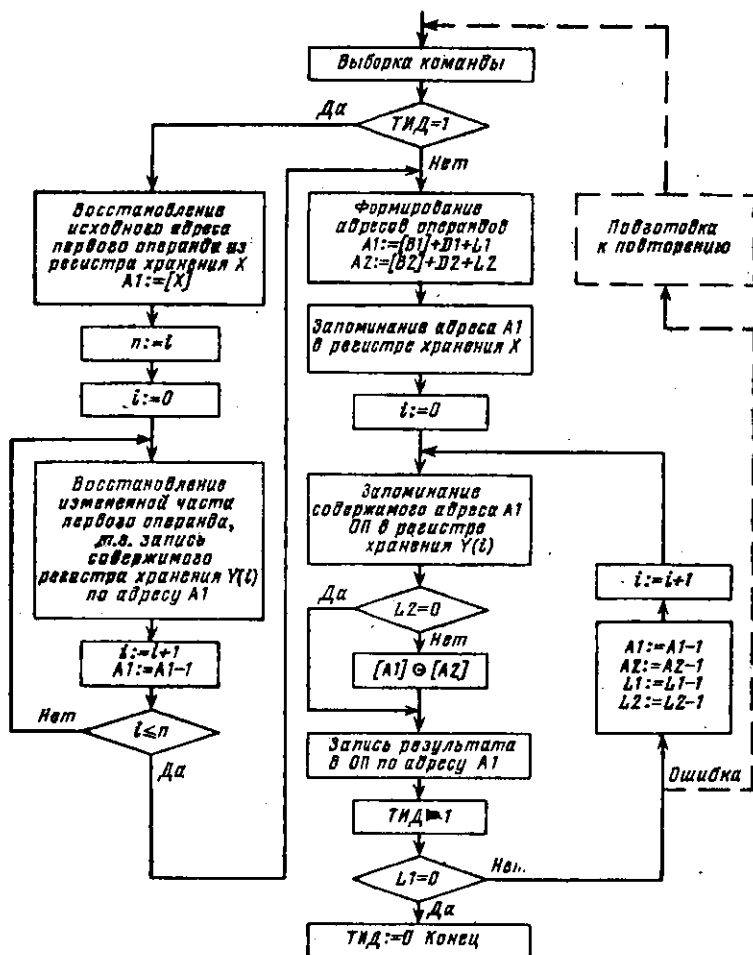


Рис. 6.4. Процедура повторения команд десятичной арифметики

мер повторения команды ЗАГРУЗКА. Ранняя ошибка (до установки ТИД) вызывает повторение команды без каких-либо изменений. Поздняя ошибка (после установки ТИД) вызывает ветвление в начале команды по ТИД=1 и восстановление эффективного адреса второго операнда из регистра хранения X. Такое решение позволяет отказаться от не-

обходимости сохранения содержимого регистра базы и его восстановления при повторении;

г) *запоминание изменяемой части первого операнда* с последующим восстановлением исходного значения при повторении по сбою. Это решение используется для повторения команд десятичной арифметики. Запоминание первого операнда по мере его обработки, а также счетчика байт производится в рабочей области местной памяти. При появлении ошибки после восстановления адреса команды и ее повторной выборки выполняется анализ ТИД (рис. 6.4), который был установлен в 1 при записи результата на место первого операнда в ОП. После ветвления по ТИД=1 микропрограммно выполняется восстановление измененных байт первого операнда, и команда выполняется сначала;

д) *запоминание счета успешно обработанных байт* с целью продолжения выполнения команды при повторении. Это решение используется для повторения логических команд формата SS. Состояние счетчика байт, который считает число обработанных байт, запоминается в рабочей области местной памяти (рис. 6.5). Запись информации в ОП приводит к установке ТИД. При повторении команды в результате ошибки состояние ТИД вызывает ветвление в микропрограмме команды. Для обновления адреса операнда используется значение счета успешно обработанных байт. При этом обработка начинается непосредственно с байта, на котором была зафиксирована ошибка;

е) *контрольные точки в микропрограммах команд*. В некоторых командах для фиксации этапа выполнения коман-

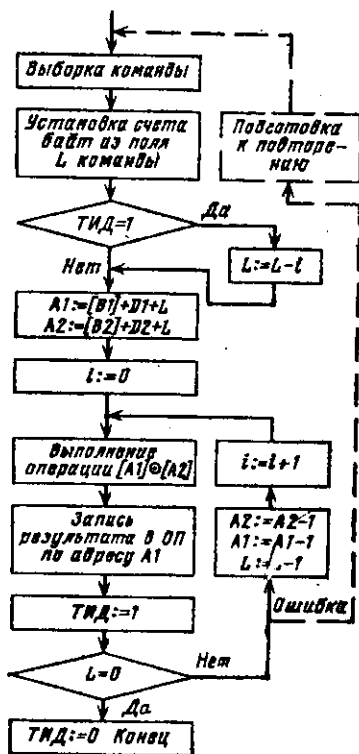


Рис. 6.5. Процедура повторения логических команд формата SS

ды используются контрольные точки, значения которых запоминаются в регистре контрольных точек. Ошибка, возникшая при выполнении команды, вызывает повторение, при котором восстанавливается состояние регистра контрольных точек, и выполнение команды продолжается с последней контрольной точки.

Повторение команд ввода-вывода и канальных команд

Основную трудность при повторении команд ввода-вывода вносит изменение положения носителя данных. Вследствие этого повторение команд ввода-вывода возможно при сочетании аппаратурно-микропрограммных и программных средств восстановления.

Повторение команд ввода-вывода на аппаратурно-микропрограммном уровне возможно до некоторого порога, определяемого началом связи с устройством по интерфейсу. Если порог аппаратурно-микропрограммного повторения команды ввода-вывода пройден, производится прерывание ввода-вывода и записываются слово состояния канала (ССК) и дополнительная информация, содержащие данные, уточняющие место, и условия, соответствующее ошибке. При этом восстановление выполняется на программном уровне с помощью специальных программ для соответствующего устройства, так как при этом требуется восстановление исходного положения носителя.

Упрощенная схема организации повторения команды SIO в ЭВМ ЕС-1045 показана на рис. 6.6. Для указания возможности повторения команды ввода-вывода предусмотрены триггеры порогов (на рис. 6.6 для простоты показан один триггер порога, определяющий возможность аппаратурно-микропрограммного повторения). Ошибки по отношению к этому порогу условно разделены на ранние и поздние и ошибки во время передачи данных. При возникновении машинных ошибок в каналах выполняется регистрация состояния ЭВМ и анализ порога повторения каналов.

Если повторение возможно, команда ввода-вывода повторяется. Если повторение невозможно, каналы, затронутые ошибкой, записывают свои слова состояния и дополнительную информацию об ошибке.

Если повторение было возможным и прошло успешно, для регистрации состояния ЭВМ в регистрационном файле выполняется прерывание от схем контроля. Машинная ошибка во время передачи данных вызывает прерывание ввода-вывода с записью ССК и дополнительной информации, содержащей код повторения, который показывает, на

каком этапе выполнения находилась команда ввода-вывода. После этого восстановление выполняется на программном уровне с использованием программ восстановления.

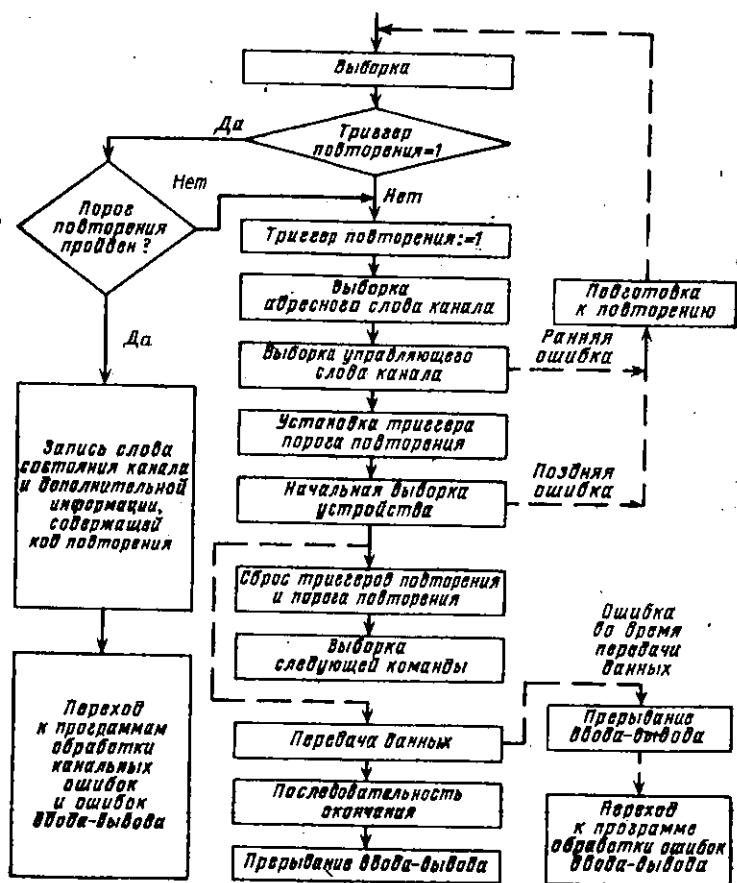


Рис. 6.6. Процедура повторения команд ввода-вывода

Восстановление после сбоев в периферийных устройствах

Восстановление от сбоев периферийных устройств (ПУ) обычно выполняется с помощью программных средств. Для этого в операционных системах для всех типов ПУ имеются программы обработки ошибок [64].

Программа обработки ошибок ПУ анализирует тип ошибки и при сбое устройства уточняет состояние ПУ и пытается исправить сбойную ситуацию путем многократного повторения начальной программы, при выполнении которой произошел сбой. Число повторений зависит от типа ошибки и ПУ.

6.3. АВТОМАТИЧЕСКОЕ ВОССТАНОВЛЕНИЕ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА ПОСЛЕ ОТКАЗОВ

Сохранение работоспособности ЭВМ при отказах имеет особенно большое значение при эксплуатации систем, работающих в реальном масштабе времени, систем с разделением времени, телеобработки и диалоговых систем. Отказ таких систем коллективного использования, которые обслуживают большое число пользователей и имеют большие базы данных, приводящий к прекращению операций с пользователями или к необходимости восстановления базы данных, может иметь катастрофические последствия. Для систем, работающих в режиме пакетной обработки, последствия отказов хотя и не очень чувствительны, тем не менее также приводят к существенным потерям. Известны способы повышения надежности и восстановления при отказах путем резервирования. Недостатком этих способов является, однако, их высокая стоимость, которая в ряде случаев делает подобные системы неприменимыми. В то же время существует ряд способов, которые позволяют существенно повысить надежность ЭВМ за счет аппаратурно-программных средств, обеспечивающих восстановление ЭВМ при отказах при минимальных дополнительных затратах. В последнее время получили распространение однородные структуры, реализованные на базе одинаковых модулей. Отказ модуля в такой системе вызывает реконфигурацию ЭВМ, и продолжение нормальной работы возможно с несколько ухудшенными характеристиками.

Наиболее известные следующие способы восстановления ЭВМ при отказах:

- исправление одиночных и некоторых двойных ошибок из-за отказов в оперативной памяти;

- реконфигурация оперативной памяти при отказе одного из ее модулей;

- исключение из работы отказавших ячеек оперативной памяти;

- переадресация в исправный блок;

- отключение части или всей буферной памяти при ее отказах;

исправление отказов постоянной управляющей памяти;
альтернативные алгоритмы выполнения операций;
реконфигурация периферийных устройств;
повторение операций ввода-вывода по альтернативному каналу.

Обычно в коде прерывания от схем контроля предусматривается бит УХУЖДЕНИЕ ХАРАКТЕРИСТИК, свидетельствующий о том, что восстановление после отказа привело к отключению какой-то части и ЭВМ работает с меньшей производительностью.

Обычно системы с постепенной деградацией характеризуются коэффициентом готовности, средней вычислительной мощностью (с учетом работы в «деградированном» режиме) и потерями времени на реконфигурацию и восстановление.

Отказ системы с постепенной деградацией имеет место при исчерпании ресурсов или при скрытых ошибках.

Средняя производительность ЭВМ с постепенной деградацией

$$M_{\text{ср}} = \sum_{i \in D} M_i p_i,$$

где M_i — производительность в i -м деградированном состоянии; p_i — вероятность нахождения в i -м деградированном состоянии; D — множество деградированных состояний системы.

В однородных структурах, где модули имеют одинаковую производительность, эта формула приобретает вид

$$M_{\text{ср}} = \sum_{i=1}^N i M p_i,$$

где N — число одинаковых модулей; i — число исправных модулей; M — производительность одного модуля, а p_i — вероятность исправности i модулей.

Исправление одиночных и некоторых двойных ошибок из-за отказов в оперативной памяти

Способ исправления одиночных ошибок в оперативной памяти описан в гл. 5 и широко известен. Обычно каждая ошибка в ЭВМ, в том числе и одиночная корректируемая ошибка памяти, фиксируется операционной системой в журнале регистрации состояния ЭВМ по ошибке. Однако

при постоянной одиночной ошибке памяти нет необходимости в регистрации информации об одной и той же ошибке. Поэтому в операционных системах предусматривается возможность перевода ЭВМ в так называемый режим умолчания, при котором одиночные ошибки не вызывают прерывания от схем контроля для регистрации состояния («ошибки»).

В связи с широким внедрением полупроводниковых оперативных памяти (ПОП) предположение о том, что в них возникают преимущественно одиночные ошибки, не соответствует действительности. Поэтому довольно часто используются также алгоритмы исправления двойных ошибок оперативной памяти [83]. Для описания одного из возможных алгоритмов исправления двойных ошибок памяти введем следующие обозначения: w — слово, записываемое в память; w_s — слово, считанное из памяти.

Исправление двойных ошибок выполняется следующим образом. При обнаружении (с помощью схем контроля и коррекции одиночных ошибок) в слове w_s двойной ошибки сигнал ошибки поступает в блок управления оперативной памяти и запускает процедуру исправления: в неисправную ячейку ОП записывается слово $\bar{w}_s$, т.е. инверсия бит считанного из памяти слова w_s . Далее после считывания записанной информации ($\bar{w}_s$), выполняется операция, которая определяет неправильные биты:

$$N = w_s \oplus (\bar{w}_s)_s,$$

где N — вектор, содержащий 1 в ошибочных битах.

Блок управления памяти на основании вектора N корректирует первый ошибочный бит и выдает слово w_s с первым исправленным битом на вход схемы коррекции одиночной ошибки, которая корректирует второй ошибочный бит.

Реконфигурация оперативной памяти при отказе одного из ее модулей

Оперативная память ЭВМ обычно состоит из нескольких одинаковых модулей. Например, память ЭВМ объемом 1 Мбайт может состоять из четырех модулей по 256 Кбайт. Если память не имеет расслоения, то ее адреса возрастают монотонно: первый модуль содержит адреса от 0 до 256 Кбайт, второй — от 256 до 512 Кбайт и т.д.

Оперативная память часто имеет переключатели, которые позволяют коммутировать любой модуль на место

другого. Рассмотрим, например, случай выхода из строя модуля памяти 1, показанный на рис. 6.7. С помощью переключателей производится перекоммутация модулей таким образом, что модуль 2 подключается на место моду-



Рис. 6.7. Реконфигурация памяти путем исключения неисправного модуля

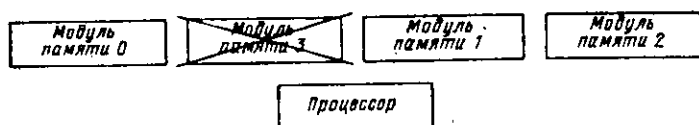


Рис. 6.8. Реконфигурация памяти путем исключения неисправного модуля и перекоммутации исправных

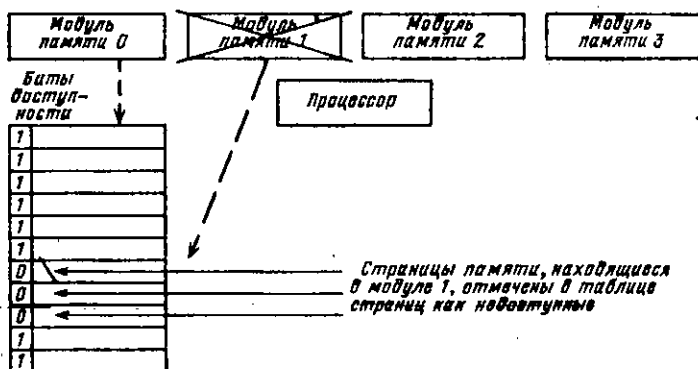


Рис. 6.9. Исключение отказавших физических страниц памяти путем указания бита недоступности

ля 1, модуль 3 — на место модуля 2, модуль 1 — на место модуля 3 и последний выводится из работы путем указания верхнего предела памяти, доступного операционной системе (рис. 6.8).

Имеется также другая возможность исключения отказавшего модуля. При инициализации операционной систе-

мы она распознает отключенный модуль памяти и заносит в таблицу распределения памяти указатели недоступности страниц памяти, находящихся в отказавшем модуле, как показано на рис. 6.9. Подобная организация реконфигурации предполагает возможность динамической переадресации памяти в операционной системе.

Такая возможность отсутствует в памяти с расслоением, так как в этом случае каждая n -я ячейка (в приведенных примерах каждая четвертая ячейка) недоступна.

Исключение из работы отказавшихся ячеек памяти

Исключение из работы отказавших ячеек памяти также предполагает использование динамической переадресации памяти при работе ОС. При обнаружении неисправных ячеек ОП программа обработки прерываний от схем контроля находит в таблице страниц ОС строку, соответствующую физической странице, в которой находятся отказавшие ячейки ОП, и устанавливает признак недоступности, как показано на рис. 6.9.

В этом случае как недоступная будет отмечена только одна страница.

Переадресация в исправный блок

Переадресация в исправный блок выполняется с помощью автоматической инверсии старшего бита адреса.

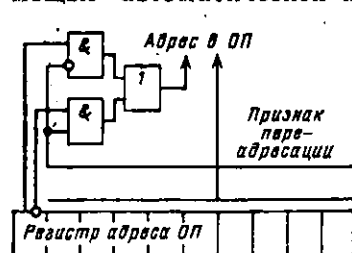


Рис. 6.10. Переадресация обращений в исправный блок ОП

При этом объем доступной памяти уменьшается вдвое, а все запросы к половине памяти, содержащей неисправность, переадресуется в другую половину. Обычно имеется возможность установки признака переадресации с пульта управления ЭВМ или с помощью команды **ДИАГНОСТИКА** (рис. 6.10).

Отключение части или всей буферной памяти при ее отказах

Буферная память (БП) предназначена для сокращения среднего времени выборки данных из ОП и, следовательно, повышения производительности центрального процессора. Блок буферной памяти позволяет осуществить быструю выборку данных. Считанная из ОП информация запо-

минается в БП. При этом объем считанных данных получается больше, чем это необходимо для одного обращения ЦП. Так как ЦП обычно использует данные, расположенные последовательно в ОП, то следующие обращения будут не к ОП, а к БП. Признак о наличии данных в БП хранится в индексной памяти (ИП). При записи ин-

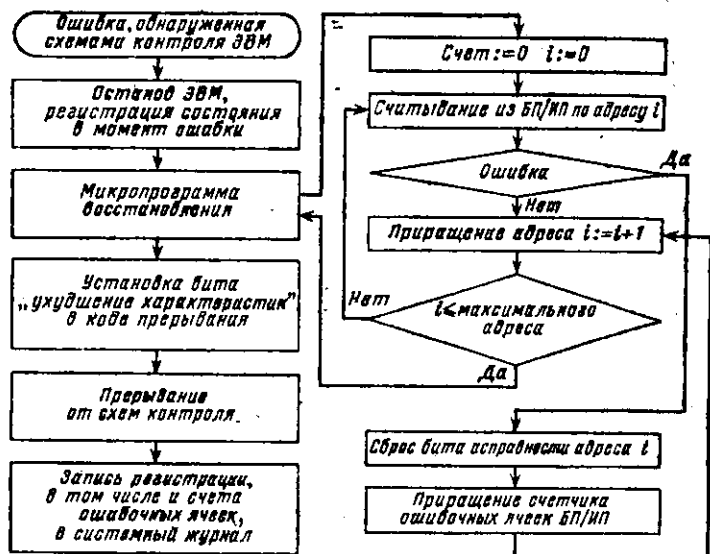


Рис. 6.11. Микропрограмма деградации буферной памяти

формации, считанной из ОП в БП, в индексную память записываются старшие биты адреса данных и устанавливается специальный бит действительности данных БП. Кроме того, каждая ячейка ИП имеет бит исправности ИП-и БП. Этот бит сбрасывается и блокирует доступ к ячейке БП, если обнаружена неисправность этой ячейки БП и соответствующей ей ячейки ИП. В этом случае считанная из ОП информация не записывается в неисправную область БП [80].

Процедура отключения ячеек БП и ИП выполняется следующим образом.

При обнаружении схемами контроля ЭВМ ошибки управление передается микропрограмме восстановления, которая в числе прочих действий выполняет последовательное обращение ко всем ячейкам БП/ИП. Если при этом схемы контроля обнаруживают ошибку, то бит исправно-

сти соответствующих ячеек. БП/ИП устанавливается в нуль (рис. 6.11). При этом в регистрации, выполняемой по прерыванию от схем контроля, записывается число неисправных ячеек БП/ИП, а в коде прерывания от схем контроля устанавливается бит ухудшения характеристик. Обычно предусматривается также возможность полного отключения БП/ИП с помощью специального триггера, который может быть установлен либо с пульта управления ЭВМ, либо с помощью команды ДИАГНОСТИКА. В указанных случаях ЭВМ полностью сохраняет работоспособность, но с несколько ухудшенными характеристиками, т. е. имеет место постепенная деградация ЭВМ.

Аналогичный принцип ухудшения характеристик может быть распространен и на другие внутренние памяти или блоки, обеспечивающие только повышение производительности ЭВМ.

Исправление отказов управляющей памяти, выполненной в виде ПЗУ

Если ПЗУ выполнено на микросхемах памяти МП, то при выходе из строя какой-либо ячейки ПЗУ ремонт МП, содержащей неисправную ячейку, затруднен. Для ремонта дефектной МП ее нужно выпаивать из ТЭЗ, заказывать на заводе-изготовителе новую МП ПЗУ и снова устанавливать ее в ТЭЗ. Неудобства подобной организации ремонта очевидны. Поэтому в таких случаях используют ПЗУ, допускающие эксплуатацию с неисправными адресами памяти [68].

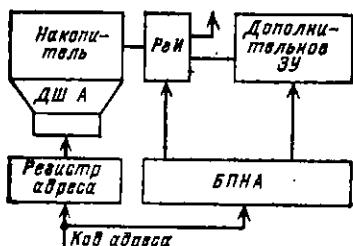


Рис. 6.12. Исправление отказов управляющей памяти на ИС ПЗУ

На рис. 6.12 приведена структурная схема ПЗУ, допускающего работу с неисправными адресами. При обнаружении постоянной ошибки в одном из адресов ПЗУ микропрограмма восстановления записывает правильное значение информации в дополнительное ЗУ и ЭВМ продолжает нормальную работу. При поступлении в регистр адреса ПЗУ кода адреса ячейки ПЗУ, содержащей ошибочную информацию, блок переадресации неисправных адресов (БПНА) переадресует обращение к дополнительному ЗУ, из которого в регистр информации (РгИ) считывается

правильная информация. Существует много вариантов исправления отказов ПЗУ. Более подробно эти варианты описаны в [68].

Альтернативные алгоритмы выполнения операций

Альтернативный алгоритм выполнения операций обеспечивает выполнение операции на другой аппаратуре при выходе из строя основной. Так, например, при неисправ-

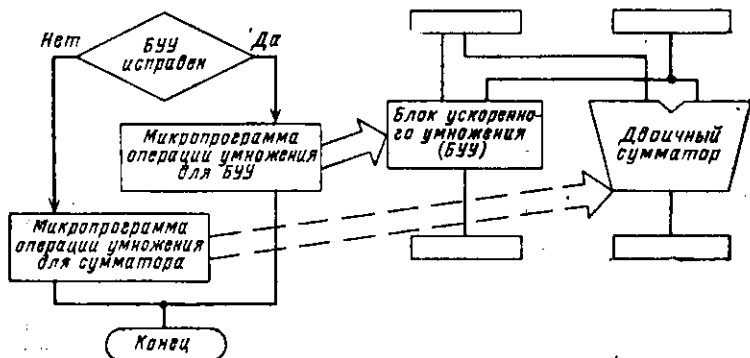


Рис. 6.13. Альтернативный алгоритм выполнения операций

ности в блоке ускоренного умножения (БУУ) операция умножения может выполняться с помощью альтернативной микропрограммы на основном сумматоре (рис. 6.13).

Реконфигурация периферийных устройств

В операционных системах, как правило, имеется программа динамической реконфигурации периферийных устройств, которая позволяет переносить тома на магнитных лентах и дисках с одного устройства на другое при отказе одного из внешних устройств [64]. Эта программа позволяет также переключать запросы на ввод-вывод с одного устройства на другое во время их использования операционной системой или пользователем. Обмен томов может быть запрошен при обнаружении неисправимой ошибки ввода-вывода операционной системой или оператором с помощью специальной команды. Перемещаемый том может быть резидентным или любым другим. Процедуры, выполняемые программой динамической реконфигурации, показаны на рис. 6.14.

Операционная система может запустить программу динамической реконфигурации НМЛ и НМД при следующих

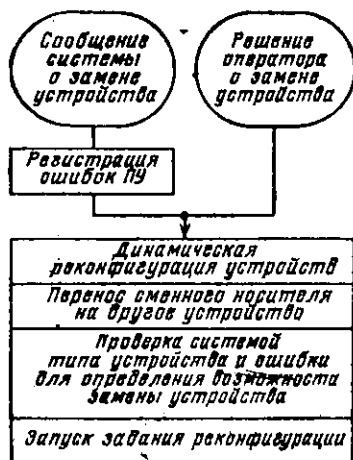


Рис. 6.14. Динамическая реконфигурация периферийных устройств

постоянных ошибках канала по другому пути доступа к периферийному устройству, т. е. по другому каналу.

6.4. СТРУКТУРНОЕ РЕЗЕРВИРОВАНИЕ

Структурным резервированием называется метод повышения надежности объекта путем включения избыточных структурных блоков, способных при отказе основного блока выполнять его функции [35, 49]. С появлением микропроцессоров и микро-ЭВМ, с которыми связано уменьшение стоимости и размеров вычислительных устройств, методы структурного резервирования получили более широкое распространение.

Основными способами резервирования являются:

- постоянное или статическое резервирование;
- резервирование с замещением или динамическое резервирование;
- гибридное резервирование.

Постоянным или статическим называется резервирование, при котором резервные блоки постоянно включены и находятся в одинаковом с основными режиме работы.

Резервированием с замещением или динамическим называется резервирование, при котором резервные блоки замещают основные блоки только при отказе последних.

типах постоянных ошибок: интерфейса, управления канала, данных канала, переполнения, а также ошибки поиска НМД.

Программа динамической реконфигурации поддерживает такие устройства, как НМЛ, НМД, ввод с перфокарт, вывод на перфокарты, алфавитно-цифровая печать.

Повторение операций ввода-вывода по альтернативному каналу

Программа повторения операции ввода-вывода по альтернативному каналу обеспечивает повторение операций ввода-вывода при

Гибридное резервирование сочетает свойства статического и динамического резервирования.

В системах со статическим резервированием восстановление работоспособности после отказа происходит мгновенно, без нарушения вычислительного процесса, а в системах с динамическим резервированием время восстановления вычислительного процесса в зависимости от степени автоматизации процедур восстановления изменяется в широких пределах.

Различают нагруженный, ненагруженный и облегченный резервы. *Нагруженным резервом* называется резервный блок, находящийся в том же рабочем режиме, что и основной. *Ненагруженным резервом* называется резервный блок, практически не несущий нагрузку. *Облегченным резервом* называется резервный блок, находящийся в менее нагруженном рабочем режиме, чем основной.

При динамическом и гибридном резервировании различают два способа включения резервных блоков: *логическое* и *по питанию*.

Включение по питанию предполагает использование ненагруженного резерва.

Логическое включение предполагает, что на резервный блок постоянно подано электропитание.

Считается, что интенсивность потока отказов электронных схем во включенном состоянии в 3—10 раз превышает интенсивность потока отказов в выключенном состоянии. С этой точки зрения предпочтительней схемы ненагруженного резерва и включения блоков по питанию. Недостатками таких схем являются их сложность и большое реальное время включения блока (в том случае, когда блоком является ЭВМ или процессор, необходимо значительное время для настройки их состояния на решение текущих задач).

Резервные блоки заменяют основные, когда принимается решение о неисправности основного блока, т. е. дополнительно требуется механизм обнаружения ошибок и замещения. Этот механизм может быть аппаратным, программным или аппаратурно-программным.

В зависимости от того, имеется в процессе эксплуатации возможность ремонта отказавших блоков или нет, различают *резервирование с восстановлением* и *без восстановления*. Системы, в которых используется резервирование с восстановлением (без восстановления), называются *обслуживаемыми* (необслуживаемыми) системами.

Для необслуживаемых систем разового действия наиболее важным параметром является вероятность безотказной работы, а для обслуживаемых — коэффициент готовности, являющийся функцией времени восстановления отказавшего блока.

Рассмотрим наиболее характерные избыточные структуры, используемые в вычислительных системах.

Дублированные системы

Дублирование ЭВМ или процессоров с целью создания систем повышенной надежности используется наиболее широко в силу относительной простоты структуры. При повышенных требованиях к достоверности информации в ЭВМ или процессорах с частичным контролем или при его отсутствии используется сравнение результатов перед их использованием. Сравнение может быть программным или аппаратным.

Подобные структуры относятся к структурам с динамическим резервированием, так как при отказе одной ЭВМ, определяемом по несовпадению результатов, выполняются процедуры проверки обеих ЭВМ с целью выявления отказавшей и только после этого принимается решение о переводе в ремонт отказавшей ЭВМ и продолжении работы системы в одномашинном режиме.

Если время восстановления t_b вычислительного процесса при этом не превышает некоторого порогового значения t_n ($t_b < t_n$), то отказа системы не происходит. В противном случае имеет место отказ системы.

Рассмотрим более подробно работу дублированной системы, состоящей из двух ЭВМ с частичным контролем ($\gamma < 100\%$) и программным сравнением результатов. Обычно в такой системе выдачу информации или управляющих воздействий выполняет основная ЭВМ, а резервная, решая ту же задачу, что и основная участвует только в сравнении результатов.

Диалог между ЭВМ и сравнение результатов выполняются следующим образом:

основная ЭВМ посылает в резервную свое слово состояния;

резервная ЭВМ отвечает основной своим словом состояния;

основная ЭВМ посылает в резервную результат для сравнения;

резервная ЭВМ сравнивает результаты, и при их совпадении нормальная работа системы продолжается.

Во время работы описываемой системы возможны ошибки двух типов:

- обнаруживаемые путем сравнения результатов;
- обнаруживаемые схемами контроля.

Если ошибка обнаружена при сравнении по несовпадению результатов, то резервная ЭВМ посылает в основную информацию о несовпадении, после чего обе ЭВМ выполняют возврат к контрольной точке и повторение вычислений.

После этого процедура обмена словами состояния повторяется. Если после этого результаты совпали, то ошибка считается сбоем и регистрируется. При повторном несовпадении в обеих ЭВМ вызывается программа опознавания ошибки, определяющая неисправную ЭВМ путем прогона тестов. ЭВМ, на которой тесты не прошли, устанавливает в слове состояния бит ошибки, а другая ЭВМ продолжает работу.

Если ошибка обнаружена схемами контроля, то в ЭВМ запускается процедура повторения, и если ошибка постоянная, то ЭВМ выводится из системы с установкой признака ошибки в слове состояния. Во время процедуры повторения на неисправной ЭВМ исправная не получит от нее слово состояния в ответ на свое слово состояния. После задержки, связанной с повторением, исправная ЭВМ снова посылает неисправной свое слово состояния и ожидает ответа. При отсутствии ответа исправная ЭВМ прочитает состояние другой машины и по анализу бита ошибки продолжит работу в одномашинном режиме.

В дублированной структуре из двух ЭВМ с аппаратурным сравнением результатов сравнение выполняет специальный блок управления (БУ), функциями которого являются: слежение за состоянием ЭВМ; сравнение результатов, поступающих от обеих ЭВМ; инициирование возврата к контрольной точке при сигнале ошибки; реконфигурация системы в одномашинный режим при отказе одной из ЭВМ.

При нормальной работе обе ЭВМ выдают данные на схему сравнения по заданию из выполняющейся программы. При несовпадении результатов или получении от ЭВМ сигнала ошибки блок управления запускает повторение программ с контрольной точки. Если ошибка от схем контроля повторяется, ЭВМ с ошибкой выводится из системы. Если происходит повторное несовпадение результатов, то на обеих ЭВМ запускаются тесты. По результатам тестирования неисправная ЭВМ выводится из системы.

Мажоритарные системы

Мажоритарными называются системы, в которых для определения правильных данных используется восстанавливающий орган (ВО), данные на выходе которого определяются по большинству совпадающих данных, подаваемых на его входы.

• Метод резервирования, основанный на использовании мажоритарных систем, называется *методом голосования* или *мажоритарным методом*.

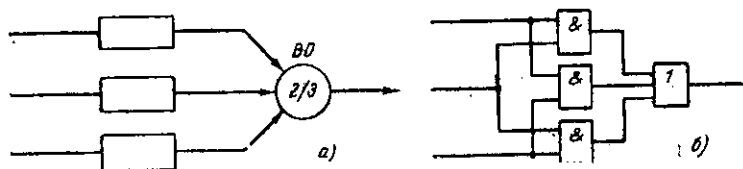


Рис. 6.15. Тройрованная мажоритарная система:

а — соединение блоков; б — схема восстанавливающего органа

Простейшей из мажоритарных систем является *тройрованная мажоритарная система* (ТМС), показанная на рис. 6.15, а. Восстанавливающий орган в этой системе представляет собой мажоритарную схему с тремя входами (рис. 6.15, б), значение сигнала на выходе которой совпадает со значением большинства входных сигналов (два из трех).

Голосование в ТМС может быть программным или аппаратным. Рассмотрим ТМС, состоящую из трех ЭВМ с частичным контролем и программным голосованием. В такой системе результаты трех ЭВМ сравниваются и в качестве правильного результата используются два одинаковых результата из трех. ЭВМ, результат которой не совпал с двумя другими, выводится из системы, т. е. система продолжает функционировать как дублированная. Обычно в ТМС с программным голосованием среди ЭВМ устанавливается приоритет 1—3 в соответствии с номером ЭВМ и голосование выполняет ЭВМ с высшим приоритетом, т. е. ЭВМ1. При отказе ЭВМ1 ведущей становится ЭВМ2 и т. д. Процедура программного голосования выполняется следующим образом:

ЭВМ1 посылает в ЭВМ2 слово состояния, в ответ на которое ЭВМ2 передает в ЭВМ1 свое слово состояния и данные для сравнения;

аналогичное действие выполняют ЭВМ1 и ЭВМ3; после получения данных из обеих ЭВМ ЭВМ1 сравнивает их и выдает в качестве результата те данные, которых большинство;

ЭВМ1 посылает в ЭВМ2 и ЭВМ3 сигналы продолжения работы.

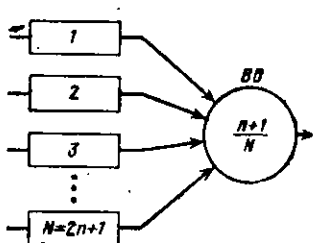


Рис. 6.16. N -уровневая мажоритарная система

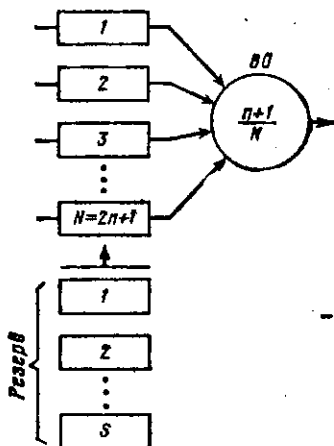


Рис. 6.17. Гибридная избыточная структура

Ошибки в ЭВМ обнаруживаются либо по установленному самой ЭВМ биту ошибки в своем слове состояния, либо по несовпадению ее выходных результатов с результатами других ЭВМ, либо по отсутствию ответа на слово состояния, посылаемое другой ЭВМ. Все ошибки принимаются сначала за случайные. ЭВМ, в которой произошла ошибка, перезагружается со стороны исправной ЭВМ и продолжает работу. При повторении ошибок они классифицируются как отказ, и отказавшая ЭВМ выводится из системы, которая продолжает работу как дублированная.

Ошибка, обнаруженная по биту ошибки в слове состояния, также вызывает перезагрузку неисправной ЭВМ со стороны одной из исправных и продолжение вычислений.

В ТМС с аппаратным голосованием для управления ТМС используется специальный блок управления БУТМС, выполняющий функции, аналогичные функциям блока управления дублированной системы.

Наиболее общим случаем мажоритарной структуры является N -уровневая мажоритарная система (NМС), приведенная на рис. 6.16. При нечетном числе блоков $N = 2n +$

+1 система сохраняет работоспособность при отказе n блоков.

Недостатком НМС является то, что отказ системы наступает при отказе $(n+1)$ -го блока несмотря на то, что еще имеется n исправных блоков. Для повышения эффективности подобных структур используются адаптивные ВО. Адаптивным называется ВО, изменяющий свой порог по мере выхода блоков из строя. Например, при использовании НМС «3 из 5» после отказа двух блоков для образования правильного выходного сигнала ВО должен использовать большинство «2 из 3». Для перестройки адаптивный ВО использует информацию о неисправных блоках.

Гибридные избыточные системы

В общем случае гибридная избыточная система (ГИС) состоит из N основных постоянно включенных по схеме НМС блоков и S резервных блоков, замещающих основные из схемы НМС при отказах последних. Гибридная избыточная система отказывает при отказе всех резервных блоков и одного из блоков ядра НМС.

Ядром НМС называется минимальное число исправных блоков $(n+1)$, необходимых для правильной работы системы. На рис. 6.17 приведена ГИС, содержащая НМС и S резервных блоков.

Подобная ГИС обозначается ГИС (N, S) .

Существенную часть ГИС составляют переключательные схемы, обеспечивающие отключение отказавших и включение резервных блоков. Надежность переключательных схем является фактором, ограничивающим общую надежность системы. Имеется предел, при котором уменьшение надежности из-за переключателя превышает увеличение надежности путем резервирования.

На рис. 6.18 показана переключательная схема ГИС $(3, 2)$ [106]. В этой схеме каждому блоку соответствует схема сравнения, которая сравнивает выходной сигнал ВО с выходным сигналом блока и при несовпадении их сбрасывает триггер состояния блока. Выходные сигналы триггеров состояния поступают на итеративную схему из пяти модулей, определяющую блоки, которые должны подключаться ко входам ВО. Состояние выходов каждого модуля V_j^i , где i —номер блока, а j —номер входа ВО, зависит от числа подключенных ко входам ВО блоков и состояния триггеров блоков. Так, $V_1^1=1$ означает, что к первому входу ВО будет подключен блок 1, а $V_3^5=1$ — что к третьему входу ВО будет подключен блок 5.

Таблица 6.1

Число исправных блоков, предшествующих i -му блоку	Состояние линий V_1^i, V_2^i, V_3^i при исправном состоянии i -го блока
0	1 0 0
1	0 1 0
2	0 0 1
3	0 0 0

Принцип работы итеративной схемы поясняется табл. 6.1. Как видно из рис. 6.18, сигналы о состоянии блоков распространяются в итеративной схеме слева направо, и если эта информация свидетельствует о том, что до i -го блока имеется всего один исправный блок, то логическая единица будет иметь место на выходе V_2^i модуля i , так как первый исправный блок уже подключен к первому входу ВО.

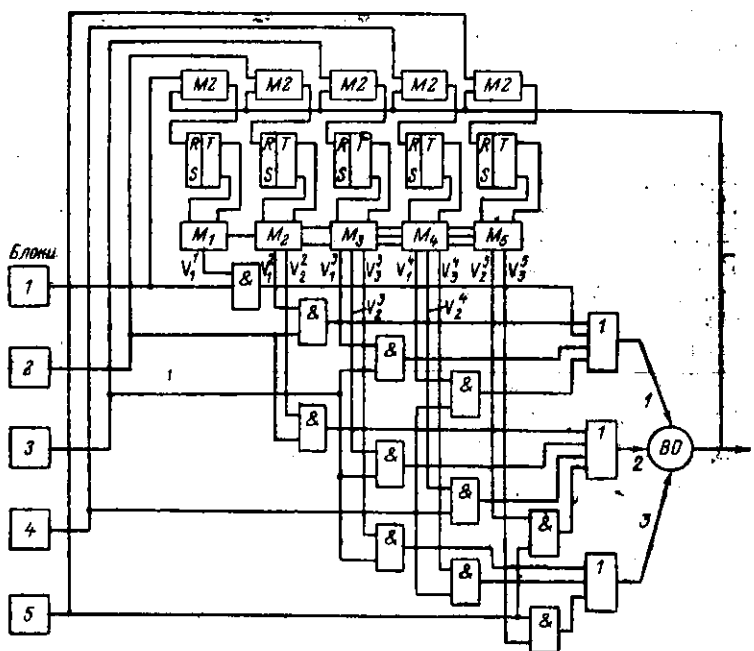


Рис. 6.18. Переключательная схема ГИС (3,2)

Рассмотрим другой случай. Пусть задано исходное состояние исправности первых трех блоков, т. е. $V_1^1=1$, $V_2^2=1$, $V_3^3=1$. Тогда при отказе в процессе работы блока 3 на его место подключается блок 4, т. е. на выходе V_3^3 установится состояние логического нуля, а на выходе V_4^4 — логической единицы. Если после этого откажет и блок 2, то произойдет переключение и ко входам 1—3 ВО будут подключены соответственно блоки 1, 4 и 5.

Недостатком приведенной итеративной переключательной схемы является то, что блоки выключаются и в том случае, когда ошибка была случайной (кратковременной). Для преодоления этого недостатка разработаны переключательные схемы, предусматривающие повторное включение блоков, когда исправным остается только ядро НМС, входящей в состав ГИС (N, S). В этих схемах имеются специальные триггеры повторения, по одному на каждый блок, фиксирующие попытки повторного включения. Кроме того, для подсчета числа повторных включений для каждого блока предусматривают счетчик попыток повторения. Блок полностью выводится из системы, когда установленное число попыток включения блока в систему оказалось безуспешным.

Самопроверяемые избыточные системы

Особый класс систем образуют *самопроверяемые избыточные системы* (СИС), состоящие из блоков, имеющих схемы встроенного контроля с вероятностью обнаружения ошибок заданного класса, близкой к 1 [56, 97].

Из самопроверяемых блоков, подобных изображенному на рис. 6.19, могут быть образованы дублированные, мажоритарные и гибридные избыточные системы. Условно будем считать, что при обнаружении схемой встроенного контроля ошибки на ее выходе имеет место логический нуль, а в противном случае — единица. Дублированную СИС будем обозначать СИС (2), мажоритарную — СИС (N), а гибридную СИС (N, S). На рис. 6.20 показана СИС (2).

При возникновении в одном из блоков ошибки, обнаруженной его схемой контроля, сигнал ошибки поступает в ВО, где блокирует выход ошибочного блока. Как видно из рис. 6.20, дополнительно к схемам контроля имеется схема сравнения выходных сигналов блоков, обеспечивающая дополнительную проверку работы блоков. Функционирование системы поясняется с помощью табл. 6.2, в которой $СК_i$ обозначает схему контроля i -го блока. Элемент K (первый

Таблица 6.2

К	СК ₁	СК ₂	Состояние системы
0	0	0	Отказ системы
0	0	1	Выходной сигнал ВО равен выходному сигналу Б ₂
0	1	0	Выходной сигнал ВО равен выходному сигналу Б ₁
0	1	1	Необходима дополнительная проверка системы
1	0	0	Отказ системы
1	0	1	Необходима дополнительная проверка системы
1	1	0	Необходима дополнительная проверка системы
1	1	1	Выходной сигнал ВО равен выходному сигналу Б ₁ или выходному сигналу Б ₂

столбец) равен 1, когда выходные сигналы обоих блоков совпадают, и 0 в противном случае.

Необходимость дополнительной проверки системы возникает, когда дополнительная схема контроля обнаружива-

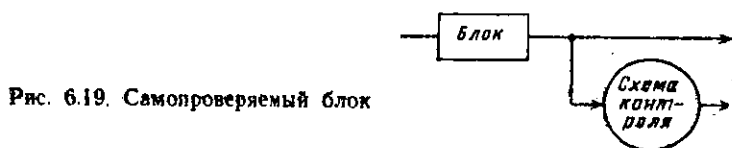


Рис. 6.19. Самопроверяемый блок

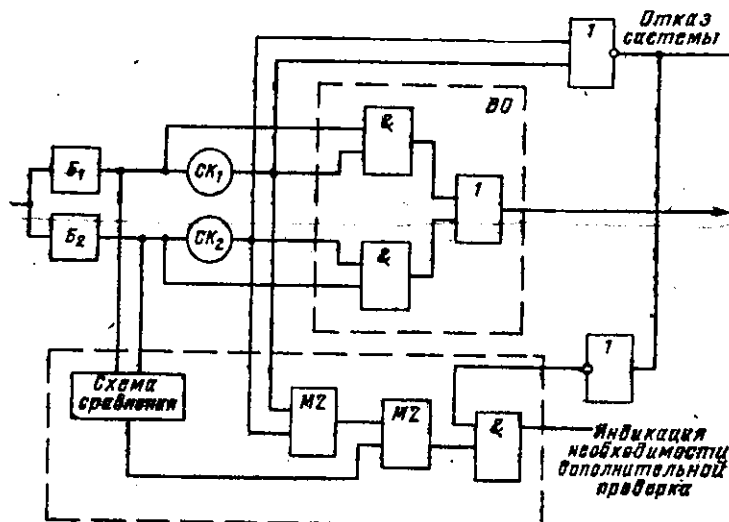


Рис. 6.20. Самопроверяемая избыточная система типа СИС (2)

Таблица 6.3

Тип избыточной системы	Вероятность безотказной работы
TMC	$p^3 + 3p(1-p)$
NMC	$\sum_{i=0}^n \binom{N}{i} (1-p)^i p^{N-i}$
ГИС (3, S)	$1 - (1-p)^{S+2} (1+p^{S+2})$
ГИС (N, S)	$\sum_{i=0}^{i=n+S} \binom{N+S}{i} (1-p)^i p^{N+S-i}$
СИС (2)	$2p - p^2$
СИС (N, S)	$1 - (1-p)^{N+S}$

$$\binom{A}{B} = \frac{A!}{(A-B)!B!}$$

ет ошибку в самих схемах контроля блоков из-за ошибки в аппаратуре схем контроля или из-за ошибочного выходного сигнала блока, не обнаруженного схемой встроенного контроля блока вследствие неполноты охвата схем блока контролем. В этом случае для локализации неисправности в блоке или его схеме контроля выполняются диагностические тесты.

В заключение приведем вероятности безотказной работы для различных необслуживаемых избыточных систем без учета надежности переключаемых схем (табл. 6.3). Вероятность безотказной работы одного блока равна p .

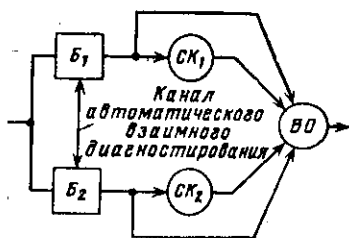
Обслуживаемые системы удобнее характеризовать средней наработкой между отказами. Для двух наиболее используемых систем СИС (2) и ТМС эта величина соответственно равна

$$T_{02} \approx T_0^2/2T_{\text{в}}; \quad T_{03} \approx T_0^2/6T_{\text{в}},$$

где T_{02} , T_{03} — средняя наработка между отказами СИС (2) и ТМС; T_0 — средняя наработка между отказами одного блока системы; $T_{\text{в}}$ — время восстановления блока.

Эти зависимости показывают влияние времени восстановления блока на среднюю наработку на отказ всей системы. Уменьшение $T_{\text{в}}$ блока позволяет существенно повысить наработку системы на отказ. Одним из резервов уско-

Рис. 6.21. Система типа СИС (2) с автоматическим взаимным диагностированием блоков



рения восстановления является сокращение времени локализации отказа путем автоматизации диагностирования.

Примером такого подхода является система локального сервиса в двухмашинном и двухпроцессорном комплексе на базе ЭВМ ЕС-1045, которые являются избыточными системами типа СИС (2) с программно реализованным восстанавливающим органом. В такой системе каждая из ЭВМ (процессоров) может выполнять диагностирование другой ЭВМ в мультипрограммном режиме параллельно с решением задач пользователя (рис. 6.21).

Глава седьмая

ЗАЩИТА, СОХРАНЕНИЕ

И ВОССТАНОВЛЕНИЕ БАЗ ДАННЫХ

7.1. ПРОБЛЕМЫ ЭКСПЛУАТАЦИИ БАЗ ДАННЫХ

Базы данных представляют собой совокупность хранимых в ЭВМ структурированных информационных массивов большой емкости, используемых в автоматизированных системах учета, планирования и управления, системах автоматизации проектирования и других системах обработки данных. Часто базы данных размещаются на десятках и сотнях пакетов магнитных дисков и бобин магнитных лент. И поскольку потеря информации подобных баз данных может иметь катастрофические последствия, обычно в системах управления базами данных (СУБД) предусматриваются средства защиты, сохранения и восстановления баз данных [44].

Сохранность («целостность») информации в базах данных может быть нарушена из-за аппаратурных неисправностей, дефектов программных средств, неверной работы операторов. Кроме того, возможны искажения информации баз данных вследствие одновременного обновления информации разными пользователями. В некоторых системах с

базами данных их сохранность имеет настолько важное значение, что основным показателем их надежности является не наработка на отказ или время восстановления, а сохранность базы данных. В подобных системах пользователи могут примириться с отказами ЭВМ, но потеря информации баз данных для них недопустима [90].

Требования к средствам обеспечения сохранности баз данных зависят от конкретных применений и режимов работы с базами данных. Пользователи баз данных должны формулировать требования к СУБД, обусловленные необходимостью обеспечения их сохранности, на основании тщательного анализа возможных ошибок и их последствий. Кроме того, должны быть изучены количественные характеристики методов обеспечения сохранности баз данных, такие как производительность системы, использующей конкретный метод защиты, время восстановления баз данных после сбоев.

Методы обеспечения сохранности баз данных зависят от режимов их работы, так как база данных может разрабатываться для режимов пакетной обработки, обработки в реальном масштабе времени или смешанном. При работе в реальном масштабе времени накладывается ограничение на время обработки запроса к базе данных. Смешанный режим обработки — это режим, когда одновременно с пакетной обработкой происходит обслуживание терминалов в реальном масштабе времени.

Обычно сохранность баз данных обеспечивается путем: защиты баз данных от ошибок при нормальной работе; восстановления баз данных при различных аварийных ситуациях [84].

Важной проблемой является обеспечение защиты баз данных от несанкционированного доступа.

7.2. ЗАЩИТА БАЗ ДАННЫХ ОТ ОШИБОК

Существуют следующие механизмы защиты баз данных от ошибок при нормальной работе:

- задание режима использования;
- управление одновременным обновлением.

Задание режима использования баз данных

Режим использования базы данных задается программой пользователя при взаимодействии с СУБД и определяет допустимые действия другой одновременно выполня-

ющейся программы по отношению к базе данных. Существуют следующие режимы использования базы данных:

1) монопольный. В этом режиме доступ к базе данных со стороны других программ запрещен. Обычно этот режим используется программами, вносящими глобальные изменения в базу данных и поэтому запрещающими все другие обращения к базе данных;

2) защищенный. Если одна из программ задает защищенный режим, то другие программы могут только читать информацию из базы данных, но не изменять ее.

Многие системные обслуживающие программы, выполняющие реорганизацию или восстановление, по своей природе требуют монопольного или защищенного режима использования базы данных;

3) разделенный. В этом режиме допускается любая операция с базой данных со стороны других программ. Однако если программа *A* намеревается внести изменение в базу данных в разделенном режиме, а программа *B* уже начала работу с базой данных в защищенном режиме, то программа *A* ожидает завершения программы *B*.

Модуль информации, относительно которого задается режим использования, может быть различным. Например, режим использования может задаваться для одной записи, файла или всей базы данных. Очевидно, что чем меньше модуль информации, относительно которого задается режим использования, тем лучше, так как в этом случае допускается большая степень параллелизма в работе программ.

Управление одновременным обновлением базы данных

Управление одновременным обновлением предотвращает ошибки, связанные с одновременным обновлением одних и тех же данных различными программами в разделенном

Таблица 7.1

Собы- тие	Про- грам- ма	Действие	Результат
1	A	Чтение записи <i>N</i>	Запись содержит параметр со значением 2
2	B	То же	То же
3	A	Прибавление 3 к параметру и запись модифицированного параметра в базу данных	Запись содержит параметр со значением 5
4	B	Прибавление 4 к параметру и запись модифицированного параметра в базу данных	Запись содержит параметр со значением 6

режиме использования базы данных. Ниже приведен пример ситуации, приводящей к искажению информации базы данных (табл. 7.1).

Одновременное обновление может привести к ошибке. Как видно из табл. 7.1, программа *В* обновляет значение параметра, считанного во время события 2. Это приводит к тому, что обновление этого же параметра со стороны программы *А* во время события 3 теряется. В результате этого может произойти неправильное изменение базы данных.

Очевидно, что ожидаемое после события 4 значение параметра равно 9, а не 6. Для предотвращения подобных ошибок в СУБД имеется механизм блокировки, который в приведенном примере запрещает программе *В* считывать запись *N* во время события 2.

Для включения механизма блокировки в некоторых СУБД при чтении данных требуется явное указание того, что запись читается с намерением обновления. В других системах подразумевается, что любое чтение может привести к обновлению информации. В обоих случаях выполняется блокировка прочитанной записи, которая будет снята после одного из следующих событий:

- завершения обновления;
- выдачи программой команды разблокировки;
- нормального или аварийного завершения программы;
- создания контрольной точки.

Следует отметить разницу между блокировкой и установкой режима использования. Если монопольный или защищенный режим использования базы данных блокирует обращение к базе данных (файлу, записи) на время выполнения программы, то блокировка запрещает обращение к записи только на время обновления.

7.3. ВОССТАНОВЛЕНИЕ БАЗ ДАННЫХ ПРИ АВАРИЙНЫХ СИТУАЦИЯХ

При эксплуатации баз данных возможны следующие аварийные ситуации: повреждение носителя, на котором размещена база данных; аварийное завершение программы; внезапное прекращение работы из-за отказа системы (например, системы электропитания).

Методы восстановления базы данных в аварийных ситуациях, используемые СУБД, зависят от средств обеспечения сохранности, имеющихся в системе.

Одним из наиболее простых методов восстановления баз данных является восстановление при наличии дублиро-

ванной базы данных (рис. 7.1). Недостатком дублирования базы данных является высокая стоимость и нарушение сохранности при таких ситуациях, как отказ электропитания, ошибки в памяти и процессоре, в пользовательских программах, которые могут испортить обе копии базы данных. Поэтому дублирование базы данных имеет смысл только

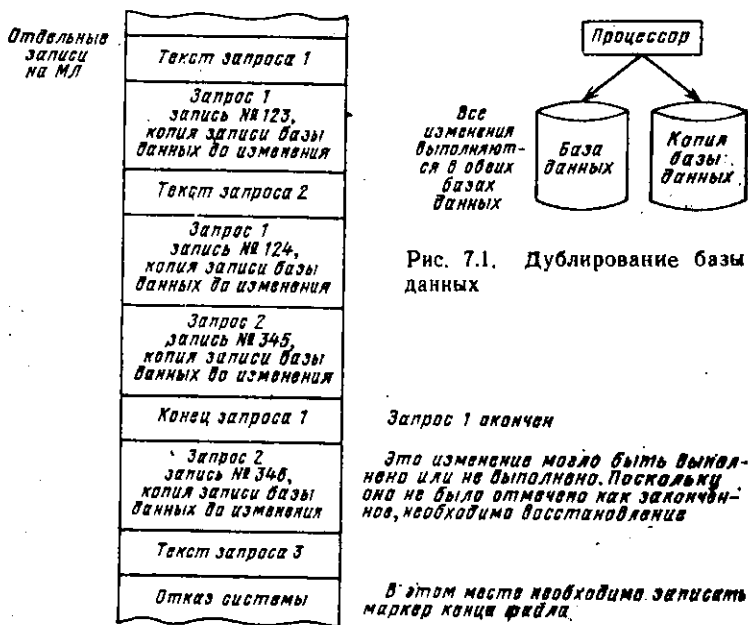


Рис. 7.2 Журнал изменений базы данных

при дублировании процессоров и гарантированном электропитании.

Другим методом обеспечения сохранности баз данных является создание поколений базы данных на магнитных лентах.

Этот метод, однако, имеет следующие недостатки:

процедура копирования базы данных с дисков на магнитную ленту может потребовать нескольких часов машинного времени ежедневно;

катастрофические отказы в течение дня приводят к потере операций текущего дня. И если некоторые пользователи могут примириться с подобной ситуацией, то для

большинства систем, работающих в реальном масштабе времени, это недопустимо;

постоянные копирования базы данных на магнитную ленту могут привести к накоплению некоторых необнаруживаемых ошибок на магнитной ленте и постепенному искажению базы данных. Чтобы исключить аварийные последствия подобных ошибок, приходится хранить по несколько недельных, месячных и годовых копий баз данных. Иногда для исключения подобных ошибок в программы копирования баз данных на магнитную ленту включаются диагностические процедуры, которые позволяют перед записью на ленту удостовериться в корректности структуры файлов, справочников, списков, а также проверить некоторую информацию на допустимость.

Наилучшие результаты дает сочетание следующих средств обеспечения сохранности баз данных: *запасных копий баз данных, журнала изменений базы данных, контрольных точек.*

Для создания через регулярные интервалы времени запасных копий базы данных почти все операционные системы содержат обслуживающие программы для сброса содержимого базы данных на магнитную ленту.

Журнал изменений базы данных представляет собой хронологическую запись информации об изменениях базы данных. В журнале изменений записывается одно из сочетаний следующих типов информации:

- теста запроса на изменение в базе данных (журнал заявок);

- меток, характеризующих начало и конец запроса;

- копии записи базы данных до внесения в нее изменения («до-журнал»);

- копии записи базы данных после внесения в нее изменения («после-журнал») [48].

Большинство СУБД записывает в журнал изменений копии записей как до, так и после изменений. На рис. 7.2 приведен пример журнала изменений, в который записываются тексты запросов, метки начала и конца запросов, а также копии записи до ее обновления.

Журналы изменений в основном ведутся на магнитных лентах, но иногда и на магнитных дисках. В системах СУБД/ПД один и тот же журнал изменений используется для протоколирования как операций с базой данных, так и сообщений телеобработки. Кроме уменьшения требуемого числа лентопротяжных устройств это имеет также дополни-

тельное преимущество при восстановлении баз данных в системах реального масштаба времени.

Для эффективного использования магнитной ленты в некоторых СУБД используется блокирование записей в журнале изменений. Обычно в СУБД заявка не выполняется до тех пор, пока копия записи базы данных, в которую вносится изменение, не будет записана в журнале изменений, а заявки ожидают своей очереди на запись в «до-журнал», что увеличивает время обработки заявок. Во избежание этого используется запоминание изменяемых записей базы данных в специальной буферной памяти и запись их в «до-журнал» только после формирования одного блока. Однако при небольшом потоке заявок и, следовательно, медленном заполнении блока может потребоваться много времени, прежде чем заявка будет обработана. Поэтому в некоторых системах размер блока или время накопления записей до их переписи в «до-журнал» может изменяться динамически.

Недостатком блокирования записей, однако, является возможность потери нескольких записей, сгруппированных в блок, что требует восстановления с более ранней точки отсчета времени.

Иногда для уменьшения времени восстановления базы данных в журнал изменений записывается файл базы данных до и после изменения. Хотя это приводит к увеличению объема журнала изменений, время восстановления базы данных при этом наименьшее.

В системах со страничной организацией памяти в качестве единицы информации, записываемой в журнал изменений, используют также страницу. Однако наиболее часто в журнал изменений копируется запись, так как она является наименьшей единицей информации, до которой распространяется логическая адресация в файле [48].

Операционные системы предоставляют возможность введения в пользовательские программы контрольных точек, которые указывают место повторного запуска (рестарта) программы при появлении ошибки или аварийном завершении программы. При рестарте с контрольной точки необходимо также восстановление базы данных в состояние, существовавшее в момент времени, соответствующий контрольной точке.

Контрольные точки в программах, работающих с базами данных, вызывают выполнение следующих действий:

внесение в базу данных всех изменений, ожидающих своей очереди;

разблокировку всех записей, обращение к которым было заблокировано;

запись информации о контрольной точке в журнал изменений.

Восстановление при повреждении носителя базы данных

Восстановление базы данных выполняется с помощью последней запасной копии базы данных. В эту копию базы данных с журнала изменений вносятся копии всех измене-

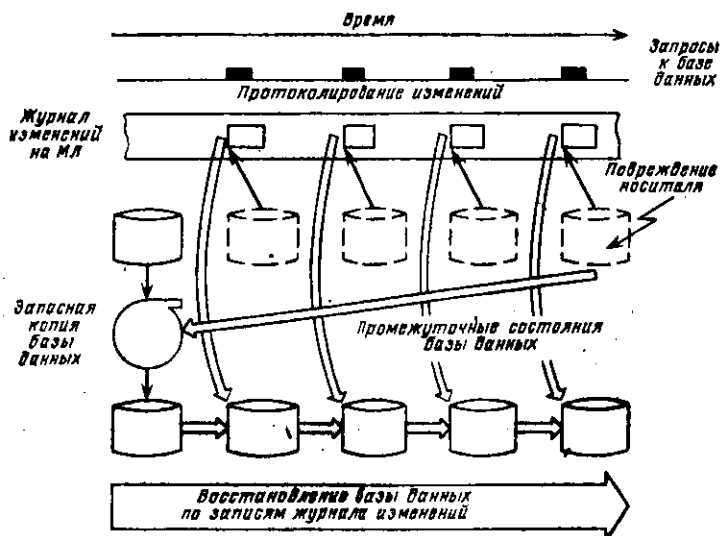


Рис. 7.3. Восстановление конечного состояния базы данных по записям журнала изменений

ний записей, зарегистрированных после получения последней копии базы данных. Эта процедура, показанная на рис. 7.3, обеспечивает восстановление базы данных в состоянии, максимально близком к временной точке отказа, и выполняется специальной обслуживающей программой, имеющейся в операционной системе. В результате этого база данных восстанавливается в состоянии, соответствующем моменту записи последней информации в журнале изменений.

Время, затрачиваемое на копирование и восстановление баз данных, является непроизводительным. Если копиро-

вание выполнять часто, то много времени уходит на создание копий. В то же время чем чаще снимаются копии базы данных, тем быстрее производится восстановление базы данных. В связи с этим возникает задача определения оптимального периода копирования базы данных [47]. Одним из резервов ускорения восстановления является внесение только последнего изменения в одну и ту же запись базы данных. Для этого в обслуживающие программы для восстановления базы данных по журналу изменений дополнительно вводится анализ номера записи, к которой относятся изменения.

Если повреждение носителя базы данных произошло во время работы (во время выполнения пакета заданий или операций реального масштаба времени), кроме восстановления базы данных необходимо обеспечить также и восстановление вычислительного процесса. При этом может оказаться удобнее восстановить базу данных только до контрольной точки и запустить прерванные программы с этой точки.

Восстановление при аварийном завершении программы

Восстановление при аварийном завершении программы предполагает восстановление как вычислительного процесса, так и базы данных.

Существуют следующие варианты восстановления:

1. База данных перезагружается и восстанавливается в состоянии, существовавшем к началу аварийно завершенной программы, после чего программа выполняется повторно.

2. База данных восстанавливается в состоянии, существовавшем в момент времени, наиболее близкий к последней контрольной точке (если были предусмотрены контрольные точки), и выполняется рестарт программы с этой контрольной точки.

Естественно, что оба варианта восстановления выполнимы при случайных сбоях.

3. Изменения в базе данных со стороны аварийно завершенной программы могут быть аннулированы с помощью операций восстановления исходного состояния. Для этого все копии записей из «до-журнала» записываются на место измененных данных. Этот процесс продолжается до восстановления всех записей, измененных программой, или только записей, измененных после последней контрольной точки.

Большинство СУБД имеет в своем составе специальные обслуживающие программы восстановления базы данных.

в исходное состояние, существовавшее до выполнения программы в момент последней контрольной точки.

Методы восстановления при аварийном завершении программы зависят от режима эксплуатации базы данных: пакетная обработка; обработка в реальном масштабе времени, смешанная обработка (обработка в реальном масштабе времени одновременно с пакетной обработкой).

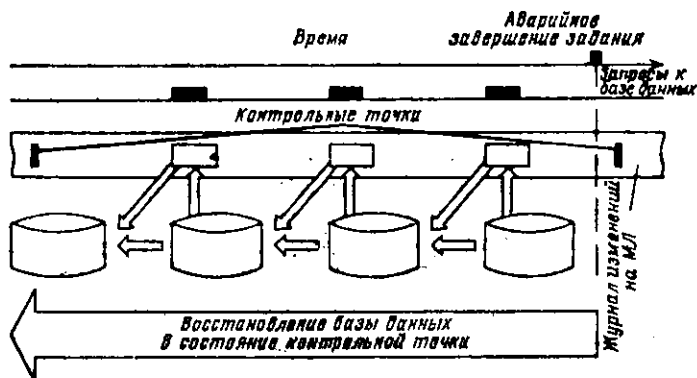


Рис. 7.4. Восстановление базы данных в состояние контрольной точки

Пакетная обработка. При пакетной обработке запросы к базе данных обрабатываются последовательно. Восстановление при аварийном завершении программ выполняется с использованием контрольных точек (рис. 7.4). Для повторения программы с контрольной точки необходимо, чтобы база данных по состоянию соответствовала данным контрольной точки программ пользователя. Для этого СУБД должна возвращать базу данных в исходное состояние путем аннулирования части изменений.

При коллективном использовании базы данных, т. е. при одновременной работе с базой данных нескольких программ, могут возникать ситуации, для преодоления которых необходимо предусматривать специальные меры. Рассмотрим, например, ситуацию, когда обновление базы данных в режиме разделенного использования выполняют две разные программы, одна из которых завершается аварийно. Последовательность событий, изображенных на рис. 7.5, такова: программа А сначала обновляет записи 1, 2 и 3, а спустя некоторое время программа В обновляет записи 4, 2 и 5.

Если в результате аварийного завершения программы А

выполняется восстановление базы данных в исходное состояние, существовавшее до обновления записи 1, то это приведет к потере обновления записи 2 со стороны программы В. Как видно из этого примера, восстановление исходного состояния базы данных для аварийно завершенной программы влияет на другую программу, работающую с той же базой данных, причем такое нарушение может произойти, даже если программа В использует базу данных

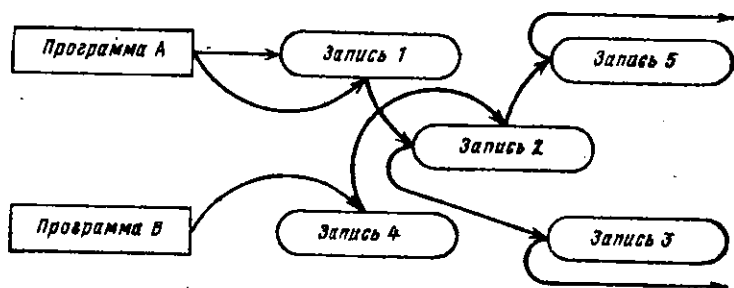


Рис. 7.5. Ситуация, при которой возможно неправильное восстановление базы данных

только для чтения информации. Так, если программа А обновила запись 2, а в программе В выполняется ветвление на основании этой обновленной информации, то восстановление базы данных в исходное для программы А состояние приведет к изменению условия ветвления в программе В и последняя будет работать неверно.

Для исключения подобных ошибок СУБД выполняет восстановление базы данных в исходное состояние для всех программ, работающих с базой данных. Это выполняется по записям в журнале изменений. Для правильного автоматического восстановления базы данных и повторного выполнения программ имеются две возможности:

1) система управления базой данных запоминает список всех измененных между контрольными точками записей. Обращение программы В к записям из списка запрещается до тех пор, пока программа А не закончится или не сформирует контрольную точку. В этом случае при аварийном завершении программа А может быть повторена автоматически без учета изменений со стороны других программ. Недостатком этого способа являются дополнительные затраты времени на запоминание списка измененных

записей и на ожидание другими программами освобождения нужных им записей базы данных;

2) синхронизация контрольных точек разных программ, с тем чтобы возврат мог осуществляться к одной и той же временной точке. Для этого требуется, чтобы все программы с обновлением записей создавали контрольные точки одновременно.

Общие контрольные точки создает СУБД. При этом система ожидает окончания активных запросов, после чего информирует программы пользователей о контрольной точке, а также записывает информацию о ней в журнале изменений.

К издержкам синхронизации контрольных точек относится некоторое снижение производительности системы, так как система ожидает окончания активных запросов, а при рестарте исключает все запросы, которые не затронуты ошибкой.

Обработка в реальном масштабе времени. При работе в реальном масштабе времени каждый запрос обычно обрабатывается путем полного прогона программы (модуля), предназначенного для его обработки. И поскольку в этом случае контрольные точки всегда совпадают с началом программы, возврат на повторение программы соответствует возврату на повторение одного запроса. Это облегчает восстановление.

Программа управления запросами может повторить подачу запроса на вход соответствующей программы обработки, которая и выполнит повторную обработку запроса.

Смешанная обработка. При смешанной обработке должна быть установлена общая контрольная точка для всей системы. Различные контрольные точки для программ пакетной обработки и программ реального масштаба времени допустимы только в том случае, когда оба режима используют различные группы базы данных. В интегрированных системах управления базой данных и телеобработки синхронизацию контрольных точек программ реального масштаба времени и контрольных точек планировщика базы данных для синхронизации рестарта обеспечивает программа управления запросами. Кроме того, программа управления запросами может вызвать контрольную точку для всей системы, приостанавливая обработку запросов до окончания всех операций в системе.

В системах с небольшим входным потоком запросов может оказаться возможным, чтобы программа обработки запросов работала в защищенном режиме использования ба-

зы данных. Это решает проблему восстановления, так как каждый запрос использует базу данных в защищенном режиме и может повторяться независимо.

Восстановление при отказе системы

Основной особенностью отказа системы является потеря информации о состоянии операционной системы и оперативной памяти ЭВМ. Поэтому информация о том, что происходило в системе в момент отказа, может быть получена только из журнала изменений базы данных и от оператора. Исходя из этого, отказ системы можно рассматривать как аварийное завершение всех выполнявшихся программ плюс потенциальная потеря информации из-за ошибок в аппаратуре.

Успешное восстановление базы данных при отказе системы зависит от качества анализа ситуации и решения, принятого специалистом. В распоряжении последнего находятся все перечисленные выше средства обеспечения сохранности и восстановления базы данных.

Существуют два варианта восстановления:

1) восстановление состояния, наиболее близкого к временной точке отказа, путем использования запасной копии базы данных и журнала изменений (см. рис. 7.3);

2) восстановление исходного состояния базы данных с помощью журнала изменений путем аннулирования части изменений.

В системах реального масштаба времени восстановление базы данных с запасной копии фактически приводит к аварийной ситуации, следствием которой является образование очереди запросов на входе системы [48].

Поэтому первый вариант восстановления применяется в основном при пакетной обработке, а второй — при работе в реальном масштабе времени. В последнем случае восстановление исходного состояния базы данных выполняется до момента контрольной точки и все программы возвращаются к общей контрольной точке.

Восстановительные процедуры обычно выполняются с учетом момента отказа системы. Отказ системы мог произойти во время записи информации в журнал изменений или в базу данных. Поэтому одним из первых шагов процедуры восстановления является прогон специальной обслуживающей программы, которая, анализируя состояние журнала изменений, определяет программу, выполнявшуюся в момент отказа, и записывает маркер конца файла на магнитной ленте журнала изменений.

Если отказ системы произошел в момент выполнения записи в журнал изменений, то информация теряется. Потеря этой информации, однако, не вызывает затруднений при восстановлении, так как запись в журнал изменений выполняется до записи в базу данных, т. е. это изменение в действительности не выполняется.

Если отказ системы произошел после записи информации в журнал изменений, но до записи ее в базу данных, проблема восстановления также разрешима. При восстановлении исходного состояния по журналу изменений последняя запись базы данных восстанавливается в состояние, существовавшее до последнего изменения, т. е. в то же самое состояние, так как изменение последней записи в базе данных не было выполнено.

Восстановление конечного состояния по запасной копии также выполняется нормально, так как оно обычно выполняется до предшествующей отказу контрольной точки, т. е. последнее изменение в журнале изменений не используется.

Опасность потери информации может возникнуть, когда в журнал изменений заносятся заблокированные записи и отказ системы произошел при накоплении записей в оперативной памяти. При этом восстановление исходного состояния базы данных и возврат программы к контрольной точке могут привести к неверным результатам, если только запись в журнал изменений не выполнять принудительно при появлении контрольной точки или по окончании программы, даже если блок информации не сформирован полностью. В противном случае обновление базы данных не сопровождается протоколированием этого события в журнале изменений и рестарт, или повторный прогон программы, приведет к двухкратной обработке одного и того же запроса к базе данных.

С учетом всех изложенных выше ограничений восстановление при отказе выполняется следующим образом: база данных восстанавливается в определенное состояние, с которого может начинаться обработка или рестарт программ, загружается СУБД и снова запускается программа обработки запросов пользователей.

В зависимости от системы процесс восстановления может быть либо совмещен с другими операциями, либо нет. Отметим следующие особенности процедур восстановления баз данных:

1. Все восстановительные процедуры, выполняемые с помощью специальных обслуживающих программ, завершаются до возобновления нормальной работы СУБД. Для га-

рантированного восстановления до начала работы в некоторых системах предусмотрена установка флажка отказа. Флажок устанавливается в самой базе данных, когда база данных или ее файл открывается для обновления и сбрасывается только после нормального завершения программы. При отказе системы и повторной ее инициализации анализируется состояние флажка, и если он установлен, дальнейшее обращение к базе данных запрещается до окончания восстановления.

2. При повторной инициализации системы после отказа все процессы и соответствующие им записи базы данных приводятся в исходное состояние.

3. Восстановление части базы данных выполняется одновременно с нормальной работой программ, использующих другие части базы данных.

7.4. МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

Широкое развитие работ по созданию на основе ЭВМ информационных систем и АСУ, предназначенных для накопления, хранения в информационных базах, обработки и выдачи по запросам различного рода данных, сделало весьма актуальной проблему защиты информации от несанкционированного доступа. Острота этой проблемы и сложность ее решения усугубляются тем, что в современной информационной системе или АСУ в базе данных (или базах данных) хранится большое количество информации, предназначенной для различных организаций-пользователей, их подразделений и отдельных лиц. При этом доступ к информации производится в режиме коллективного пользователя, в том числе с использованием передачи данных по линиям связи. В последние годы проблеме защиты данных в зарубежной литературе уделяется большое внимание, о чем свидетельствует, например, список литературы в [71].

Согласно современным представлениям под *защитой информации от несанкционированного доступа*, или *безопасностью данных*, понимают предотвращение случайного или преднамеренного разрушения, раскрытия (утечки) или искажения данных.

В работах по методам защиты информации *секретностью* называют право пользователя объявлять информацию закрытой, т. е. подлежащей защите. Статус, представленный данным и определяющий требуемую степень защиты, называется *конфиденциальностью данных* [71].

Системой защиты информации называют совокупность методов, аппаратурных и программных средств, организационных и других мер, предназначенных для предотвращения разрушения, искажения или неразрешенного использования данных.

При реализации защиты информации возникает необходимость в *идентификации* как пользователей, обращающихся к данным, так и устройств, участвующих в процессе взаимодействия с информационной системой — терминалов, машин и их устройств, каналов связи, файлов данных, программ и т. п. Идентификация предполагает присвоение пользователям, терминалам и другим техническим средствам, программам и файлам *идентификаторов*, т. е. уникальных имен или кодов.

Доступу пользователя к запрашиваемой информации должны предшествовать процедуры установления подлинности, проверки полномочий и регистрации факта запроса и выдачи (или отказа в выдаче) информации.

Установление подлинности состоит в проверке того, что лицо или устройство, сообщившее идентификатор, в действительности ему соответствует. Строго говоря, требуется установление подлинности не только пользователя и использованного им терминала по отношению к системе, но и системы (ЭВМ, программы, файла) по отношению к запрашиваемым пользователем ресурсам. С одной стороны, могут существовать ограничения, разрешающие выдачу некоторой информации лишь на определенные терминалы. С другой стороны, пользователь, связываясь, например, в вычислительной сети с определенной машиной, программой, файлом, должен быть уверен, что он действительно оказался связанным с тем средством, которое он запросил. В противном случае при ошибочном или злонамеренном установлении связи пользователя с незапрошенными им средствами (ЭВМ, файлами и т. п.) возможна утечка информации и нарушение соответствующих данных в выделенных для них файлах (*нарушение целостности данных*).

Одним из способов установления подлинности пользователя и средств системы является использование паролей, которыми они должны обмениваться.

После установления подлинности вступивших в связь пользователя и средств системы необходима *проверка их полномочий* по отношению к используемому в запросе элементу данных (файлу, записи, полю и др.), т. е. проверка права пользователя (а также терминала) на доступ к за-

прашиваемым данным, выполнение над ними определенных операций (например, чтения или обновления).

Важным элементом системы защиты информации является осуществляемая ЭВМ автоматическая регистрация в специальном журнале всех удовлетворенных и отвергнутых запросов к данным с указанием пользователя, терминала, времени поступления запроса, запрошенной информации и некоторых других сведений.

Криптографическое закрытие или, другими словами, шифрование информации рассматривается специалистами как наиболее надежный метод защиты информации, особенно при передаче ее по линиям связи.

Криптографическое закрытие состоит в преобразовании информации с помощью шифров и процедур шифрования, в результате чего по внешнему виду данных оказывается невозможным определить их содержание. Шифрование информации делает ее для лица, не владеющего шифром, нераспознаваемой (с определенной вероятностью).

Шифрование информации производится перед вводом ее в информационную систему или перед передачей по линии связи. Дешифрование информации выполняется при санкционированной выдаче информации из системы, а при передаче данных по линиям связи — на приемном пункте для передачи исходного текста по месту назначения.

Упомянутые выше методы и процедуры защиты информации, а также некоторые другие защитные меры реализуются с помощью специальных аппаратурных и (или) программных средств и их комбинаций.

К аппаратурным средствам защиты информации относятся в частности, замки-выключатели, логически отключающие от ЭВМ различные пульты, дисплеи, клавиатуры, другие устройства ввода-вывода, препятствующие тем самым попыткам их несанкционированного использования для ввода-вывода, искажения информации в ЭВМ. Использование указанных устройств возможно только при открытии замков специальными ключами.

Применяются также замки, блокирующие двери шкафов ЭВМ и включающие сигнализацию при попытке несанкционированного (без соответствующего ключа) доступа к аппаратуре.

К аппаратурным средствам защиты относятся специальные регистры, хранящие для соответствующих файлов или блоков данных уровни секретности, пароли, идентификаторы пользователей и терминалов, определяющих, каким лицам и с каких терминалов разрешен доступ к информации,

а также схемы, автоматически формирующие (при попытке с терминала установить связь с ЭВМ) присвоенный терминалу код.

Шифрование и дешифрование информации может производиться как специальными программами, так и встраиваемой в систему специальной аппаратурой [71].

На специальные программы возлагают установление подлинности пользователей и средств системы (терминалов, устройств ввода-вывода, машин, файлов, программ), проверку их полномочий, введение журнала обращений к системе, сигнализацию при попытках несанкционированного доступа.

Глава восьмая

СИСТЕМЫ ДИАГНОСТИРОВАНИЯ ЭВМ

8.1. ХАРАКТЕРИСТИКИ СИСТЕМ ДИАГНОСТИРОВАНИЯ

Быстро увеличивается число ЭВМ, находящихся в эксплуатации, и возрастает их сложность. В результате растет численность обслуживающего персонала и повышаются требования к его квалификации. Увеличение надежности машин приводит к тому, что поиск неисправных элементов и ремонт их производятся сравнительно редко. Поэтому наряду с повышением надежности машин наблюдается тенденция потери эксплуатационным персоналом определенных навыков отыскания и устранения неисправностей. Таким образом, возникает проблема обслуживания непрерывно усложняющихся вычислительных машин и систем в условиях, когда не хватает персонала высокой квалификации.

Современная вычислительная техника решает эту проблему путем создания систем автоматического диагностирования неисправностей, которые призваны облегчить обслуживание и ускорить ремонт машин.

Система автоматического диагностирования представляет собой комплекс программных, микропрограммных и аппаратурных средств и справочной документации (диагностических справочников, инструкций, тестов).

Введем некоторые определения, которые потребуются в дальнейшем при описании различных систем автоматического диагностирования [53].

Различают системы *тестового* и *функционального* диагностирования. В системах тестового диагностирования воз-

действия на диагностируемое устройство (ДУ) поступают от средств диагностирования (СД). В системах функционального диагностирования воздействия, поступающие на ДУ, заданы рабочим алгоритмом функционирования. Обобщенные схемы систем тестового и функционального диагностирования показаны на рис. 8.1.

Классификация средств диагностирования приведена на рис. 8.2.

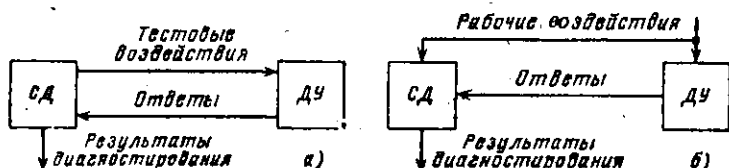


Рис. 8.1. Обобщенные схемы систем тестового (а) и функционального (б) диагностирования

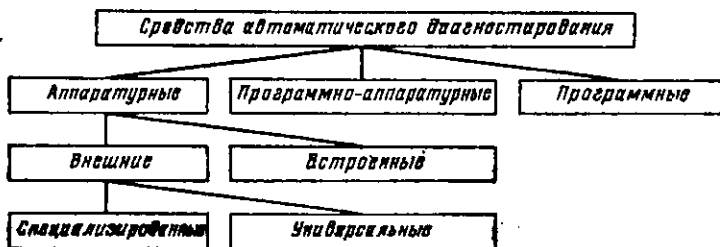


Рис. 8.2. Классификация средств автоматического диагностирования

В средних и больших ЭВМ используются, как правило, встроенные (специализированные) средства диагностирования. В микро-ЭВМ чаще используются встроенные средства подачи тестовых воздействий и внешние универсальные средства (например, сигнатурные анализаторы) для снятия ответов и анализа результатов.

Процесс диагностирования состоит из определенных частей (*элементарных проверок*), каждая из которых характеризуется подаваемым на устройство тестовым или рабочим воздействием и снимаемым с устройства ответом. Получаемое значение ответа (значения сигналов в контрольных точках) называется *результатом элементарной проверки*.

Объектом элементарной проверки назовем ту часть аппаратуры диагностируемого устройства на проверку которой рассчитано тестовое или рабочее воздействие элементарной проверки.

Совокупность элементарных проверок, их последовательность и правила обработки результатов определяют алгоритм диагностирования.

Алгоритм диагностирования называется безусловным, если он задает одну фиксированную последовательность реализации элементарных проверок [53].

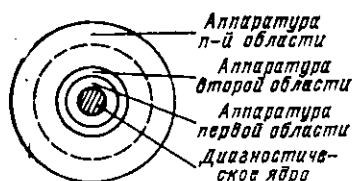


Рис. 8.3. Процесс диагностирования по принципу раскрутки

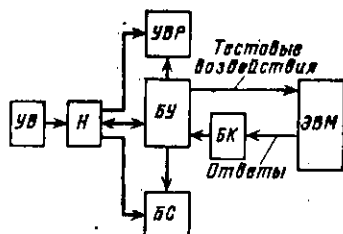


Рис. 8.4. Структурная схема встроенных средств тестового диагностирования

Алгоритм диагностирования называется *условным*, если он задает несколько различных последовательностей реализации элементарных проверок.

Средства диагностирования позволяют ЭВМ самостоятельно локализовать неисправность при условии исправности *диагностического ядра*, т. е. той части аппаратуры, которая должна быть заведомо работоспособной до начала процесса диагностирования.

При диагностировании ЭВМ наиболее широкое распространение получил *принцип раскрутки*, или *принцип расширяющихся областей*, заключающийся в том, что на каждом этапе диагностирования ядро и аппаратура уже проверенных исправных областей устройства представляют собой средства тестового диагностирования, а аппаратура очередной проверяемой области является объектом диагностирования.

Процесс диагностирования по принципу раскрутки, или расширяющихся областей, показан на рис. 8.3 [53]. Диагностическое ядро проверяет аппаратуру первой области, затем проверяется аппаратура второй области с использованием ядра и уже проверенной первой области и т. д.

Диагностическое ядро, или встроенные средства тестового диагностирования (СТД), выполняет следующие функции:

- загрузку диагностической информации;
- подачу тестовых воздействий на вход проверяемого блока;
- опрос ответов с выхода проверяемого блока;
- сравнение полученных ответов с ожидаемыми (эталонными);
- анализ и индикацию результатов.

Для выполнения этих функций встроенные СТД в общем случае содержат устройства ввода (УВ) и накопители (Н) диагностической информации (тестовые воздействия, ожидаемые ответы, закодированные алгоритмы диагностирования), блок управления (БУ) чтением и выдачей тестовых воздействий, снятием ответа, анализом и выдачей результатов диагностирования, блок коммутации (БК), позволяющий соединить выходы диагностируемого блока с блоком сравнения, блок сравнения (БС) и устройство вывода результатов диагностирования (УВР). На рис. 8.4 приведена структурная схема встроенных средств тестового диагностирования.

Показанные на структурной схеме блоки и устройства могут быть частично или полностью совмещенными с аппаратурой ЭВМ. Например, в качестве устройств ввода могут использоваться внешние запоминающие устройства ЭВМ, в качестве накопителя — часть оперативной или управляющей памяти, в качестве блока управления — микропрограммное устройство управления ЭВМ, в качестве блока сравнения — имеющиеся в ЭВМ схемы сравнения, в качестве блока коммутации — средства индикации состояния аппаратуры ЭВМ, в качестве устройства вывода результатов — средства индикации пульта управления или пишущая машинка.

Как видно из структурной схемы, приведенной на рис. 8.4, встроенные средства диагностирования имеют практически те же блоки и устройства, что и универсальные ЭВМ. И не удивительно, что с развитием интегральной микроэлектроники и массовым выпуском недорогих микропроцессоров и микро-ЭВМ их стали использовать в качестве средств диагностирования ЭВМ. Такие специализированные процессоры, используемые в целях обслуживания и диагностирования ЭВМ, получили название *сервисных процессоров* (рис. 8.5). Благодаря своим универсальным возможностям и развитой периферии, включающей пультный накопитель, кла-

виатуру, пишущую машинку и дисплей, сервисные процессоры обеспечивают комфортные условия работы и представление результатов диагностирования обслуживающему персоналу в максимально удобной форме.

Для классификации технических решений, используемых при реализации систем диагностирования, введем понятие метода диагностирования.

Метод диагностирования характеризуется объектом элементарной проверки, способом подачи воздействия и снятия ответа.

Существуют следующие методы тестового диагностирования:

- двухэтапное диагностирование;
- последовательное сканирование;
- эталонные состояния;
- микродиагностирование;
- диагностирование, ориентированное на проверку сменных блоков.

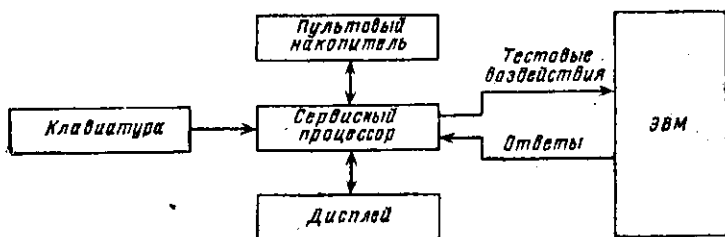


Рис. 8.5. Структурная схема средств тестового диагностирования на базе сервисного процессора

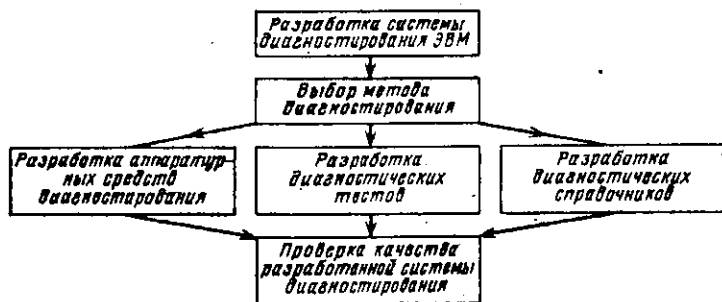


Рис. 8.6. Этапы проектирования систем тестового диагностирования

Методы функционального диагностирования включают в себя:

диагностирование с помощью схем встроенного контроля;

диагностирование с помощью самопроверяемого дублирования;

диагностирование по регистрации состояния.

Процесс разработки систем диагностирования состоит из следующих этапов (рис. 8.6):

выбора метода диагностирования;

разработки аппаратурных средств диагностирования;

разработки диагностических тестов;

разработки диагностических справочников;

проверки качества разработанной системы диагностирования.

Для сравнения различных систем диагностирования и оценки их качества чаще всего используются следующие показатели:

вероятность обнаружения неисправности (F);

вероятность правильного диагностирования (D). Неисправность диагностирована правильно, если неисправный блок указан в разделе диагностического справочника, соответствующем коду останова. В противном случае неисправность считается обнаруженной, но нелокализованной. Для ЭВМ с развитой системой диагностирования обычно $F \geq 0,95$, $D \geq 0,90$. В том случае, когда неисправность только обнаружена, необходимы дополнительные процедуры по ее локализации. Однако благодаря тем возможностям, которые система диагностирования предоставляет обслуживающему персоналу (возможность зацикливания тестового примера для осциллографирования, эталонные значения сигналов в схемах на каждом примере, возможность останова на требуемом такте), локализация неисправности после ее обнаружения не требует больших затрат времени;

средняя продолжительность однократного диагностирования ($\bar{\tau}_д$). Величина $\bar{\tau}_д$ включает в себя продолжительность выполнения вспомогательных операций диагностирования и продолжительность собственно диагностирования. Часто удобнее использовать коэффициент продолжительности диагностирования

$$k_d = 1 - \bar{\tau}_д / \bar{T}_в, \quad (8.1)$$

где $\bar{T}_в$ — время восстановления. Коэффициент k_d показывает, какая часть времени восстановления остается на вос-

становительные процедуры. Так, например, если $\bar{T}_d = 15$ мин, а $\bar{T}_B = 60$ мин, $k_d = 1 - 15/60 = 0,75$;

глубина поиска дефекта (L). Величина L указывает составную часть диагностируемого устройства с точностью, до которой определяется место дефекта.

В ЭВМ за глубину поиска дефекта L принимается число предполагаемых неисправными сменных блоков (ТЭЗ), определяемое по формуле

$$L = \left(\sum_{i=1}^N n_i \right) / N, \quad (8.2)$$

где n_i — число предполагаемых неисправными сменных блоков (ТЭЗ) при i -й неисправности; N — общее число неисправностей.

В качестве показателя глубины поиска дефекта можно также использовать коэффициент глубины поиска дефекта $k_{г.п.д.}$, определяющий долю неисправностей, локализуемых с точностью до M сменных блоков (ТЭЗ), $M = 1, 2, 3, \dots, m$.

Пусть $d_i = 1$, если при i -й неисправности число подозреваемых сменных блоков не превышает M . В противном случае $d_i = 0$. Тогда ($n_i \leq M$)

$$k_{г.п.д.} = \sum_{i=1}^N d_i / N. \quad (8.3)$$

Для ЭВМ с развитой системой диагностирования для $M \leq 3$ обычно $k_{г.п.д.} \geq 0,9$. Это означает, что для 90 % неисправностей число предполагаемых неисправными сменных блоков, указанных в диагностическом справочнике, не превышает трех;

объем диагностического ядра h — доля той аппаратуры в общем объеме аппаратуры ЭВМ, которая должна быть заведомо исправной до начала процесса диагностирования. В качестве показателя объема диагностического ядра можно пользоваться также величиной

$$H = 1 - h. \quad (8.4)$$

Для ЭВМ, использующих принцип раскрутки и метод микродиагностирования, $H \geq 0,9$.

В качестве интегрального показателя системы диагностирования можно пользоваться коэффициентом

$$k_n = F D k_d k_{г.п.д.} H. \quad (8.5)$$

Для приведенных в качестве примеров количественных показателей системы диагностирования интегральный коэффициент

$$k_{\text{и}} = 0,95 \cdot 0,90 \cdot 0,75 \cdot 0,90 \cdot 0,90 \approx 0,51.$$

8.2. МЕТОД ДВУХЭТАПНОГО ДИАГНОСТИРОВАНИЯ

Метод двухэтапного диагностирования — это метод диагностирования, при котором объектами элементарных проверок на разных этапах диагностирования являются

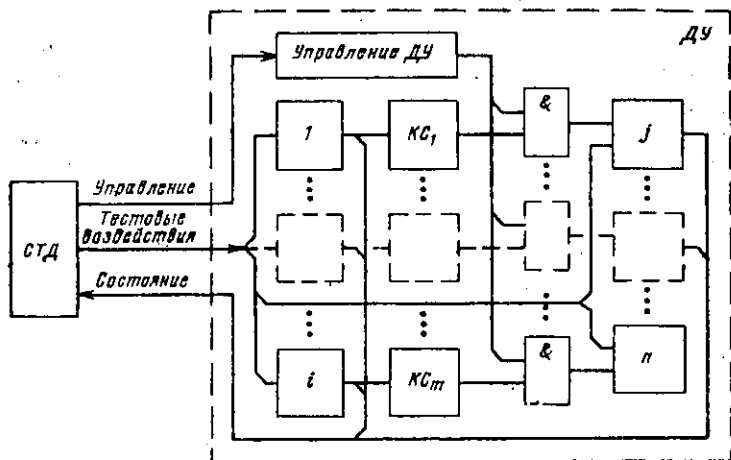


Рис. 8.7. Обобщенная схема системы диагностирования, реализующей метод двухэтапного диагностирования:

ДУ — диагностируемое устройство; 1, ..., i, j, ..., n — регистры; КС₁, ..., КС_m — комбинационные схемы

схемы с памятью (регистры и триггеры) и комбинационные схемы. Диагностическая информация, включающая в себя данные тестового воздействия, результат и состав контрольных точек элементарной проверки, адреса следующих элементарных проверок в алгоритме диагностирования, имеет стандартный формат, называемый *тестом локализации неисправностей* (ТЛН).

Обобщенная схема системы диагностирования, использующей метод двухэтапного диагностирования, показана на рис. 8.7.

Подача тестовых воздействий, снятие ответа, анализ и выдача результатов реализации алгоритма диагностирования выполняются с помощью стандартных диагностических операций «Установка», «Опрос», «Сравнение» и «Ветвление».

Установочная информация	Установочная информация для микропрограммы установки регистров процессора в требуемое для теста состояние
Управляющее слово	Управляющее слово задает адрес микрокоманды, содержащей проверяемую микрооперацию и число выполняемых микрокоманд
Адрес результата	Адрес диагностической области ОП, в которую в результате опроса записывается состояние проверяемого регистра
Маска	Маска выделяет проверяемые биты
Эталон	Эталон содержит ожидаемые результаты
Адрес перехода по удаче	Задаст начальный адрес следующего теста (адрес первого слова) при совпадении результата с эталоном
Адрес перехода по неудаче/ признак окончания номер теста	Для конечных тестов задает номер теста; для остальных тестов — начальный адрес (адрес первого слова) при несовпадении результата с эталоном

Рис. 8.8. Формат ТЛН

Стандартный формат ТЛН показан на рис. 8.8. Тест локализации неисправностей содержит установочную и управляющую информацию, адрес ячейки памяти, в которую записывается результат элементарной проверки, эталонный результат, адреса ТЛН, которым передается управление при совпадении и несовпадении результата с эталонным, и номер теста. Стандартные диагностические операции, последовательность которых приведена на рис. 8.9, могут быть реализованы аппаратурно или микропрограммно.

Диагностирование аппаратуры по этому методу выполняется в два этапа:

на первом этапе проверяются все регистры и триггеры, которые могут быть установлены с помощью операции «Установка» и опрошены по дополнительным выходам операцией «Опрос»;

на втором этапе проверяются все комбинационные схемы, а также регистры и триггеры, не имеющие непосредственной установки или опроса.

Каждая элементарная проверка, которой соответствует один ТЛН, выполняется следующим образом: с по-

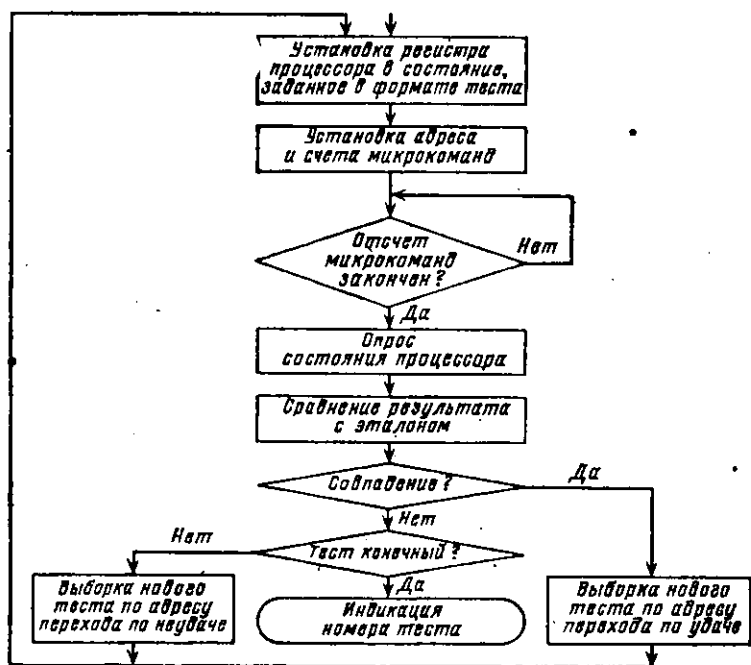


Рис. 8.9. Операции, выполняемые при диагностировании по методу двухэтапного диагностирования

мощью операции «Установка» устанавливаются регистры и триггеры ДУ, в том числе и не проверяемые данным ТЛН, в состояние, заданное установочной информацией ТЛН (установка регистров и триггеров может выполняться по существующим или дополнительным входам). Управляющая информация задает адрес микрокоманды (из числа рабочих микрокоманд), содержащей проверяемую микрооперацию и число микрокоманд, которые необходимо выполнить, начиная с указанной. В тестах первого этапа эта управляющая информация отсутствует, так как после

информационным входом оперативной или служебной памяти.

С помощью операции «Сравнение и ветвление» обеспечивается сравнение ответа ДУ на тестовое воздействие с эталонной информацией. В ТЛН задается адрес состояния проверяемого регистра или триггера в оперативной и служебной памяти, записываемого с помощью операции «Опрос», а также его эталонное состояние. Возможны два исхода операции «Сравнение и ветвление» — совпадение и несовпадение ответа с эталоном. Метод двухэтапного диагностирования использует, как правило, условный алгоритм диагностирования. Поэтому ТЛН содержит два адреса ветвления, задающих начальный адрес следующих ТЛН в оперативной памяти.

Для хранения ТЛН, как правило, используется магнитная лента, а для их ввода — стандартные или специальные каналы ввода.

Тесты локализации неисправностей обычно загружаются в оперативную память и подзагружаются в нее по окончании выполнения очередной группы ТЛН. Поэтому до начала диагностики по методу ТЛН проверяется оперативная память и микропрограммное управление.

При обнаружении отказа на пути индицируется номер теста, по которому в диагностическом справочнике отыскивается неисправный сменный блок.

В качестве примера реализации метода двухэтапного диагностирования рассмотрим систему диагностирования процессора ЭВМ ЕС-1030. Для нормальной загрузки и выполнения диагностических тестов процессора ЭВМ ЕС-1030 необходима исправность одного из секторных каналов и начальной области оперативной памяти (ОП). Поэтому вначале выполняется диагностирование ОП. Для этого имеется специальный блок, обеспечивающий проверку ОП в режимах записи и чтения нулей (единиц) тяжелого кода/обратного тяжелого кода. Неисправность ОП локализуется с точностью до адреса и бита [78].

Следующие стадии диагностирования, последовательность которых приведена на рис. 8.11, используют уже проверенную оперативную память.

На первой стадии диагностические тесты загружаются в начальную область ОП (первые 4 Кслов) и затем выполняются с помощью диагностического оборудования. Тесты расположены на магнитной ленте в виде массивов. После выполнения тестов очередного массива в ОП загружается и выполняется следующий массив тестов. Загрузка

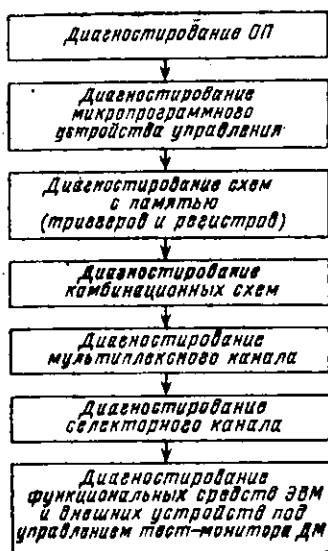


Рис. 8.11. Порядок диагностирования блоков и устройств ЭВМ ЕС-1030

тестов выполняется по одному из селекторных каналов в специальном режиме загрузки ТЛН.

На второй стадии диагностирования проверяется микропрограммная память процессора, которая используется на следующих стадиях диагностирования. В ней содержатся микропрограммы операций установки, опроса, сравнения и ветвления.

На третьей стадии диагностирования выполняется проверка триггеров (регистров) процессора. Эти тесты называются тестами нулевого цикла. Опрос состояния триггеров (регистров) выполняется по дополнительным линиям опроса. Триггеры (регистры) проверяются на установку в 0—1—0. Результаты проверки сравниваются с эталонными,

записанными в формате теста. Место неисправности определяется по номеру теста, который обнаружил несоответствие. В диагностическом справочнике тестов нулевого цикла номеру теста соответствует конструктивный адрес и название неисправного триггера на функциональной схеме.

С помощью тестов единичного цикла проверяются комбинационные схемы. Их последовательность определяется условным алгоритмом диагностирования. Тесты комбинационных схем выполняются следующим образом: с помощью операции установки в регистре процессора, расположенном на входе проверяемой комбинационной схемы, задается состояние, соответствующее входному тестовому воздействию. Выполняется микрооперация приема выходного сигнала комбинационной схемы в регистр, расположенный на выходе комбинационной схемы. Состояние этого регистра записывается в диагностическую область ОП, а затем сравнивается с эталонным. В зависимости от исхода теста выполняется переход к следующему тесту. При обнаружении неисправности индицируется номер тес-

та. В диагностическом справочнике тестов единичного цикла указаны не только подозреваемые ТЭЗ, но и значения сигналов на входах, промежуточных точках и выходах комбинационной схемы. Такая подробная информация позволяет уточнить локализацию до монтажных связей или микросхем. На следующих стадиях диагностирования, использующих другие методы диагностирования, проверяются мультиплексный и селекторный каналы, а также функциональные средства ЭВМ с помощью тест-секций диагностического монитора.

8.3. МЕТОД ПОСЛЕДОВАТЕЛЬНОГО СКАНИРОВАНИЯ

Метод последовательного сканирования является вариантом метода двухэтапного диагностирования, при котором схемы с памятью (регистры и триггеры) в режиме диагностирования превращаются в один сдвигающий регистр с возможностью установки его в произвольное состояние и опроса с помощью простой операции сдвига [70, 108].

Обобщенная схема системы диагностирования, использующей метод последовательного сканирования, показана на рис. 8.12.

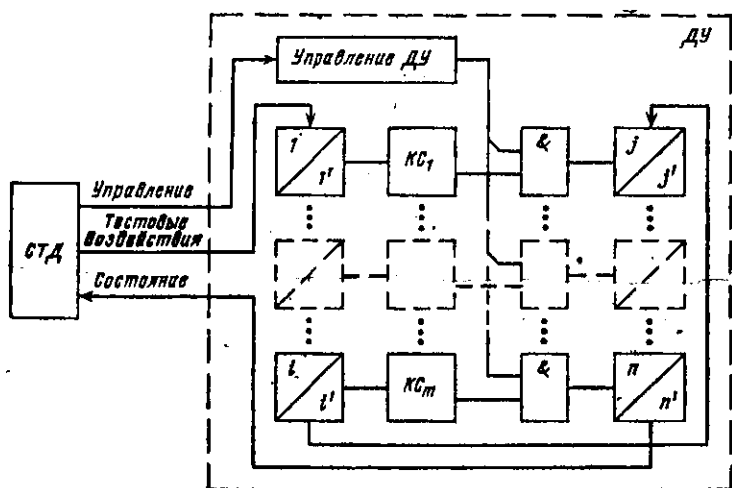


Рис. 8.12. Обобщенная схема системы диагностирования, реализующей метод последовательного сканирования:

$i, \dots, l, j, \dots, n$ — основная часть регистра; $i', \dots, l', j', \dots, n'$ — дополнительная часть регистра (триггеры образования сдвигового регистра)

Этот метод получил распространение в ЭВМ на больших интегральных микросхемах (БИС). Вместе с очевидными достоинствами БИС их использование затрудняет проблему диагностирования ЭВМ в связи с ограниченными возможностями доступа к схемам, расположенным внутри БИС. При диагностировании ЭВМ, построенной на БИС,

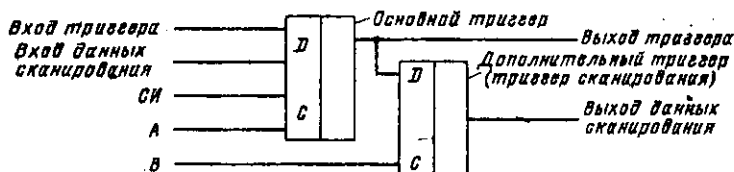


Рис. 8.13. Основной триггер и триггер сканирования

возникает проблема проверки БИС, содержащих комбинационные схемы и схемы с памятью при небольшом числе дополнительных входов и выходов.

Для превращения всех триггеров БИС в один сдвигающий регистр каждому триггеру логической схемы придается дополнительный триггер типа *D*, причем каждая пара триггеров, основной и дополнительный, соединяется таким образом, что образует один разряд сдвигающего регистра.

Первый триггер каждой пары, или триггер данных (рис. 8.13), используется как для выполнения основных функций при работе машины, так и для тестирования. Поэтому он имеет два входа данных: рабочий и сканирования, а также два входа синхронизации: от процессора и от средств тестового диагностирования.

Второй триггер пары, или триггер сканирования, используется главным образом для тестирования. Его вход постоянно соединен с выходом первого триггера, а синхросигнал поступает только от средств тестового диагностирования.

В режиме диагностирования состояние первого триггера передается второму триггеру по сигналам СТД, и таким образом могут быть опрошены СТД, которые посылают синхросигнал на второй триггер и путем сдвига выдают его информацию через выходной контакт данных сканирования.

Эти триггерные пары соединяются последовательно в несколько сдвигающих регистров. Выход данных одной пары триггеров соединяется с входами данных сканирования другой пары и т. д. (рис. 8.14).

Средства тестового диагностирования могут подавать синхросигналы на все триггеры сканирования и путем сдвига выдавать их содержимое в виде последовательности бит по одной линии. Поскольку каждый бит в этой последовательности соответствует своей триггерной паре, можно определить состояние каждого триггера логической схемы.

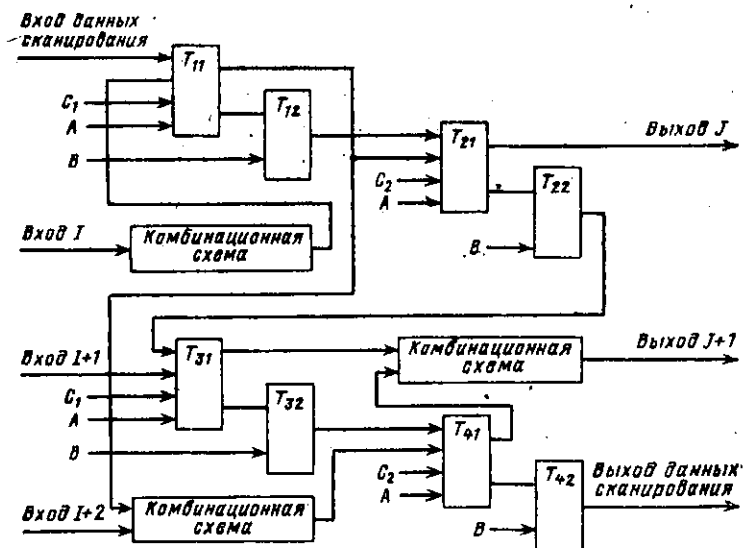


Рис. 8.14. Соединение триггеров схемы в режиме диагностирования

Средства тестового диагностирования могут задавать любое состояние триггеров, подавая на линию входа данных сканирования требуемую установочную последовательность.

Диагностирование выполняется в два этапа.

Первый этап. Диагностирование схем с памятью (регистров и триггеров). Выполняется следующим образом: устанавливается режим сдвигающего регистра; осуществляется проверка сдвигающего регистра и, таким образом, всех схем с памятью путем последовательно сдвига по нему нулей и единиц.

Второй этап. Диагностирование комбинационных схем. Выполняется следующим образом: устанавливается режим сдвигающего регистра;

входной регистр комбинационной схемы устанавливается в состояние, соответствующее тестовому воздействию, путем подачи последовательного потока данных на вход сдвигающего регистра;

выполняется переход в нормальный режим;

выполняется микрооперация передачи сигналов с выходов комбинационной схемы;

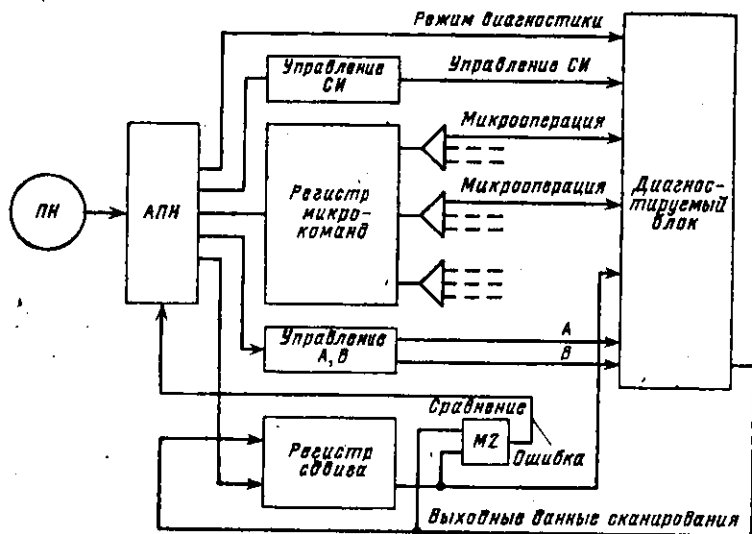


Рис. 8.15. Аппаратура СТД, используемая при методе последовательного сканирования:

ПН — пультовый накопитель; АПН — адаптер пультового накопителя

выполняется опрос состояния выходного регистра комбинационной схемы (результата) путем последовательного сдвига его содержимого в аппаратуру тестового диагностирования;

осуществляется сравнение результата с эталоном.

На рис. 8.15 показан состав средств тестового диагностирования, используемых при методе последовательного сканирования [109].

Процедура диагностирования заключается в следующем. Из адаптера пультового накопителя (АПН) в регистр диагностики и сдвигающий регистр загружается информация, содержащая биты запуска синхросигналов А и В, а также тестовый набор. После этого информация тес-

тового набора последовательно, под управлением синхросигналов А и В сдвигается, поступает на вход проверяемого блока и далее на установку состояния триггеров проверяемого блока. После этого выполняется проверяемая микрооперация и данные состояния триггеров проверяемого блока начинают последовательно поступать в сдвигающий регистр. До этого в сдвигающий регистр загружается эталонная информация и при последовательном сканировании результатов тестирования в схеме сложения по модулю 2 происходит сравнение результата с эталоном.

При обнаружении несовпадения происходит сообщение о неисправности.

8.4. МЕТОД МИКРОДИАГНОСТИРОВАНИЯ

Метод микродиагностирования характеризуется тем, что объектом элементарной проверки является аппаратура, участвующая в выполнении микрооперации.

Микропрограмма проверки очередной микрооперации использует уже проверенные микрооперации и тракты передачи информации. Транспортировка тестового воздействия на вход проверяемой аппаратуры выполняется с помощью имеющегося в ЭВМ набора микроопераций по существующим информационным трактам.

Снятие ответа и передача его с выхода проверяемой аппаратуры на входы схем сравнения выполняются либо

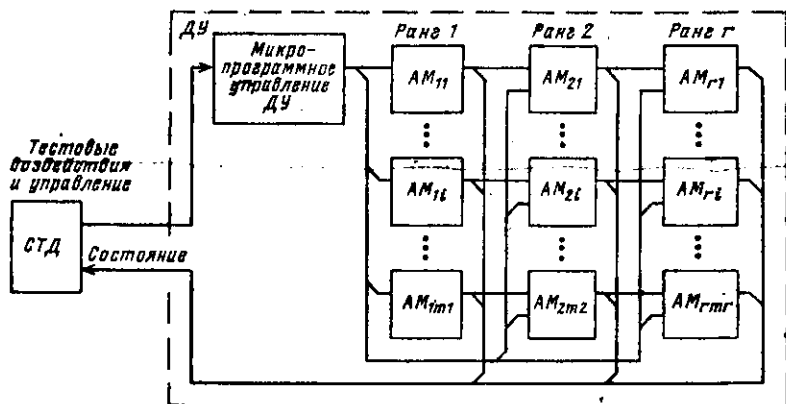


Рис. 8.16. Обобщенная схема системы диагностирования, реализующей метод микродиагностирования

микропрограммно, либо с помощью специальных диагностических операций «Опрос» и «Сравнение».

Обобщенная схема системы диагностирования показана на рис. 8.16.

Совокупность процедур, диагностических микропрограмм и специальных схем, обеспечивающих транспортировку тестового набора на вход проверяемого блока, выполнение проверяемой микрооперации, транспортировку результатов проверки к схемам анализа, сравнение с эталоном и ветвление по результатам сравнения, называется *микродиагностикой*.

Различают два типа микродиагностики: встроенную и загружаемую.

В случае *встроенной микродиагностики* диагностические микропрограммы размещаются в постоянной микропрограммной памяти ЭВМ, а при загружаемой — на внешнем носителе данных.

При хранении в постоянной микропрограммной памяти микродиагностика представляет собой обычную микропрограмму, использующую стандартный набор микроопераций. Однако вследствие ограниченного объема постоянной микропрограммной памяти на объем микродиагностики накладываются довольно жесткие ограничения, в результате чего приходится использовать различные способы сжатия информации. Для этой цели иногда используют специальные микрокоманды генерации тестовых наборов. Это позволяет уменьшить требуемый для тестовых констант объем микропрограммной памяти.

Как правило, при хранении микродиагностики в постоянной микропрограммной памяти для транспортировки результатов проверки к месту сравнения с эталоном используются стандартные микрооперации, а для сравнения — такие схемы, как сумматор, схемы контроля или анализа условий. В качестве микропрограммы анализа используется также микропрограмма опроса состояния схем контроля ЭВМ.

Встроенная микродиагностика применяется обычно в малых ЭВМ с небольшим объемом микродиагностики.

Для средних и больших ЭВМ при большом объеме микродиагностики применяется *загружаемая микродиагностика*. Существует несколько вариантов загрузки и выполнения загружаемой микродиагностики [46]:

внешний носитель данных — регистр микрокоманд (РгМк) (рис. 8.17, а);

внешний носитель данных — оперативная память (ОП) — регистр микрокоманд (рис. 8.17, б);

внешний носитель данных — загружаемая управляющая память (ЗУП) микрокоманд — регистр микрокоманд (рис. 8.17, в).

В качестве устройства ввода микродиагностики чаще всего используются так называемые пульты накопители на гибких магнитных дисках или кассетных магнитных лентах.

Первый вариант загрузки скорее имитирует «быстрый» тактовый режим, чем выполнение микрокоманд с реальным быстродействием, так как накопление и выполнение микрокоманд определяются скоростью ввода данных с внешнего носителя. Микрокоманды выполняются по мере их поступления из внешнего носителя данных.

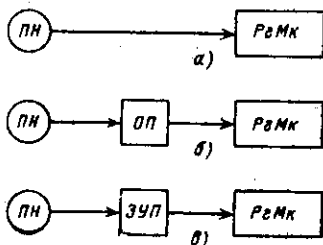


Рис. 8.17. Варианты загрузки и выполнения загружаемой микродиагностики

Второй вариант загрузки предусматривает возможность хранения и выполнения микрокоманд из основной памяти ЭВМ, т.е. совместимость форматов оперативной и управляющей памяти. В этом варианте должен быть предусмотрен специальный вход в регистр микрокоманд из оперативной памяти,

Третий вариант загрузки обеспечивает загрузку в управляющую память микродиагностики определенного объема и выполнение ее с реальным быстродействием. По окончании выполнения загружается следующая порция микродиагностики.

Существуют и другие варианты загрузки и выполнения, несущественно отличающиеся от приведенных выше. Возможно также использование разных вариантов загрузки и выполнения на разных этапах диагностирования ЭВМ.

Для средних и больших ЭВМ с хранением микродиагностики на внешних носителях данных, для опроса состояния и сравнения его с эталоном используется дополнительная аппаратура. В последнее время эти функции все больше передаются так называемым сервисным процессорам, имеющим универсальные возможности по управлению пультными накопителями, опросу состояния ЭВМ, сравнению результатов с эталонными и индикации списка воз-

можных неисправностей. При микродиагностировании с использованием дополнительной аппаратуры средства тестового диагностирования выполняют специальные диагностические операции, такие как запуск микрокоманд, опрос состояния, сравнение с эталоном и сообщение о неисправности. Процедура выполнения микродиагностики обычно такова: средства тестового диагностирования за-

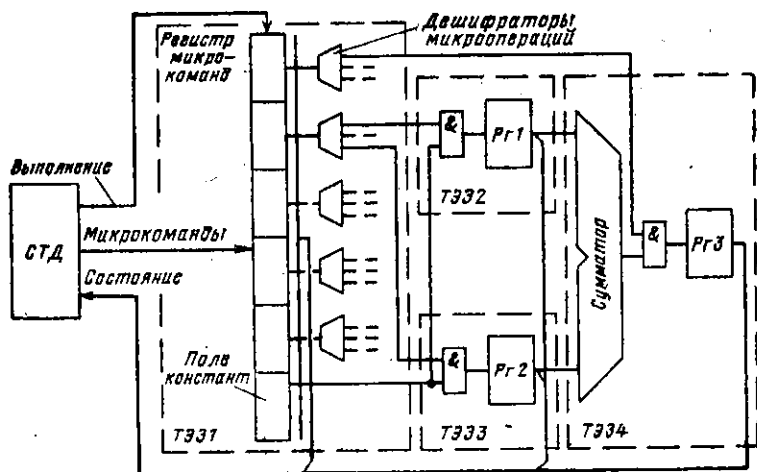


Рис. 8.18. Фрагмент аппаратуры диагностируемого устройства

гружают в ЭВМ микрокоманды и дают приказ на их выполнение; ЭВМ обрабатывает микрокоманды, после чего средства тестового диагностирования производят опрос состояния, сравнение с эталоном и сообщение о неисправности. Обычно при микродиагностике тестовые наборы являются частью микрокоманды (поле констант).

Глубина поиска дефекта при микродиагностике зависит от числа схем, для которых предусмотрена возможность непосредственного опроса состояния. В связи с этим в современных ЭВМ имеется возможность непосредственного опроса состояния практически всех триггеров и регистров ЭВМ.

Рассмотрим пример реализации метода микродиагностирования.

На рис. 8.18 показан фрагмент аппаратуры диагностируемого устройства, а на рис. 8.19 — документация микродиагностики, обеспечивающей проверку этого фрагмента.

Операция	Признак операции	Комментарий
Загрузка RгМк	Д	В RгМк загружаются «все нули»
Сравнение RгМк=x' 000 ... 0'	Д	Проверка RгМк на «все нули»
** Останов 1 * * * * *		
* RгМк ≠ x' 000 ... 0'		*
* Неисправен ТЭ31		*
* * * * *		
Загрузка RгМк	Д	В RгМк загружаются «все единицы»
Сравнение RгМк=x' FFF ... F'	Д	Проверка RгМк на «все единицы»
** Останов 2 * * * * *		
* RгМк ≠ x' FFF ... F'		*
* Неисправен ТЭ31		*
* * * * *		
Загрузка RгМк	Д	Загрузка в RгМк микрокоманды передачи поля константы К в Rг1
Rг1:=К	М	Выполнение микрокоманды
Сравнение Rг1=x' FF'	Д	Проверка Rг1 на единицы
** Останов 3 * * * * *		
* Rг1 ≠ x' FF'		*
* Неисправен ТЭ32		*
* * * * *		
Загрузка RгМк	Д	Загрузка в RгМк микрокоманды Rг2:=К
Rг2:=К	М	Выполнение микрокоманды
Сравнение Rг2=x' FF'	Д	Проверка Rг2 на единицы
** Останов 4 * * * * *		
* Rг ≠ x' FF'		*
* Неисправен ТЭ33		*
* * * * *		
Загрузка RгМк	Д	Загрузка в RгМк микрокоманды Rг3:=Rг1+Rг2
Rг3:=Rг1+Rг2	М	Сложение результата и передача его в Rг3
Сравнение Rг3=x' FE'	Д	Проверка Rг3 на x' FE'
** Останов 5 * * * * *		
* Rг3 ≠ x' FE'		*
* Неисправен ТЭ34		*
* * * * *		

Рис. 8.19. Документация микродиагностики:

Д — диагностическая операция; М — микрокоманда

Регистр микрокоманд устанавливается средствами тестового диагностирования с помощью диагностической операции «Загрузка РгМк».

Состояние регистров поступает в СТД, где выполняется диагностическая операция сравнения с эталоном.

При несовпадении результата с эталоном происходит останов с индикацией номера останова.

8.5. МЕТОД ЭТАЛОННЫХ СОСТОЯНИЙ

Метод эталонных состояний характеризуется тем, что объектом элементарных проверок является аппаратура, используемая на одном или нескольких тактах выполнения рабочего алгоритма функционирования, реализуемого в

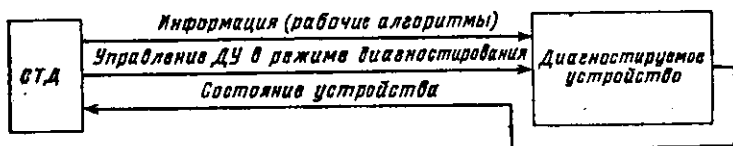


Рис. 8.20. Обобщенная схема системы диагностирования, реализующей метод эталонных состояний

режиме диагностирования. В качестве результата элементарной проверки используется состояние аппаратурных средств диагностируемого устройства.

Процесс диагностирования по методу эталонных состояний, заключается в потактовом выполнении рабочих алгоритмов ДУ, опросе состояния ДУ на каждом такте, сравнении состояния ДУ с эталонным и ветвлении в зависимости от исхода сравнения к выполнению следующего такта или сообщению о неисправности.

При реализации метода эталонных состояний средства тестового диагностирования представляют собой совокупность аппаратурных и программных средств.

Обобщенная схема системы диагностирования, реализующей метод эталонных состояний, приведена на рис. 8.20.

При представлении алгоритмов операций ЭВМ в виде графов каждому пути i из множества путей на графе можно поставить в соответствие последовательность состояний ЭВМ на каждом такте:

$$S_{i0}, S_{i1}, \dots, S_{in}, \dots, S_{in}$$

Рис. 8.21. Процедура диагностирования по методу эталонных состояний:

i — номер такта ветви алгоритма; j — номер ветви алгоритма

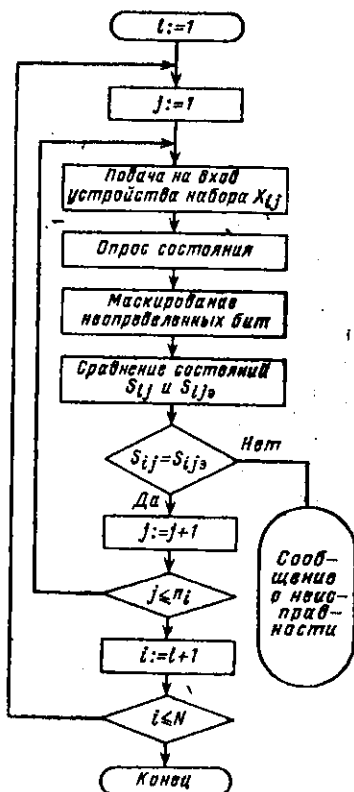
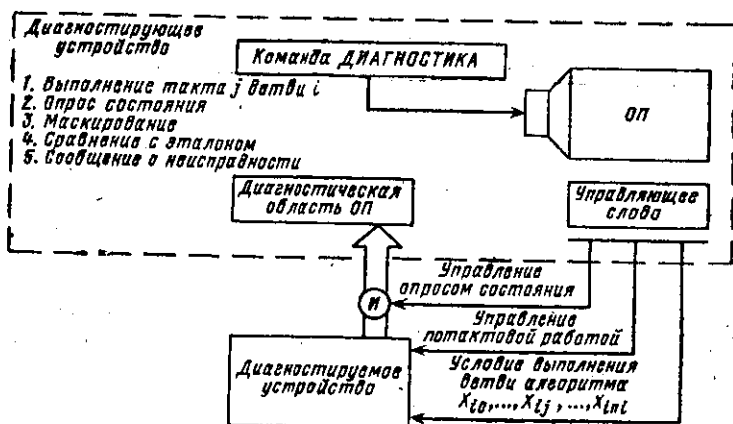


Рис. 8.22. Схема взаимодействия диагностирующего и диагностируемого устройств при диагностировании по методу эталонных состояний



где n — число вершин граф-схемы алгоритма, соответствующее числу тактов выполнения операции с конкретными условиями. Эталонной последовательностью состояний считается последовательность состояний S_{ij}^0 , $j=0, 1, \dots, n$, имеющих место при отсутствии ошибок.

Проверка выполняется путем сравнения реального состояния ЭВМ S_{ij} на j -м такте i -го пути с эталонным S_{ij}^0 . Несовпадение S_{ij} и S_{ij}^0 является признаком неисправности.

Процедура диагностирования по методу эталонных состояний приведена на рис. 8.21.

Для реализации метода эталонных состояний средства тестового диагностирования должны иметь:

- средства управления потактовой работой ЭВМ;
- средства опроса состояния ЭВМ;
- средства сравнения состояния с эталонным и средства сообщения о неисправности.

Обычно этот метод используется в тех случаях, когда средства тестового диагностирования имеют достаточно большие возможности. Например, этот метод может использоваться при диагностировании каналов с помощью процессора. Наибольшее применение этот метод находит в устройствах со схемной интерпретацией алгоритмов функционирования.

В силу неопределенности состояний некоторых триггеров каждому состоянию S_{ij} может соответствовать некоторое подмножество состояний S_{ijk} , где $k=0, 1, \dots, m$, m — множество неопределенных состояний. Поэтому обычно до сравнения с эталоном выполняется маскирование состояний. Маска снимает неопределенные состояния [78].

Обычно управление потактовой работой устройства и опрос состояния устройства выполняются с помощью команды ДИАГНОСТИКА, а сравнение с эталоном, маскирование и сообщение о неисправности — с помощью команд на программном уровне.

Команда ДИАГНОСТИКА адресует управляющее слово в ОП, которое поступает на вход диагностируемого устройства, как показано на рис. 8.22. Сочетание бит управляющего слова обеспечивает продвижение тактов, а также опрос состояния и запись его в ОП.

Остальные операции, такие как маскирование состояния с целью исключения неопределенных бит, сравнение его с эталонным состоянием и сообщение о неисправности, выполняются программой диагностирующего устройства [78].

8.6. МЕТОД ДИАГНОСТИРОВАНИЯ, ОРИЕНТИРОВАННЫЙ НА ПРОВЕРКУ СМЕННЫХ БЛОКОВ

Этот метод диагностирования характеризуется тем, что объектом элементарных проверок являются сменные блоки. Тестовые воздействия подаются на входы сменных блоков, а ответы снимаются с их выходов непосредственно или через дополнительную аппаратуру [56, 105].

Использование этого метода позволяет локализовать неисправность в ЭВМ с помощью диагностических тестов для сменных блоков. Такой подход позволяет сократить затраты на разработку диагностического обеспечения ЭВМ, состоящего, как правило, из диагностических тестов ЭВМ и диагностических тестов сменных блоков. В большинстве случаев в качестве сменных блоков ЭВМ используются ТЭЗ. Тогда применение данного метода позволяет производить диагностирование ЭВМ с помощью диагностических тестов ТЭЗ.

Существуют два варианта реализации данного метода.

Вариант 1. В каждый сменный блок вводится дополнительная аппаратура, обеспечивающая подачу на вход каждого сменного блока тестового воздействия от STD и передачу ответов к точкам опроса [105].

Эта аппаратура представляет собой селекторы, которые в нормальном режиме работы ЭВМ пропускают на выходы сменного блока состояния его выходов, а в режиме управления — состояния входов, как бы шунтируя расположенную в сменном блоке аппаратуру. В зависимости от соотношения числа входов и выходов существуют три типа сменных блоков:

а) сменный блок типа 1 удовлетворяет условию

$$N(I_{в.с}) \leq N(Z) \leq N(I_{в.с}) + N(I_{п.в}),$$

где $N(I_{в.с})$ — число входов, связанных с другими сменными блоками; $N(I_{п.в})$ — число первичных входов; $N(Z)$ — число выходов сменного блока. Структура сменного блока типа 1 приведена на рис. 8.23. На управляющий вход c в рабочем режиме подается 0 и на выход селекторов поступают состояния выходов основной аппаратуры блока, обозначенной B_0 . В режиме управления $c=1$, и на выход блока поступают состояния его входов;

б) сменный блок типа 2 удовлетворяет условию

$$N(I_{в.с}) > N(Z), k = [N(I_{в.с})/N(Z)],$$

где скобки $[]$ означают округление дроби до ближайшего целого числа. Структура сменного блока типа 2 ($k=2$) по-

входам диагностирующего устройства. В каждый сменный блок вводится дополнительная аппаратура для передачи состояния входов блока на его выход.

Все сменные блоки одного уровня (ранга) проверяются одновременно и управляются одним управляющим сигналом с. Остальные управляющие сигналы формируются в СТД с помощью счетчиков. На рис. 8.26 дан пример диагностируемого устройства, а на рис. 8.27 — его диагностирование с помощью СТД и дополнительной аппаратуры, встраиваемой в каждый сменный блок.

Рис. 8.26. Пример диагностируемого устройства

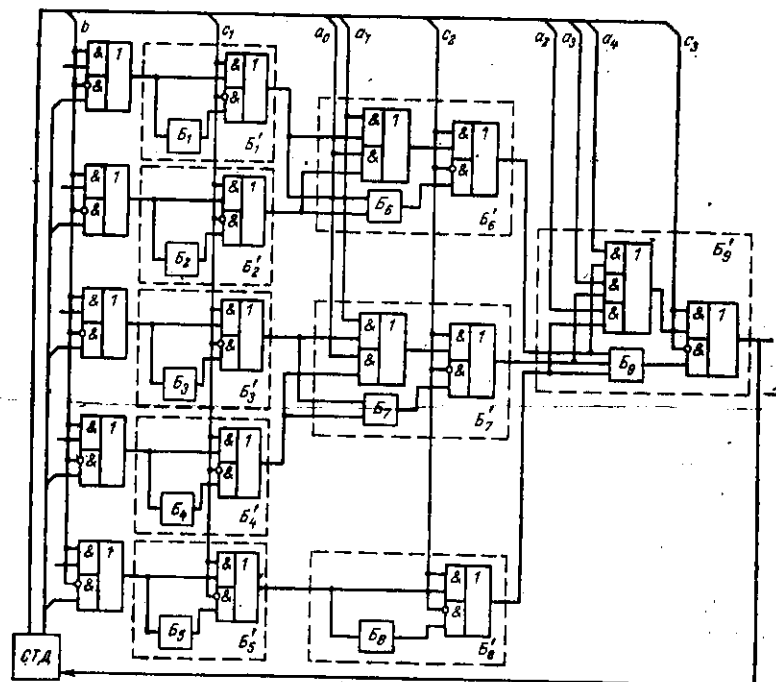
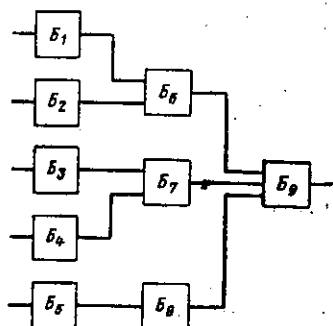


Рис. 8.27. Диагностируемое устройство с дополнительной аппаратурой для диагностирования сменных блоков

Диагностирование заключается в настройке схем управления на проверку сменного блока i -го уровня, подаче тестовых воздействий на его вход, опросе результатов и сравнении их с эталонными.

При тестировании сменных блоков i -го уровня все управляющие сигналы кроме i -го равны 1. Так, при тестировании сменных блоков первого уровня (рис. 8.27) $c_1 c_2 c_3 = 011$, второго уровня — 101 и третьего — 110.

Вариант 2. Селекторы устанавливаются только в сменных блоках, соединенных с первичными входами диагностируемого устройства, и на обратных связях, обеспечивая их разрыв в режиме диагностирования [56].

Выходы каждого сменного блока соединены со средствами тестового диагностирования. Средства тестового диагностирования имеют информационные выходы, соединенные с дополнительными входами всех селекторов и служащие для выдачи тестовых воздействий на блоки устройства. Кроме того, СТД имеют один управляющий выход, соединенный с управляющими входами дополнительных селекторов устройства.

На входы СТД, которые соединены с выходами сменных блоков диагностируемого устройства, поступают ответы последних на тестовые воздействия. В СТД выполняет-

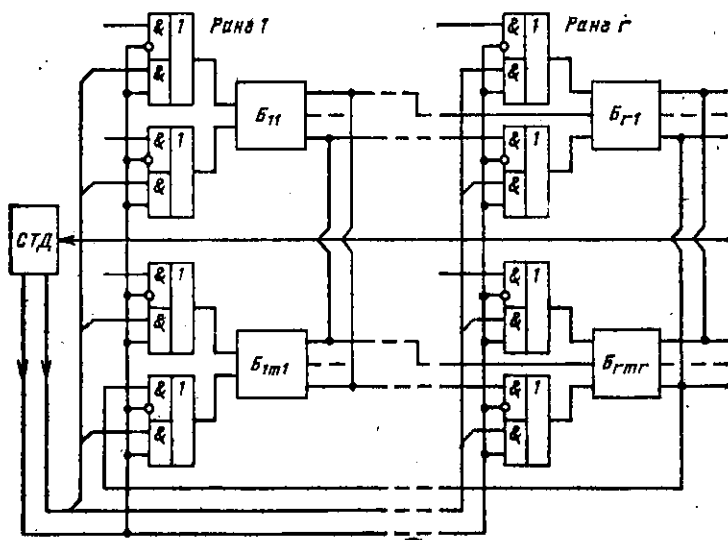


Рис. 8.28. Обобщенная схема второго варианта системы диагностирования, ориентированной на проверку сменных блоков

ся анализ этих ответов, в результате которого выдается диагноз.

На рис. 8.28 показана обобщенная схема, отражающая связи СТД с блоками устройства и дополнительными селекторами.

При определении последовательности диагностирования сменные блоки устройства ранжируются, т. е. располагаются по уровням по отношению к первичным входам устройства. Диагностирование сменных блоков выполняется последовательно в порядке возрастания номеров рангов. Тесты для сменных блоков j -го ранга ($j \geq 2$), построенные первоначально в терминах этих блоков, пересчитываются затем в терминах тех входов блоков $(j-s)$ -х рангов ($s \geq 1$), к которым подключены селекторы. Это — внешние входы устройства, а также входы, принадлежащие местам обрыва обратных связей.

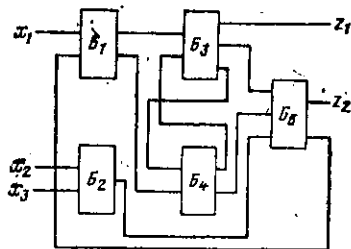


Рис. 8.29. Пример диагностируемого устройства

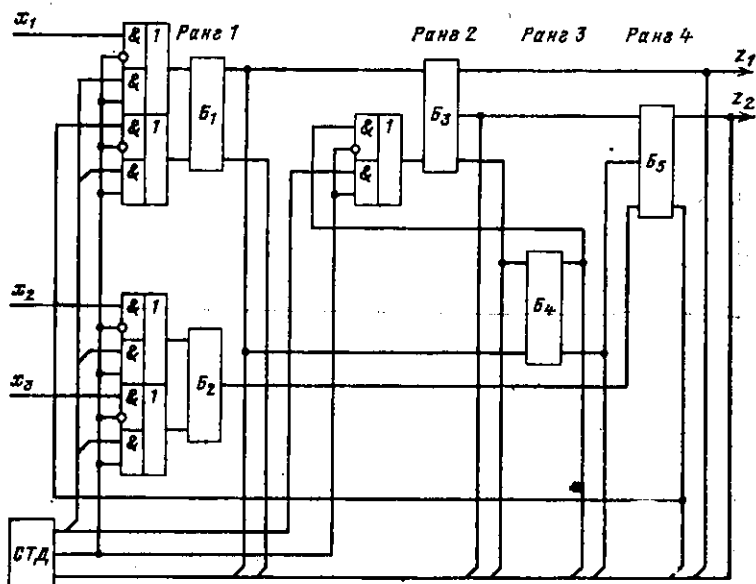


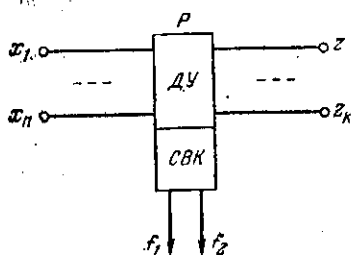
Рис. 8.30. Диагностируемое устройство с дополнительной аппаратурой для диагностирования сменных блоков

На рис. 8.29 дан пример диагностируемого устройства, а на рис. 8.30 изображено то же устройство с дополнительными средствами, обеспечивающими диагностирование сменных блоков [56].

8.7. МЕТОД ДИАГНОСТИРОВАНИЯ С ПОМОЩЬЮ СХЕМ ВСТРОЕННОГО КОНТРОЛЯ

Этот метод характеризуется тем, что объектом элементарной проверки является сменный блок, а средствами функционального диагностирования являются схемы встроенного контроля (СВК), конструктивно совмещенные с каждым сменным блоком.

На рис. 8.31 показаны диагностируемое устройство и схемы встроенного контроля, образующие самопроверяемый сменный блок.



при полной проверяемости ДУ и самопроверяемости СВК.

Определение самопроверяемости СВК было дано в гл. 5.

Поэтому здесь приводится только определение полной проверяемости ДУ.

Диагностируемое устройство называется *полностью проверяемым*, если любая его

Рис. 8.31. Самопроверяемый сменный блок

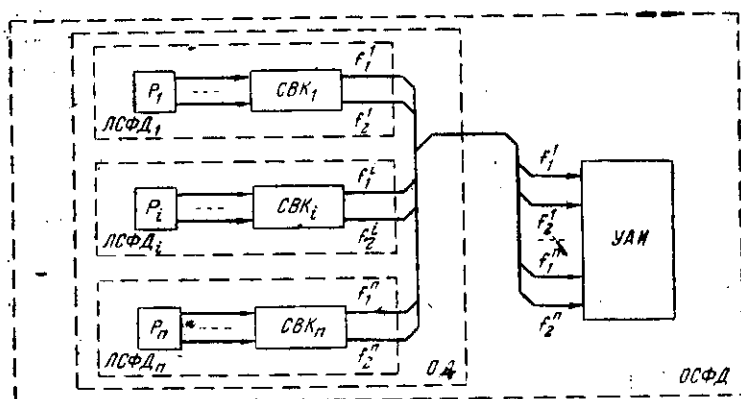


Рис. 8.32. Структура системы диагностирования, использующей схемы встроенного контроля

неисправность заданного класса обнаруживается СВК в момент ее первого проявления на выходных устройствах [56].

Требование полной проверяемости ДУ и самопроверяемости СВК приводит к значительным аппаратным затратам, что ограничивает применяемость данного метода устройствами, реализованными в основном на больших интегральных микросхемах.

На рис. 8.32 приведена структура системы функционального диагностирования. Локальными средствами функционального диагностирования ЛСФД являются самопроверяемые СВК с парами выходов f'_1, f'_2 , приданные каждому сменному блоку B_i , общим средством функционального диагностирования ОСФД — устройством анализа и индикации УАИ. Назначением последнего является синхронизация сигналов ошибок от сменных блоков с учетом их связей, предотвращение возможной неоднозначности индикации из-за распространения сигналов ошибок и однозначная индикация неисправного блока.

Достоинством метода диагностирования с помощью схем встроенного контроля является практически мгновенное диагностирование сбоев и отказов, сокращение затрат на локализацию перемежающихся отказов и на разработку диагностических тестов.

8.8. МЕТОД ДИАГНОСТИРОВАНИЯ С ПОМОЩЬЮ САМОПРОВЕРЯЕМОГО ДУБЛИРОВАНИЯ

Этот метод аналогичен предыдущему, так как он тоже основан на принципе самопроверяемости сменных блоков. Разница состоит в том, что самопроверяемость сменных блоков достигается введением в него дублирующей аппа-

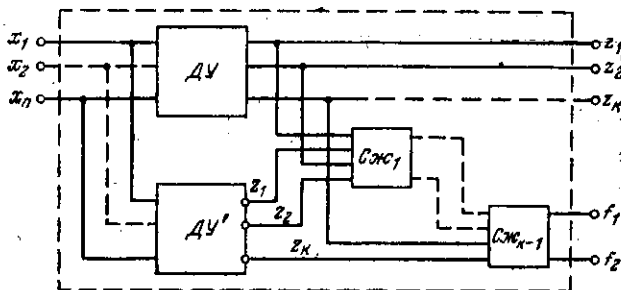


Рис. 8.33. Структурная схема самопроверяемого блока:

$СЖ_1, \dots, СЖ_{k-1}$ — схемы сжатия

ратуры и самопроверяемых схем сжатия, обеспечивающих получение сводного сигнала ошибки, свидетельствующего о неисправности сменного блока. На рис. 8.33 приведена структурная схема самопроверяемого блока. Этот способ обеспечения самопроверяемости приводит к большим дополнительным затратам аппаратуры, что оправдывает его применение в больших интегральных микросхемах. При реализации ЭВМ на больших и сверхбольших интегральных микросхемах последние часто используются неполностью, так как ограничивающим фактором является не число вентилей БИС, а число выводов. Поэтому введение в БИС дублирующих схем, обеспечивающих ее самопроверяемость, позволяет более полно использовать возможность БИС без значительного увеличения объема аппаратуры [56].

8.9. МЕТОД ДИАГНОСТИРОВАНИЯ ПО РЕЗУЛЬТАТАМ РЕГИСТРАЦИИ СОСТОЯНИЯ

Этот метод диагностирования характеризуется тем, что неисправность или сбой локализуется по состоянию ЭВМ, зарегистрированному в момент проявления ошибки и содержащему информацию о состоянии схем контроля, регистров ЭВМ, адресов микрокоманд, предшествующих моменту появления ошибки, и другую информацию. Место возникновения ошибки определяется по зарегистрированному состоянию путем прослеживания трассы ошибки от места ее проявления до места ее возникновения. Диагноз выполняется с помощью программных средств диагностирования самой ЭВМ, если диагностируется место возникновения сбоя, либо другой ЭВМ, если диагностируется отказ. В ЭВМ, имеющих сервисные процессоры, диагноз выполняется с помощью микропрограмм сервисного процессора.

Для пояснения метода диагностирования по регистрации состояния рассмотрим схему, показанную на рис. 8.34. Эта схема размещена в трех разных блоках $B_1 - B_3$. Выходы регистров $P_{g1} - P_{g3}$, триггеров ошибок $T_{gOш1} - T_{gOш3}$, а также состояние регистра микрокоманд (на схеме не показан) поступают на регистрацию состояния.

Предположим, что в момент возникновения ошибки зарегистрировано следующее состояние:

$$T_{gOш1} = 1; P_{g1} (0-7, K) = 11101111 \text{ 1};$$

$$T_{gOш2} = 0; P_{g2} (0-7, K) = 00000000 \text{ 1};$$

$$T_{gOш3} = 0; P_{g3} (0-7, K) = 11111111 \text{ 1}.$$

Регистр микрокоманд содержит код микрооперации $P_{g1} := P_{g3}$.

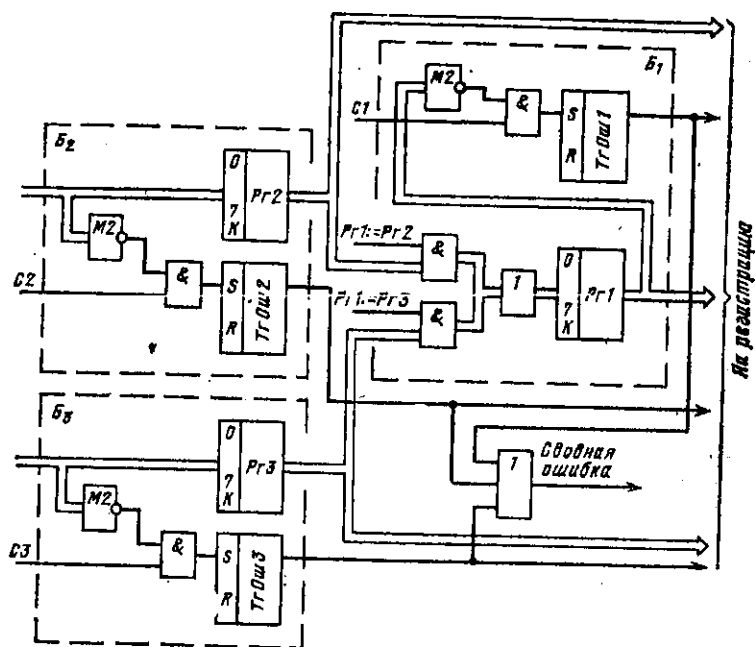


Рис. 8.34. Пример к методу диагностирования по регистрации состояния

После анализа $TgOsh1$ программные или микропрограммные средства диагностирования анализируют состояние $Pr1$ с целью обнаружения в нем несоответствия информационных и контрольных бит. Поскольку такое несоответствие обнаружено, выполняется анализ регистра передатчика. В конкретном случае это — регистр $Pr3$, так как регистр микрокоманд в момент сбоя содержал код микрооперации $Pr1 := Pr3$. Анализ содержимого регистра $Pr3$ показывает отсутствие в нем ошибок. В результате этого делается заключение о том, что наиболее вероятной причиной ошибки является сменный блок B_1 или связи между блоком B_3 и B_1 .

8.10 СЕРИАЛИЗАЦИЯ ДАННЫХ О СОСТОЯНИИ ЭВМ

При осуществлении процессов технического обслуживания необходимо иметь представление о состоянии ЭВМ. Так, при возникновении в ЭВМ ошибки ее состояние должно быть зарегистрировано для последующего анализа, а при диагностировании ЭВМ ее состояние должно передаваться в средства тестового диагностирования.

Состояние ЭВМ определяется состоянием управляющих триггеров, основных регистров и счетчиков (регистра кода операции, счетчика тактов, регистра кода прерывания и т. п.), уровнями сигналов на соответствующих шинах. В современных ЭВМ число двоичных элементов (сигналов), совокупность состояний которых определяет состояние машины, достигает нескольких тысяч. Все эти сигналы индицируются на пульте управления машины.

Поскольку так или иначе все сигналы о состоянии машины выдаются на пульт управления, а кроме того, пульт управления содержит аппаратуру и имеет соответствующие цепи связи для ручного задания режима работы, адреса, кода операции, потактового режима работы, оказывается целесообразным в меру возможности использовать аппаратуру и цепи пульта при построении системы диагностирования. Поэтому, как это имеет место в машине ЕС-1045, средства тестового диагностирования (сервисный адаптер, пультовый накопитель) непосредственно связываются и в определенной степени конструктивно объединяются с аппаратурой пульта управления.

При построении системы индикации и регистрации состояния аппаратуры машины возникает проблема передачи на пульт управления сигналов о состояниях нескольких тысяч двоичных элементов машины. Совокупность сигналов о состояниях в некоторый момент времени элементов, характеризующих состояние машины, может рассматриваться как некоторый многобитный (несколько тысяч бит) параллельный код. Этот код должен быть передан на пульт и воспроизведен (по существу также в виде параллельного кода) индикаторами пульта управления. Однако передача объединенного кода состояния машины на пульт параллельным кодом практически невозможна, так как для этого потребуется огромное число (несколько тысяч) проводов (на каждый двоичный сигнал один или два провода соответственно при однофазной и парафазной связи).

Проблема решается путем преобразования параллельного кода состояния машины в последовательный (это преобразование называется *сериализацией*), передачи информации на пульт последовательным кодом, а затем преобразования последовательного кода в параллельный при отображении на индикаторах пульта и запоминании в памяти пульта (*десериализация*).

Рассмотрим принцип построения системы сериализации на примере машины ЕС-1045. В этой машине система сериализации допускает преобразование формы представления информации о

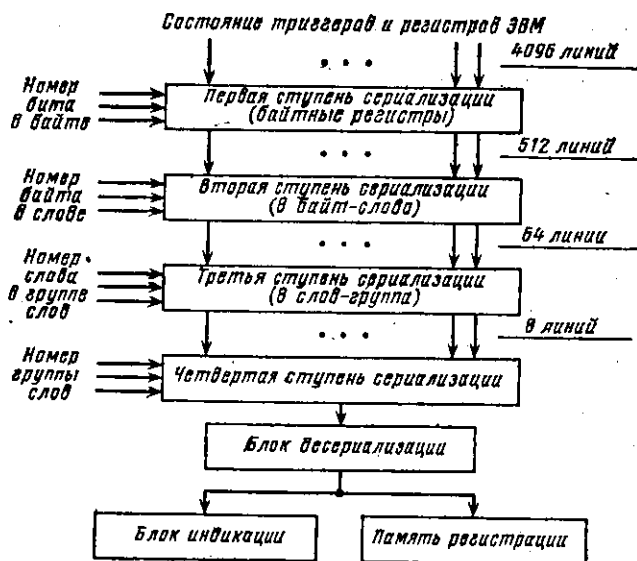


Рис. 8.35. Структура системы сериализации данных о состоянии ЭВМ

состоянии элементов машины общим объемом 512 9-битных байт.

Система сериализации данных построена на основе использования для опроса состояния элементов аппаратуры и преобразования данных в последовательный код — мультиплексоров, рассредоточенных по всему контролируемому этой системой оборудованию ЭВМ. Система сериализации имеет иерархическую структуру (рис. 8.35), образованную четырьмя ступенями преобразования кода. Первая ступень имеет 4096 входных линий, по которым поступают сигналы состояния соответствующих элементов оборудования. На выходе четвертой ступени имеется одна выходная линия, по которой в пультовое оборудование поступает последовательный код, несущий информацию о состоянии машины. В каждой ступени с помощью мультиплексоров, управляемых синхросигналами, производится восьмикратное уплотнение — восьмикратное уменьшение числа выходных линий по отношению к числу входных за счет последовательной во времени передачи информации [3].

Оборудование пульта управления включает в себя блок десериализации, развертывающий входной последовательный код в последовательность байт, передаваемых с помощью коммутаторов цепей в регистры индикации и память пульта управления.

Система сериализации управляется блоком синхросигналов пульта управления, вырабатывающим серию из 13 синхросигналов, имеющих разные частоты повторения. Сигналы этой серии (сигналы бита

сериализации) СИБС1, СИБС2, СИБС4 с двоичными весами 1, 2, 4 и сигнал сериализации контрольного бита СИБСК управляют в первой ступени параллельным во времени развертыванием в последовательный код каждого байта данных о состоянии машины. Девять других синхросигналов с двоичными весами 1, 2, ..., 256, называемых сигналами адреса сериализации СИАС1—СИАС256, управляют соответствующими степенями схемы, указывая номер байта, подвергающегося сериализации. Вторая ступень производит сериализацию групп из 8 байт, третья — из 64 байт. Четвертая ступень содержит один мультиплексор, на выходе которого получается последовательность бит, характеризующих состояние машины.

Блок десериализации восстанавливает байтную структуру данных и передает последовательно во времени восстановленные байты на индикаторы пульта и в буферную память пульта управления; при этом коммутаторы, осуществляющие передачу байт на соответствующие индикаторы, управляются дешифраторами, на входы которых поступают сигналы СИАС-1—СИАС-256, фиксирующие положение текущего байта среди 512 байт данных о состоянии ЭВМ. Блок десериализации, формируя очередной байт, снабжает его дополнительным десятым, контрольным (по четности) битом, используемым при контроле считываемых из памяти пульта управления (ППУ) данных. Память пульта управления может хранить 512 10-битных байт.

В блок десериализации сериализованные биты данных поступают с интервалом 480 нс (соответствует четырем машинным тактам ЭВМ ЕС-1045). Время сериализации байта, равное времени сериализации 10 бит, составляет 4,8 мкс. Продолжительность цикла сериализации передаваемых состояний аппаратурных средств ЭВМ составляет 2,5 мс. Следовательно, каждый контролируемой системой двоичный элемент аппаратуры опрашивается с частотой 400 Гц.

В ЭВМ ЕС-1045 используется как прямая, так и косвенная сериализация. При прямой сериализации информация о состоянии оборудования поступает непосредственно с элементов аппаратуры на входы первой ступени сериализации, а при косвенной данные о состоянии оборудования передаются на входы первой ступени после предварительного группирования, которое выполняется коммутаторами, принадлежащими основному оборудованию ЭВМ и участвующими в нормальной работе ЭВМ. В нормальном режиме работы эти коммутаторы управляются управляющими сигналами микрооперации. При блокировании поступления основных синхросигналов ЭВМ коммутаторы управляются специальными синхросигналами, вырабатываемыми из серии СИАС1 — СИАС256. Косвенная сериализация применяется для экономии оборудования.

Индикация данных, доставляемых на пульт при прямой сериализации, возможна при работе машины, а при косвенной — только при блокировании поступления синхросигналов ЭВМ.

Метод микродиагностирования благодаря своим достоинствам (небольшое диагностическое ядро, большая глубина поиска дефекта, возможность использования при наладке ЭВМ на заводе-изготовителе) получил широкое распространение. В частности, все ЭВМ ЕС II очереди используют этот метод диагностирования.

В качестве примера реализации метода микродиагностирования рассмотрим систему автоматического диагностирования ЭВМ ЕС-1045 [80].

Диагностическое ядро или средства тестового диагностирования ЭВМ ЕС-1045, включает в себя малогабаритный пультовый накопитель (ПН) и сервисный адаптер (СА), управляющий чтением с ПН, накоплением диагностической информации и выполнением диагностических операций. Опрос состояния ЭВМ выполняется по системе сериализации (см. § 8.10).

Сервисный адаптер осуществляет поиск нужных тестов на магнитной ленте. При этом СА преобразует поступающий с ПН последовательный код в параллельный код данных, передаваемых в ЭВМ.

Структурная схема диагностического ядра ЭВМ ЕС-1045 приведена на рис. 8.36.

Данные, считанные с ПН, накапливаются в регистре сдвига РгСдв и из него поступают либо в регистр диагностики РгДг, либо на дешифратор диагностических операций.

Тесты микродиагностики записаны на магнитной ленте в виде зон. Начало зоны определяется маркером зоны, за которым следует адрес зоны. Адрес зоны считывается с ПН, накапливается в РгСдв и затем поступает в РгДг, сравнивается с содержимым регистра зоны (РгЗн). Если заданная зона найдена, ее информация считывается, накапливается в РгСдв и из него поступает либо в РгДг, либо на дешифратор диагностических операций. Информация считывается до появления кода операции «Конец зоны», за которым на ленте записан адрес следующей зоны с тестами микродиагностики.

Если адрес зоны, принятый в РгДг, больше адреса зоны в РгЗн, то лента продвигается вперед до тех пор, пока не будет найдена заданная зона. Если адрес зоны, принятый в РгДг, меньше адреса зоны в РгЗн, то выполняется реверс магнитной ленты, продвижение назад на четыре и повторное включение движения вперед. Эта процедура повторяет-

ся до тех пор, пока не будет найдена заданная в РгЗн зона.

Микродиагностика ЭВМ ЕС-1045 состоит из двух частей: статической и динамической.

Первая часть микродиагностики, *статическая*, осуществляется путем загрузки микрокоманды в регистр микрокоманд, ее выполнения и сравнения результата с эталонным. Эта часть микродиагностики называется статической, так как она выполняется со скоростью, определяемой скоростью ввода информации с ПН, и, как отмечалось, в большей степени имитирует «быстрый» тактовый режим, чем работу ЭВМ с реальным быстроедействием. Статическая микродиагностика обеспечивает проверку процессора, каналов, управления памятью, частично самих памяти, аппаратуры пульта и пультовой печатающей машинки.

При выполнении статической микродиагностики можно выделить следующие шаги:

загрузка микрокоманды из ПН в регистр микрокоманд процессора и ее выполнение. Тестовая процедура состоит из выполнения некоторой последовательности микрокоманд, проверяющей соответствующий узел аппаратуры;

считывание с ПН адреса (номера) байта в сериализованной последовательности данных и занесение байта в регистр данных пульта;

сравнение результатов тестовой последовательности микрокоманд с эталоном;

останов при несовпадении и индикации на пульте управления кода останова и состояния ЭВМ.

По коду останова и информации о состоянии ЭВМ на пульте персонал в распечатках микродиагностики находит название принявшего неправильное состояние триггера, список неисправных ТЭЗ (для замены) и рекомендации по поиску конкретной причины неисправности. Пример распечаток микродиагностики ЭВМ ЕС-1045 дан на рис. 8.37.

Рассмотрим более подробно работу средств диагностирования при выполнении статической микродиагностики.

Загрузка микрокоманды в регистр микрокоманд ЭВМ ЕС-1045 производится следующим образом: если байт данных, накопленный в РгСдв, содержит 1 в седьмом бите ($\text{РгСдв}(7) = 1$), то он представляет собой код диагностической операции. В противном случае ($\text{РгСдв}(7) = 0$) байт данных передается в РгДг. В РгДг передаются биты РгСдв (0—6). Регистр диагностики рассчитан на прием трех 7-битных байт (21 бит). Для загрузки микрокоманд имеются диагностические операции «Установка групп 1—4». С их помощью содержимое РгДг заполняет различные поля ре-

Зона ОВО*	Тест ВВЗ	УОБ и регистры (проверяемая аппаратура)					
Адрес	Микрокоманда	Операция	Биты переходов		Поле константы	Микрооперации (мнемоника)	Комментарий
IFF3 IE00 IE04	800000000000000100 00800280000000000100 81000540000000000100	M*** M***	0 0 0	0 0 0	00 00 00	Блокировка ошибок ; MISC=E ; AB=E57 СБАР 100 00С	Диагностическая операция сравнения байтного регистра с адресом 00С (регистр PU) с эталонным значением 100 (первая цифра — бит четности)
02**							
	Ошибка регистрации: PU ≠ 100 (01С—2В20) 005, (01С—1Е04) 020, (01С—2С18) 422						Список подозреваемых на неисправность ТЭЗ
IE08	818005808000000100		0	0	00	, := EU ; CB=E57 СБАР 100 00С	
03**		M***					
	Ошибка регистрации: неверная передача PU: = поле константы (01С—2В20) 005, (01С—1ЕТ6) 020, (01С—1Е05) 020						
IE0С	82000000008000031FF	M***	0	0	FF	FGW, := EU СБАР IFF 00С	Проверка передачи PU=поле константы
04**							
	Ошибка регистрации: PU ≠ IFF (01С—2В20) 005, (01С—1Е06) 020						

* Номер зоны на кассете; ** код останова; *** признак диагностической операции.

Рис. 8.37. Пример распечатки микродиагностики ЭВМ ЕС-1045

гистра микрокоманд. При наличии 1 в бите 4 кода операции «Установка групп» накопленная микрокоманда выполняется. Выполнение каждой микрокоманды сопровождается сравнением последних бит адреса микрокоманды с эталонными значениями этих бит, заданных битами 5 и 6 кода операции «Установка групп». Так проверяется выполнение всех микроопераций анализа, которые изменяют состояние последних бит адреса микрокоманды. Несовпадение этих значений вызывает установку триггера ошибки АВ перехода (ТгОшАВ), останов диагностирования и индикацию на пульте ЭВМ ЕС-1045 причины останова.

Анализ состояния аппаратных средств ЭВМ после выполнения проверяемых микрокоманд производится сервисным адаптером путем опроса состояния и сравнения его с эталоном. Этой цели служат диагностические операции «Регистрация», «Проверка регистрации», «Сравнение байта» и «Сравнение бита». Операция «Регистрация» обеспечивает запись состояния ЭВМ в память пульта управления для последующего сравнения его с эталонным. Операция «Проверка регистрации» задает адрес проверяемого регистра для выделения его состояния из общего потока сериализованных данных о состоянии ЭВМ и приема в регистр данных пульта управления.

Операции «Сравнение бита» и «Сравнение байта» обычно следуют за операцией «Проверка регистрации». Эти операции разрешают сравнение информации о состоянии проверяемых средств, находящихся в регистре данных пульта управления, с эталоном, находящимся в РгДг. При несовпадении устанавливается триггер ошибки регистрации (ТгОшР), процесс диагностирования заканчивается и на ПУ ЭВМ индицируется причина останова.

Для проверки переключателей и кнопок пульта управления имеется операция «Имитация кнопок». Эта операция обеспечивает передачу содержимого РгДг в триггеры кнопок и переключателей, вызывая тем самым схемную последовательность кнопочных операций, аналогичную той, которая вызывается путем нажатия кнопок или установки переключателей.

В начале каждого теста микродиагностики выполняются операция «Блокировка ошибок», которая устанавливает триггер блокировки ошибок, блокирующий реакцию ЭВМ на сигналы от схем контроля во время микродиагностики, а также операция «Диагностический сброс», выполняющая сброс аппаратуры.

Операция «Сигнал» включает звуковую сигнализацию и используется для оповещения оператора об окончании тестов микродиагностики, расположенных на одной кассете.

Поскольку во время статической микродиагностики темп выполнения микрокоманд отличается от реальной скорости машины, могут остаться необнаруженными некоторые дефекты, проявляющие себя в динамическом режиме.

Для обнаружения более сложных неисправностей, связанных с временной диаграммой, используется *динамическая микродиагностика*.

Динамической микродиагностикой охвачена вся аппаратура ЭВМ, включая все памяти машины (кроме внешних).

Динамическая микродиагностика выполняется с помощью управляющей памяти (УП) процессора, в которую с пульта накопителя предварительно загружаются диагностические микропрограммы. Функционирование ЭВМ под управлением диагностических микропрограмм происходит в соответствии с рабочей тактовой частотой машины. Загрузку диагностических микропрограмм в управляющую память производит сервисный адаптер. Результаты диагностической проверки (состояния ЭВМ) передаются в пульттовую аппаратуру, и сервисный адаптер сравнивает их с эталоном.

Для загрузки микропрограмм в УП используются диагностические операции «Повторение микрокоманды» и «Вход в микропрограмму». С их помощью выполняется переход к микропрограммному загрузчику УП, который принимает байты данных с ПН, формирует микрокоманду и помещает ее в УП. По окончании ввода тестовой микропрограммы в УП управление передается этой микропрограмме. По ее завершении результаты анализируются СА, и при несовпадении с эталоном происходит останов с индикацией.

Неисправность определяется по номеру кассеты, номеру зоны и коду останова.

Наличие двух видов микродиагностики — статической, имитирующей режим с низкой тактовой частотой, и динамической, выполняющейся с реальной машинной скоростью, — позволяет последовательно выявлять неисправности, сначала простые, типа постоянных отказов, а затем и более сложные, связанные с изменением параметров схем, затягиванием фронтов сигналов, отсутствием согласующих сопротивлений, напряженной временной диаграммой. Совокупность этих средств позволяет практически гарантировать работоспособность центральной части ЭВМ после прохождения микродиагностики.

Комплект микродиагностики рассчитан на максимально возможный состав оборудования ЭВМ. Тесты микродиагностики сами проверяют состав действительно включенного оборудования. Например, признак недействительного адреса используется для определения объема наличной оперативной памяти. Для определения наличия в системе канала под определенным номером и его типа тесты опрашивают средства канала, хранящие эти данные. При отсутствии устройств тесты, предназначенные для их проверки, игнорируются.

Для обнаружения некоторых сложных неисправностей, которые могут проявиться только при работе машины, в ЭВМ ЕС-1045 имеется комплект программных автономных диагностических тестов центральной части ЭВМ.

Проверку ЭВМ ЕС-1045 завершают тесты функциональной диагностики. На этой стадии диагностирования проверяется правильность выполнения всех команд, исправность периферийных устройств, блоков управления периферийными устройствами, параллельная работа периферийных устройств.

Комплект программ функциональной диагностики содержит базовый тест, тест-монитор ДМ и комплект тест-секций.

Функциональная диагностика начинается с выполнения базового теста, который под управлением сервисного адаптера загружается в оперативную память с пультавого накопителя. С помощью базового теста проверяется правильность выполнения машиной основных команд, команд ввода и связи машины с запоминающими устройствами на магнитной ленте и пишущей машинкой. Затем производится проверка остальных периферийных устройств с помощью входящих в комплект программ технического обслуживания тест-секций, выполняемых под управлением тест-монитора ДМ.

Поскольку глубина поиска дефекта при микродиагностике составляет 2 — 3 ТЭЗ, для быстрого определения неисправного ТЭЗ из группы подозреваемых и локализации в нем неисправной ИС в ЭВМ ЕС-1045 предусмотрено дополнительное средство — интегрированное устройство диагностики ТЭЗ (автотестер), выполненных на ИС серии 500. Структурная схема этого устройства приведена на рис. 8.38. Диагностические тесты ТЭЗ записаны на кассетной магнитной ленте и считываются с пультавого накопителя МЛ-45. Прогон тестов одного ТЭЗ выполняется за несколько секунд.

Проверяемый ТЭЗ устанавливается в специальный разъем, с каждым сигнальным контактом (всего 96) которого со-

эталонами. Эталонные коды, так же как и вся служебная информация, считываются с магнитной ленты, т. е. скорость тестирования определяется скоростью чтения магнитной ленты. При несовпадении состояния выходов с эталонными тестирование прекращается и на пульте управления ЭВМ ЕС-1045 индицируется порядковый номер теста, на котором обнаружена неисправность.

После этого визуально (по индикации регистра автотестера) определяется номер контакта, на котором обнаружено несоответствие. По номеру теста и контакта в диагностической таблице находится список возможных неисправностей ИС. Для подтверждения локализации состояние контактов ИС проверяется с помощью логического пробника ПЭТ-1 и сравнивается с эталонным, приведенным в таблице состояний ИС.

Таблицы проверки, диагностические таблицы, таблицы состояний и магнитная лента ТЭЗ получаютс^я с помощью системы автоматизированного проектирования ТЭЗ.

Время локализации неисправной ИС с помощью автотестера составляет несколько минут. Быстрая локализация неисправностей в ТЭЗ с точностью до ИС позволяет сократить время восстановления ЭВМ и повысить ее ремонтпригодность.

8.12. ОСОБЕННОСТИ ДИАГНОСТИРОВАНИЯ КАНАЛОВ ВВОДА-ВЫВОДА

Каналы представляют собой специализированные процессоры, предназначенные для управления вводом-выводом. Подключение периферийных устройств (ПУ) к каналам выполняется с помощью стандартного интерфейса ввода-вывода. В интерфейсе ввода-вывода выделяют две группы шин:

шины канала, состоящие из информационных шин прямой передачи (из канала в ПУ) и шин управления. Информационные шины, по которым передаются выводимый байт информации, приказ (код операции) ПУ и номер (адрес) ПУ, обозначаются ШИН-К. Шины управления, по которым в ПУ передаются управляющие сигналы, обозначаются СИГ-К;

шины абонента, состоящие из информационных шин обратной передачи (из ПУ в канал) и шин управления. Информационные шины, по которым передаются вводимый байт информации, байт состояния и номер (адрес) ПУ, обозначаются ШИН-А. Шины управления, по которым в канал передаются управляющие сигналы, обозначаются СИГ-А.

Для диагностирования каналов необходимо имитировать сигналы на шинах ШИН-А и СИГ-А. Существуют следующие способы имитации сигналов абонента:

микрокомандный; с помощью регистра имитации интерфейса (РИИ); с помощью другого канала; с помощью встроенного в каналы блока управления периферийным устройством (УПУ).

Схемы соединений при каждом способе имитации сигналов абонента приведены на рис. 8.39, а—г.

Микрокомандный способ применяется в основном при микродиагностировании. Имитация СИГ-А выполняется с помощью специальной микрооперации и поля константы микрокоманды. Имитация ШИН-А выполняется путем засылки в буферный регистр ШИН-А требуемого кода с помощью соответствующей микрооперации ЭВМ.

Способ имитации сигналов абонента с помощью регистра имитации интерфейса универсален и может быть использован как при микро-

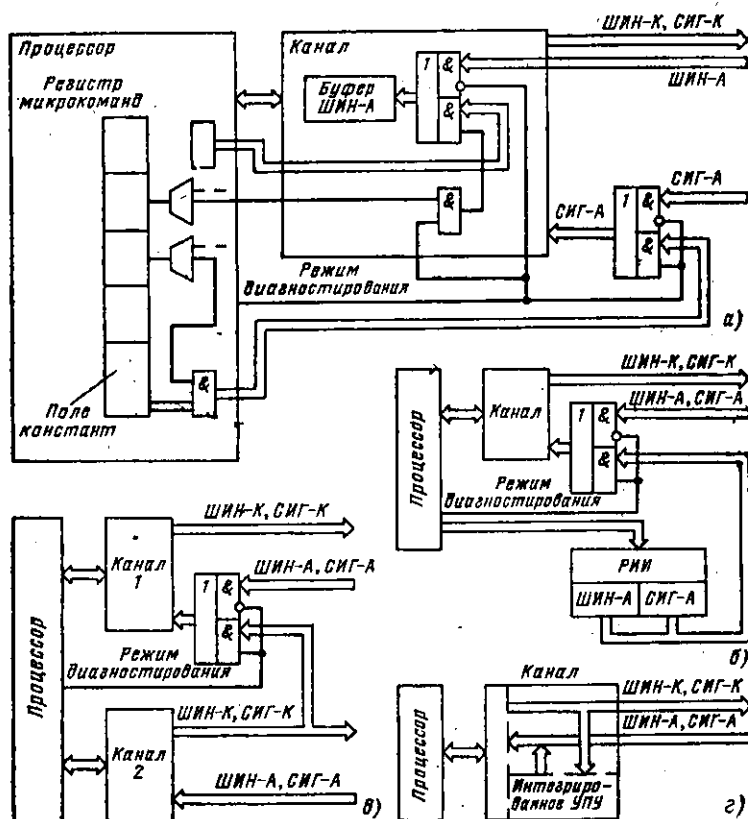


Рис. 8.39. Способы имитации сигналов абонента при диагностировании каналов ввода-вывода:

а — микрокомандный; б — с помощью регистра имитации интерфейса; в — с помощью другого канала; г — с помощью встроенного в каналы имитатора УПУ

ропрограммном, так и при схемном управлении каналов. Загрузка РИИ из процессора позволяет имитировать любые сочетания сигналов абонента.

Имитация сигналов абонента с помощью другого канала позволяет интерпретировать сигналы ШИН-К и СИГ-К одного канала как сигналы ШИН-А и СИГ-А для другого канала.

В некоторых ЭВМ для диагностирования каналов ввода-вывода используется встроенный имитатор УПУ, представляющий собой первое устройство, подключенное к интерфейсу ввода-вывода.

Встроенный имитатор снабжается загружаемой управляющей памятью, в которую из процессора может загружаться произвольная микропрограмма, в том числе и микропрограмма, обеспечивающая работу имитатора УПУ при диагностировании каналов.

Имитатор УПУ способен также имитировать нормальный режим передачи с максимальной скоростью и другие режимы работы каналов.

8.13. ДИАГНОСТИРОВАНИЕ ОПЕРАТИВНЫХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ

Для диагностирования оперативных запоминающих устройств (ОЗУ) используются следующие типы тестов [8].

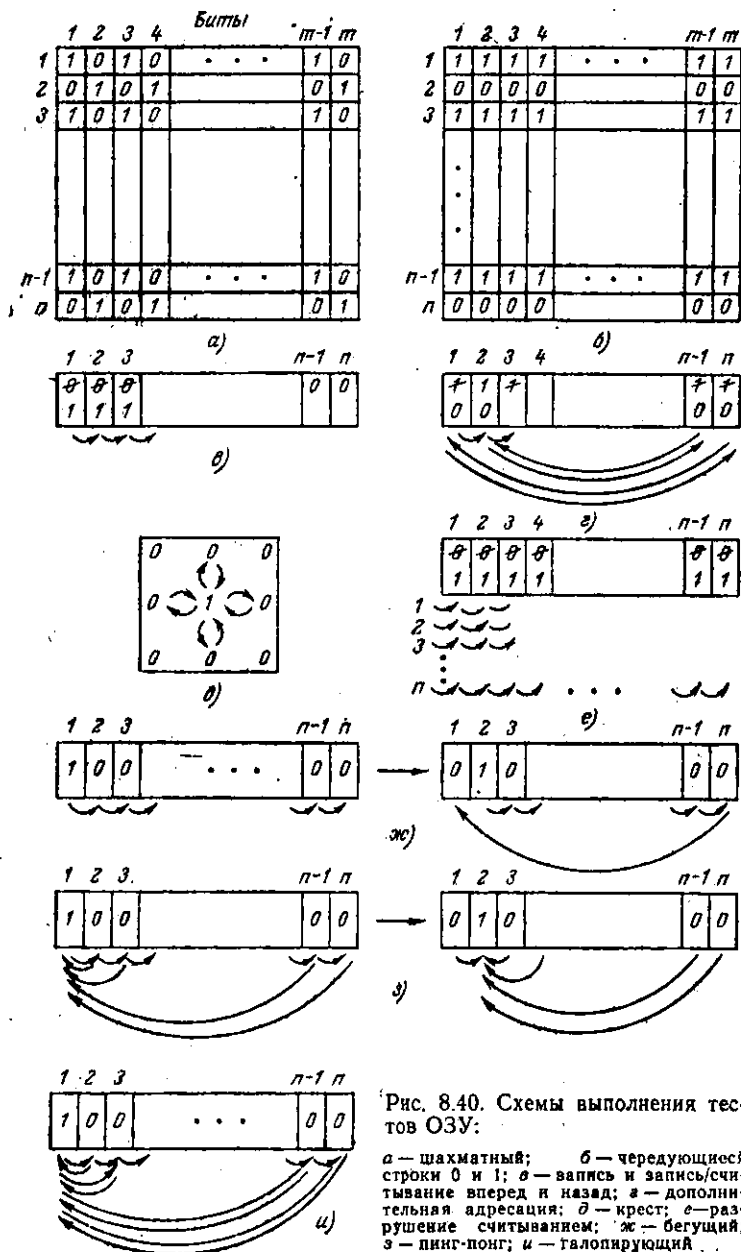
Все нули (все единицы). Во все ячейки ОЗУ производится запись нулей (единиц), после чего производится последовательное считывание и проверка этой информации.

Адресный. В каждую ячейку ОЗУ записывается код собственного адреса, затем производится последовательное считывание и проверка этой информации. Адресный тест обеспечивает проверку адресных дешифраторов ОЗУ.

Шахматный. В ОЗУ записывается информация, имеющая шахматное распределение (рис. 8.40), затем производится последовательное считывание и проверка этой информации. Шахматный тест используется для проверки взаимовлияния ячеек, содержащих информацию, записанную в обратном коде.

Сканирующий. Производится запись нулей (единиц) во все ячейки ОЗУ, затем выполняется последовательное считывание и проверка. После этого во все ячейки записываются единицы (нули) и процесс повторяется. Сканирующий тест используется для проверки ОЗУ в условиях максимальной статической помехи, вызванной суммарным током утечки всех ячеек ОЗУ, находящихся в одном состоянии.

Чередующиеся строки 0 и 1. В смежные ячейки ОЗУ записывается информация в обратном коде (рис. 8.40), затем производится последовательное считывание и проверка этой информации. Этот тест используется для проверки взаимовлияния адресных шин по строкам.



Чередующиеся столбцы 0 и 1. В ячейки ОЗУ записывается такая информация, чтобы смежные столбцы ОЗУ содержали информацию в обратном коде, затем производится последовательное считывание и проверка этой информации. Этот тест используется для проверки взаимовлияния адресных шин по столбцам.

Запись и запись/считывание вперед и назад. По всем адресам ОЗУ записываются нули, затем производится последовательное считывание и проверка этой информации. После проверки каждой очередной ячейки в нее записывается информация в обратном коде (единицы). После проверки последней ячейки и записи в нее единиц процедура повторяется от старшего адреса к младшему с чтением единиц, их проверкой и записью нулей. Этот тест используется для проверки взаимовлияния соседних ячеек при смене в них информации. Схема выполнения теста для одного бита ОЗУ приведена на рис. 8.40, а.

Марширующий. Во все ячейки ОЗУ записываются единицы, затем производится последовательное считывание информации с проверкой и заменой ее на нули. После обращения к последнему адресу процедура повторяется с данными в обратном коде, т. е. последовательное считывание нулей, начиная с первой ячейки, с проверкой и заменой ее на единицы. После обращения к последнему адресу процедура повторяется с данными в обратном коде, т. е. с нулями, и в обратном направлении — от последней ячейки к первой. После обращения к первой ячейке процедура повторяется. Считываются нули и на их место записываются единицы. После обращения к последнему адресу выполняется чтение с проверкой единиц всех ячеек ОЗУ — от первой до последней. Этот тест является модификацией теста «запись и запись/считывание вперед и назад».

Дополнительная адресация. Во все ячейки ОЗУ записывается фоновый набор единиц (нулей), затем производится считывание ячейки, начиная с первой, с последующей проверкой и записью в нее противоположной информации. Каждое второе обращение выполняется по адресу, код которого является дополнением к предыдущему. Схема выполнения теста для одного бита ОЗУ приведена на рис. 8.40, б.

Этот тест предназначен для проверки адресных цепей, информация которых в этом тесте подвергается максимальному изменению.

Долбление. Во все ячейки ОЗУ записывается тестовая информация, после чего производится многократное считывание

вание по каждому адресу с последующей проверкой по всем адресам. Процедура повторяется при замене информации в каждой ячейке на информацию в обратном коде. Этот тест предназначен для проверки способности ячеек выдерживать многократные обращения по считыванию.

Крест. В каждую ячейку ОЗУ записывается тестовое слово (единицы или нули), а в каждую из четырех соседних ячеек — фоновое слово (нули или единицы соответственно). Затем информация в соседних ячейках изменяется и проверяется влияние этого изменения на проверяемую ячейку.

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

Схема выполнения теста «крест» для одного бита ОЗУ приведена на рис. 8.40, д.

Разрушение считыванием. Во все ячейки ОЗУ записывается, считывается и проверяется тестовое слово (все единицы). Выполняется приращение адреса и тестовое слово записывается во вторую ячейку. После этого информация из первой и второй ячеек считывается и проверяется. Процедура продолжается до тех пор, пока во все ячейки ОЗУ не будет записано тестовое слово. К нулевой ячейке производится n обращений, к первой — $n - 1$, к последней — одно. Схема выполнения теста для одного бита ОЗУ приведена на рис. 8.40, е.

Тест «разрушение считыванием» используется для проверки взаимовлияния ячеек ОЗУ при записи в них одной и той же информации.

Бегущий. В первую ячейку ОЗУ записываются единицы (нули), а во все остальные — фоновые нули (единицы). Затем все адреса последовательно считываются с проверкой; последней считывается первая ячейка с последующей записью в нее нулей (единиц). Последовательность операций повторяется для второй ячейки, третьей и т. д. вплоть до последней.

Схема выполнения теста для одного бита ОЗУ показана на рис. 8.40, ж. Тест «бегущий» предназначен для обнаружения сбоев в ОЗУ, вызванных переходными процессами в разрядных цепях, так как перемещение 1 на фоне 0 (или наоборот) создает наихудшие условия для усилителей считывания.

Пинг-понг. В первую ячейку ОЗУ записываются единицы, а во все остальные — нули. Затем последовательно считываются и проверяются ячейки 2,1, затем 3,1; 4,1 и т. д.,

пока все пары переходов, включающие ячейку 1, не будут проверены. После этого в ячейку 1 записываются нули, а во вторую — единицы. В той же последовательности операции повторяются для ячейки 2 и т. д. Цикл повторяется для инверсной информации.

Схема выполнения теста для одного бита ОЗУ показана на рис. 8.40, з. С помощью этого теста проверяется функционирование накопительной части ОЗУ, дешифратора, а также влияние записи на сохранность информации.

Галопирующий. В первую ячейку ОЗУ записываются единицы, а в остальные — нули. Затем последовательно считываются и проверяются ячейки 2,1,2, затем 3,1,3 и т. д., пока все пары переходов, включая ячейку 1, не будут проверены. После этого в ячейку 1 записываются нули и информация считывается. Последовательность операций повторяется для ячейки 2,3 и т. д. вплоть до последней. Схема выполнения теста для одного бита ОЗУ показана на рис. 8.40, и. По эффективности тест эквивалентен тесту «пинг-понг».

По характеру зависимости длительности выполнения тестов от числа ячеек ОЗУ приведенные выше тесты можно разделить на две группы: пропорциональные n ; пропорциональные n^2 , где n — емкость ОЗУ.

Таблица 8.1

Тест	Обнаруживаемые отказы					
	в матрице			в дешифраторе		
	Отсутствие записи	Ложная запись	Ложное считывание	Отсутствие выборки	Многократная выборка	Неоднозначная выборка
Сканирующий	—	0	0	0	0	0
Шахматный	+	—	0	—	0	0
Запись и запись/считывание вперед и назад	+	+	—	—	—	—
Марширующий	+	+	—	—	—	—
Дополнительная адресация	+	—	—	+	—	+
Крест	+	+	—	—	—	0
Бегущий	+	+	+	—	—	—
Пинг-понг	+	+	+	+	+	+
Галопирующий	+	+	+	+	+	+

Примечание. + — обнаруживает; — — не обнаруживает; 0 — неэффективное обнаружение.

Ниже приведены длительности тестов, а в табл. 8.1—их обнаруживающая способность.

Тест	Длительность
Все нули (единицы)	$2n$
Адресный	$2n$
Шахматный	$4n$
Сканирующий	$4n$
Чередующиеся строки 0 и 1	$4n$
Чередующиеся столбцы 0 и 1	$4n$
Запись и запись / считывание вперед и назад	$5n$
Марширующий	$10n$
Дополнительная адресация	$10n$
Крест	$128n$
Разрушение считыванием	n^2
Бегущий	$2(n^2 + 2n)$
Пинг-понг	$2(2n^2 + 2n)$
Галопирующий	$2(3n^2 + 3n)$

8.14. ОСОБЕННОСТИ ОБСЛУЖИВАНИЯ И ДИАГНОСТИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ И МИКРО-ЭВМ. СИГНАТУРНЫЙ АНАЛИЗ

В ЭВМ общего назначения улучшение обслуживаемости достигается, в частности, путем использования систем автоматического диагностирования. При этом, как следует из § 8.4, обычно автоматическое диагностирование производится по методу «раскрутки» с диагностическим ядром, представляющим собой достаточно сложную аппаратуру (например, пультовый накопитель и сервисный адаптер).

Во многих случаях для микропроцессорной аппаратуры и микро-ЭВМ не удастся применить автоматическое диагностирование на основе встроенных или внешних тестовых средств диагностирования или их комбинаций из-за ее сложности и высокой стоимости. Тогда следует позаботиться об облегчении ручного обслуживания, в том числе ручного поиска неисправностей, с тем чтобы обслуживание не требовало персонала с высокой квалификацией.

Предположим, что имеется устройство с несколькими входами, на которые поступают двоичные входные сигналы, и несколькими выходами, с которых снимаются двоичные выходные сигналы.

Для проверки работоспособности устройства на его входы необходимо подать тестовую последовательность комбинаций (векторов) входных сигналов и сравнить получаемые значения выходных сигналов (выходных двоичных векторов) со значениями, указанными в документации по обслуживанию.

Существенной проблемой при ручном обслуживании является сжатие информации о правильных (эталонных) и наблюдаемых при контроле реакциях аппаратуры на тестовые последовательности.

Необходимость сжатия двоичных векторов произвольной длины диктуется тем, что современная микропроцессорная и микромашинная техника использует достаточно сложные функциональные узлы, которые требуют для диагностирования большого числа тестовых векторов, с тем чтобы обеспечивалась проверка всех возможных состояний устройства. Длина выходного двоичного вектора равна числу входных тестовых векторов [29].

Сжатие информации облегчает фиксирование правильных (эталонных) реакций в технической документации и восприятие оператором результатов проверки.

Существует несколько способов сжатия информации о реакциях аппаратуры на тестовые последовательности. Так, может подсчитываться и указываться в документации правильное число логических переключений выходного сигнала при подаче тестовой последовательности. Другой способ состоит в подсчете и указании в технической документации контрольной суммы выходных векторов, порожденных совокупностью тестов. Недостаток этих методов состоит в том, что при диагностическом (проверочном) подсчете числа переключений или контрольной суммы выходных векторов контрольная аппаратура, реализующая эти подсчеты, должна иметь цепи распространения переносов при суммировании, что замедляет процесс проверки и усложняет контрольную аппаратуру.

Перспективным направлением в повышении обслуживаемости микропроцессорной аппаратуры и микро-ЭВМ является «сигнатурный анализ», предполагающий использование циклических избыточных кодов для сжатия длинных двоичных кодов — реакций аппаратуры на тестовые последовательности в короткий, обычно 4-, 5-разрядный шестнадцатиричный код, который просто индицируется и сравнивается с указанным в документации для каждой контролируемой точки контрольным кодом («сигнатурой») [10, 41, 51].

Для сжатия длинных двоичных последовательностей и получения кодов сигнатур используется сигнатурный анализатор, основу которого составляет сдвиговой регистр с внутренними обратными связями, замыкаемыми через сумматор по модулю 2, на вход которого также поступает

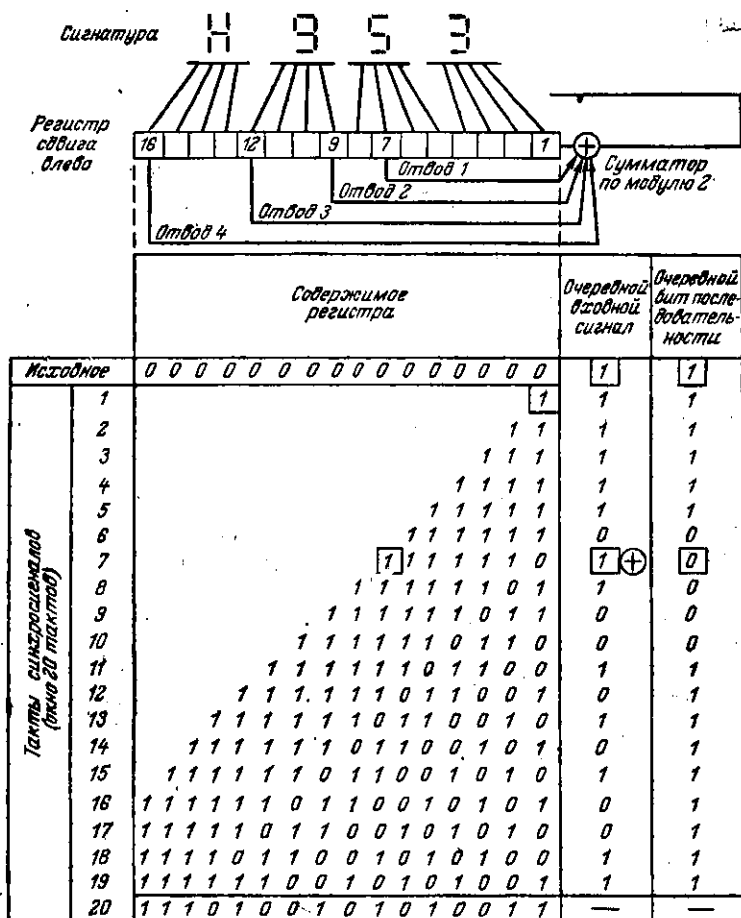


Рис. 8.41. Принцип действия сигнатурного анализатора

последовательность бит, снимаемая с контролируемой точки схемы (рис. 8.41); сдвиговый регистр снабжается индикаторами, указывающими содержимое регистра в шестнадцатичном коде.

Сигнатурный анализатор основывается на следующем принципе сжатия данных: двоичная последовательность x в виде информационного полинома $G(x)$ поступает с выхода проверяемой схемы на сдвиговый регистр и делится в виде полинома $x^k G(x)$ (где k — количество разрядов сдви-

гового регистра) на порождающий полином $P(x)$ степени k . Деление на порождающий полином $P(x)$ реализуется с помощью сдвигового регистра с обратными связями. Результатом деления является остаток $R(x)$, получающийся в сдвиговом регистре после приема входной последовательности. Математически этот процесс описывается формулой (см. § 5.3)

$$x^k G(x) = Q(x) P(x) \oplus R(x),$$

где $Q(x)$ — частное; $R(x)$ — остаток; $P(x)$ — порождающий полином степени k , а $G(x)$ — информационный полином, соответствующий входной двоичной последовательности x .

При прохождении последовательности x через сдвиговый регистр $R(x)$ изменяется до тех пор, пока не закончится вся последовательность x . Конечное выражение $R(x)$ является сигнатурой.

Для проверки устройства щуп сигнатурного анализатора присоединяется к контролируемому выходу схемы. На входы контролируемой схемы поступает последовательность параллельных кодов тестов (векторов-стимулов). Каждому вектору-стимулу соответствует 1 бит в выходной последовательности (выходном векторе), снимаемой с контролируемого выхода.

Подача последовательности тестовых векторов, снятие с контролируемой точки последовательности бит на вход сумматора по модулю 2 и работа сдвигового регистра синхронизируются одной и той же серией синхросигналов.

Процесс получения сигнатуры предполагает задание некоторого фиксированного интервала времени — «окна», соответствующего определенному числу синхросигналов, а следовательно, и такому же числу векторов-стимулов и вызываемых ими бит в выходной последовательности, поступающей на вход сигнатурного анализатора. После закрытия «окна» (остановки сигнатурного анализатора) на представленном на рис. 8.41 сдвиговом регистре остается 16-битный остаток, который индицируется на семиэлементных индикаторах в виде четырех шестнадцатиричных цифр. На рис. 8.42 приведена временная диаграмма работы сигнатурного анализатора. Полученная на индикаторах сигнатура сравнивается с эталонной, указанной в документации для данной контролируемой точки.

Эталонные сигнатуры могут быть получены путем проверки с помощью сигнатурного анализатора заведомо исправных схем или расчетным путем на ЭВМ.

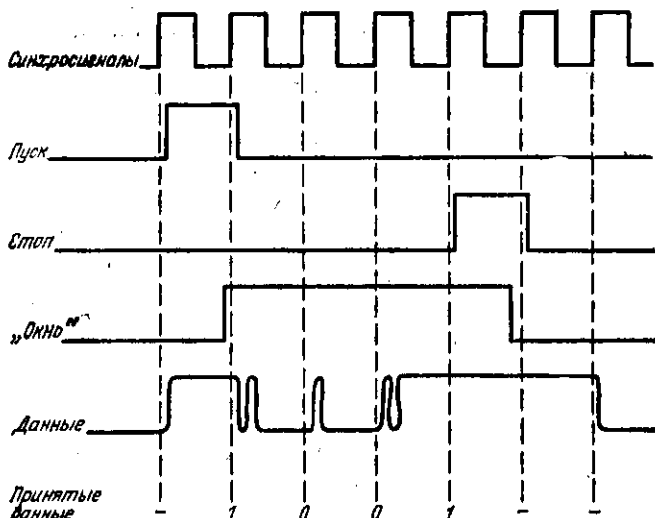


Рис. 8.42. Временная диаграмма процесса сигнатурного контроля

На рис. 8.41 показан процесс формирования сигнатуры для входной последовательности 111111000001111111, соответствующей окну длительностью 20 синхросигналов. На протяжении первых семи тактов по цепям обратной связи поступают 0 и в регистр вдвигаются без каких-либо изменений первые 7 бит входной последовательности. После седьмого такта по обратной цепи от первого отвода (седьмой разряд регистра) на сумматор по модулю 2 поступает 1, которая в восьмом такте заносится в первый разряд регистра (в восьмом разряде входной последовательности стоит 0). Аналогичным образом работает анализатор в последующие такты.

После закрытия «окна» (после 20 тактов синхросигналов) в регистре образуется 16-битный остаток 1110, 1001, 0101, 0011, соответствующий сигнатуре H953.

Метод сигнатурного анализа требует небольшого (около 3—4 %) дополнительного оборудования и достаточно простого универсального переносного сигнатурного анализатора. Тест-программы проверки аппаратуры могут быть записаны в БИС программируемых постоянных запоминающих устройств, размещаемых в самом проверяемом оборудовании.

При выполнении диагностической процедуры оператор последовательно просматривает в тестовом режиме сигнала

туры на выходах устройства и при обнаружении расхождения с указанными в документации сигнатурами переходит к просмотру сигнатур в точках неисправной цепи, двигаясь от выхода к входу, пока не будет найден неисправный элемент. При использовании сигнатурного анализа поиск неисправностей в микропроцессорной аппаратуре становится похожим на обслуживание телевизора, когда мастер, ремонтирующий его, с помощью тестера и шупа сравнивает сигналы в данной цепи с сигналом, указанным на чертеже схемы, и при обнаружении расхождения для определения места неисправности выполняет просмотр сигналов при движении от выхода к входу схемы.

Введение контрольной цифровой информации (сигнатур) в техническую документацию на цифровую аппаратуру очень удобно, так как для этой аппаратуры не представляется возможным документировать признаки правильной работы схем в виде параметров напряжений, токов или даже форм сигналов.

При использовании сигнатурного анализа отпадает необходимость хранить длинные двоичные коды эталонных реакций проверяемого устройства. Однако по-прежнему сохраняется проблема синтеза последовательностей тестов (векторов-стимулов).

Сигнатурный анализ благодаря использованию сравнительно быстродействующих узлов (сдвигающего регистра и сумматора по модулю 2) отличается высоким быстродействием по сравнению с другими методами сжатия информации, например циклическим суммированием или методом подсчета числа переключений в выходном векторе.

Высокое быстродействие СА позволяет в ряде случаев, например при диагностировании комбинационных схем, отказаться от синтеза минимальных или квазiminимальных тестовых последовательностей и просто использовать все допустимые комбинации входных двоичных векторов, которые в этом случае могут задаваться счетчиком.

Метод сигнатурного анализа обладает высокой достоверностью, которая в данном случае определяется вероятностью того, что различающиеся двоичные векторы имеют неодинаковые сигнатуры. Любая одиночная ошибка в выходном двоичном векторе обнаруживается с вероятностью, равной единице. При числе ошибок более одной вероятность обнаружения ошибок равна

$$1 - 1/2^n,$$

где n — число разрядов сигнатурного анализатора. При $n=16$ вероятность равна 0,99998. Это означает, что при неодионой ошибке в выходном векторе можно ожидать, что не более 0,002 % различных векторов будут иметь одинаковые сигнатуры.

При необходимости тестирования достаточно сложной микроаппаратуры для повышения достоверности результатов сигнатурного анализа можно увеличить длину сигнатурного генератора. Например, уже при $n=20$ вероятность обнаружения неисправности при многократной ошибке составляет 0,999999, другими словами, только одна миллионная часть всех возможных сигнатур состояний выходного вектора произвольной длины с неодиоными ошибками может быть одинаковой.

Совпадение сигнатур возможно только в том случае, когда при формировании сигнатуры последующие ошибки в выходном векторе компенсируют влияние предыдущих ошибок (искажений).

Можно показать, что при двойной ошибке компенсация имеет место, когда эти ошибки разделены $2^n - 1$ нулями, что является маловероятным событием.

Высокая достоверность метода является, в частности, следствием свойства линейности сдвигового регистра с обратными связями. Это свойство позволяет использовать принцип суперпозиции входных сигналов и считать, что реакция регистра на сумму по модулю 2 двух входных сигналов равна сумме его реакций от каждого из сигналов в отдельности.

Если считать, что на один вход подан безошибочный входной вектор, а на второй — вектор ошибок, то результирующая сигнатура будет равна сумме по модулю 2 сигнатур, которые были бы получены от двух отдельно поданных на вход регистра векторов.

Ошибка обнаруживается в том случае, когда сигнатура вектора ошибок не является нулевой. Ошибки не обнаруживаются, если они расположены по длине двоичного вектора так, что по цепям обратных связей они взаимно компенсируются.

Структура цепей связи (места съема сигналов обратной связи) в сдвиговом регистре на рис. 8.41 выбрана из условия получения наибольшей достоверности результата.

Сигнатурные анализаторы можно использовать для локализации неисправностей устройств, в которых при разработке заложена эта возможность. В частности, в устройстве должен быть предусмотрен тактовый сигнал, по

которому начинается прием данных в сдвиговой регистр анализатора. Должен быть предусмотрен также сигнал запуска и останова, с тем чтобы длительность поиска данных была строго определенной, а также средства разрыва обратных связей, иначе сигнатуры будут неоднозначными [10].

Глава девятая

ПРОЕКТИРОВАНИЕ ДИАГНОСТИЧЕСКИХ ТЕСТОВ

9.1. ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ СИСТЕМ

ПРОЕКТИРОВАНИЯ ДИАГНОСТИЧЕСКИХ ТЕСТОВ (СПДТ) ДЛЯ ЭВМ

При разработке тестового обеспечения эксплуатации ЭВМ возникают две задачи:

разработка диагностических тестов для ЭВМ («машинная диагностика»);

разработка диагностических тестов для сменных блоков (ТЭЗ) машины.

В данной главе сначала рассматриваются вопросы проектирования «машинной диагностики», а затем методика проектирования тестов для сменных блоков.

Проектирование диагностических тестов для ЭВМ состоит из двух основных этапов:

1) определения тестовых наборов, обеспечивающих полную проверку схем ЭВМ;

2) разработки диагностических справочников.

Для проектирования диагностических тестов используются описание структуры ЭВМ на каком-либо языке и данные о размещении логических схем в конструкции (рис. 9.1).

В большинстве случаев в качестве описания ЭВМ используется массив описаний логических схем системы автоматизации проектирования. Может также использоваться описание структуры ЭВМ в виде графа или описание алгоритмов функционирования ЭВМ в виде блок-схем алгоритма.

Конкретные алгоритмы проектирования диагностических тестов зависят от метода диагностирования, реализованного в ЭВМ. Так, например, при методах двухэтапного диагностирования и последовательного сканирования проектирование диагностических тестов основывается на выделении регистров и фрагментов комбинационных схем и синтезе тестов для этих фрагментов. При методе микро-

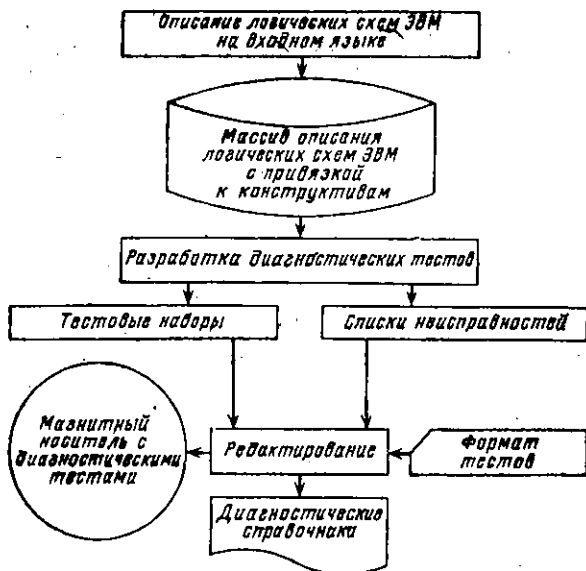


Рис. 9.1. Общая структура системы проектирования диагностических тестов

диагностирования задача проектирования диагностических тестов заключается в определении микропрограмм, обеспечивающих транспортировку наборов к проверяемому блоку и результатов — к схемам анализа. При методе диагностирования, ориентированном на проверку ТЭЗ, задача сводится к синтезу тестов проверки ТЭЗ, содержащих как последовательную, так и комбинационную логику. При методе эталонных состояний в качестве диагностических тестов используются наборы команд и операндов, обеспечивающих выполнение всех алгоритмов управления.

На этапе разработки диагностических справочников выполняется моделирование неисправностей, логическое или физическое. На основании результатов моделирования формируется диагностический справочник, в котором каждому исходу тестов ставится в соответствие список подозреваемых на неисправность сменных блоков.

При логическом моделировании неисправностей необходимо обеспечить адекватное представление логико-временного поведения цифрового устройства в процессе диагностирования [6]. Обычно используют асинхронное моделирование по принципу собатийности, т. е. в каждый

моделируемый момент времени (такт моделирования) определяются выходы только тех элементов схемы, у которых хотя бы один выход изменил свое состояние [36].

Недостатком логического моделирования являются большие затраты машинного времени на моделирование, что часто делает невозможным его применение. Для получения приемлемого времени моделирования необходимы следующие условия [57]:

быстродействующие ЭВМ;

отказ от универсальности программ моделирования;

использование двух-трех уровней описания моделируемой логики;

использование (по возможности) микропрограммной поддержки часто работающих фрагментов монитора моделирования;

сокращение списка моделируемых неисправностей путем анализа топологии схемы для выявления групп принципиально неразличимых неисправностей и моделирования только представителей каждой группы (это сокращает список неисправностей в среднем на 20 %) [36].

Физическое моделирование неисправностей может выполняться, например, способом, показанным на рис. 9.2.

Специальный переходник, изображенный на рис. 9.2, позволяет по сигналам УПР1 логически имитировать 0 и 1 на контактах разъема ТЭЗ. Номер контакта и тип неисправности задает блок управления, управляемый от ЭВМ [57].

Физическое моделирование неисправностей выполняется следующим образом:

моделирующая программа начинает работу с выдачи в блок управления информацией УПР2 о типе контакта (вход/выход) и режиме имитации (фаза I);

программа переводит блок управления в режим имитации и запускает тесты (фаза II);

в это время программное управление снято, а программа управления ждет, когда в блоке сработает режим неисправности, выполнится регистрация и управление перейдет к моделирующей программе (фаза III). В фазе III моделирующая программа редактирует результаты и записывает их для дальнейшей обработки и генерации диагностических справочников.

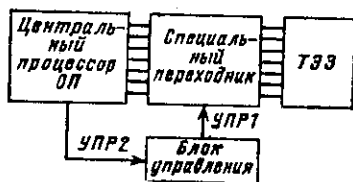


Рис. 9.2. Способ физического моделирования неисправностей

9.2. ПРОЕКТИРОВАНИЕ ТЕСТОВ ДЛЯ СИСТЕМ ДИАГНОСТИРОВАНИЯ, РЕАЛИЗУЮЩИХ МЕТОД ДВУХЭТАПНОГО ДИАГНОСТИРОВАНИЯ

Проектирование тестов в этом случае состоит из следующих этапов:

- 1) выделения триггеров и регистров, имеющих непосредственную установку и опрос;
- 2) построения тестов по принципу «установка 0» — «установка 1» — «установка 0»;
- 3) составления ТЛН для триггеров и регистров, имеющих непосредственную установку и опрос;
- 4) формирования диагностических справочников для тестов этого цикла по принципу однозначного соответствия «триггер — ТЭЗ»;
- 5) выделения фрагментов комбинационных схем, заключенных между проверенными регистрами;
- 6) построения тестов для комбинационных схем;
- 7) упорядочения тестов в условную диагностическую процедуру;
- 8) составления ТЛН по дереву решений;
- 9) формирования диагностических справочников для тестов комбинационных схем [60].

Комплекс программ для подготовки тестов локализации неисправностей в качестве входных данных использует описание логических схем ЭВМ, хранимое на ленте основного логического массива автоматизации проектирования. Этот комплекс программ подготавливает данные для ТЛН, упорядочивает их, модифицирует при внесении изменений, выдает документацию. Структура комплекса программ для составления ТЛН показана на рис. 9.3. Из основного логического массива извлекается информация, определяющая логические схемы ЭВМ, определяются регистровые и триггерные схемы, выделяются комбинационные схемы. Информация о физическом размещении схем (конструктивные адреса) отделяется от информации о логической структуре.

Следующая программа предназначена для синтеза тестов комбинационных схем и использует один из методов, приведенных в § 9.7. Результатом работы этой программы являются входные наборы для проверки комбинационных схем. После нахождения очередного набора с помощью моделирующей программы определяется, какие еще неис-

правности он обнаруживает. Одновременно ведется список необнаруженных неисправностей, для которых программа синтезирует дополнительные тесты. После этого полученное множество входных наборов организуется в условную диагностическую процедуру, которая обеспечивает требуемую точность локализации.

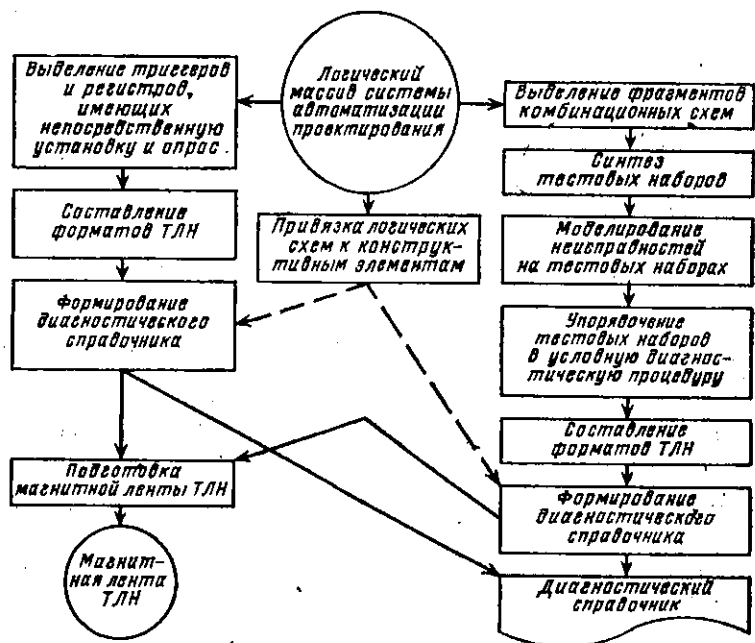


Рис. 9.3. Система проектирования диагностических тестов для метода стандартных форматов

Очередная программа комплекса преобразует наборы в стандартные форматы. Комплекс программ включает в себя также программу подготовки диагностических справочников. Обычно в диагностических справочниках для систем диагностирования, использующих метод двухэтапного диагностирования, указываются все элементы логических схем, активные в конкретном тестовом наборе, номер теста и все относящиеся к проверяемому фрагменту комбинационной схемы номера контактов разъемов ТЭЗ и значения сигналов на них. Эти сигналы можно наблюдать

с помощью осциллографа при зацикливании тестового примера. Указываются также конструктивные адреса ТЭЗ, в которых размещен проверяемый фрагмент.

9.3. ПРОЕКТИРОВАНИЕ ТЕСТОВ ДЛЯ СИСТЕМ ДИАГНОСТИРОВАНИЯ, РЕАЛИЗУЮЩИХ МЕТОД ЭТАЛОННЫХ СОСТОЯНИЙ

Проектирование тестов при этом методе состоит из следующих этапов (рис. 9.4):

1) вычисления условий прохождения всех ветвей алгоритма работы устройства и подготовки программы, обеспечивающей выполнение этих условий;

2) вычисления эталонных состояний устройства на тактах проверки и масок для исключения неопределенности состояния;

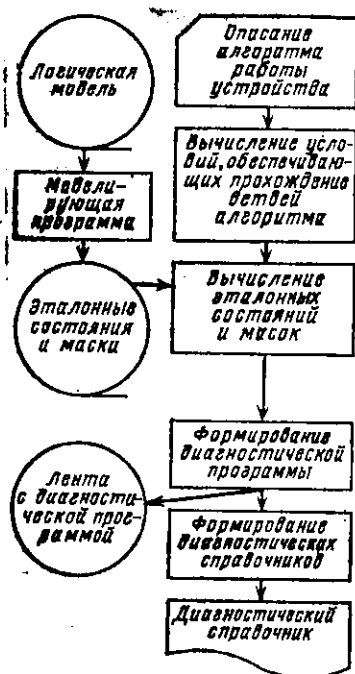


Рис. 9.4. Система проектирования диагностических тестов для метода эталонных состояний

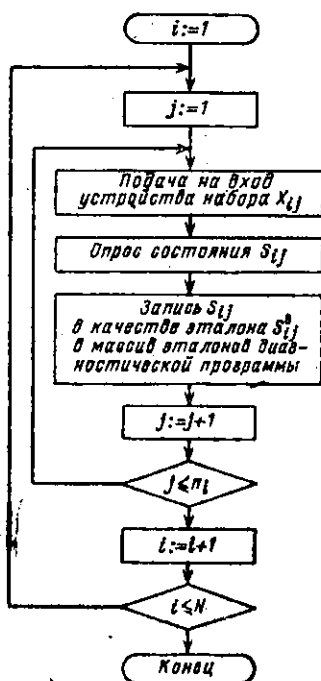


Рис. 9.5. Снятие «физического эталона» с работающего устройства:

i — номер такта ветви алгоритма;
 j — номер ветви алгоритма

- 3) формирования диагностической программы;
- 4) формирования диагностических справочников.

При представлении алгоритмов функционирования устройства в виде графа каждому допустимому пути в графе алгоритма ставятся в соответствие входные условия, обеспечивающие прохождение определенной ветви алгоритма.

Совокупность входных условий превращается в последовательность команд диагностической программы, поступающих на вход устройства и задающих в устройстве режим принудительной синхронизации, при котором выполнение каждого очередного такта проверяемого устройства задается диагностической программой.

Условия прохождения всех путей вычисляются по графу алгоритма или по его табличному описанию.

Эталонные состояния могут быть получены с помощью логического моделирования, или, как это чаще практикуется, методом «физического эталона», т. е. путем снятия эталонов с работающего устройства (рис. 9.5).

В силу неопределенности состояний некоторых триггеров в той части алгоритма, в которой не происходит их установка, одной и той же группе задающих команд могут соответствовать различные последовательности состояний. Использование этих состояний в качестве эталона при диагностике неисправного устройства требует устранения такой неопределенности. Это выполняется с помощью автоматически или вручную генерируемой маски, которая позволяет отсечь значения неопределенных сигналов [6].

Формирование диагностических справочников выполняется вручную, путем логического или физического моделирования.

При разработке диагностических справочников вручную каждому биту состояния устройства ставится в соответствие триггер m_i , который размещен в ТЭЗ с конструктивным адресом T_i .

Таблица соответствия $m_i—T_i$ и образует диагностический справочник. При сравнении действительного состояния устройства с эталонным диагностическая программа определяет номер элемента m_i , состояние которого не совпадает с эталонным и задает, таким образом, вход в диагностический справочник.

Этот метод, однако, имеет недостаток, так как в сообщении о неисправности указывается фактически следствие, а не причина возникновения неисправности.

Существует и другой метод ручного составления диагностического словаря. Он основан на анализе конструк-

тивных компонентов и структурно-конструкционной модели устройства, отражающей его логическую структуру и связь с конструктивами.

9.4. ПРОЕКТИРОВАНИЕ ТЕСТОВ ДЛЯ СИСТЕМ ДИАГНОСТИРОВАНИЯ, РЕАЛИЗУЮЩИХ МЕТОД МИКРОДИАГНОСТИРОВАНИЯ

При проектировании микродиагностирования логическая структура ЭВМ задается в виде направленного графа, представляющего собой совокупность вершин и дуг. Вершины графа представляют собой функциональные блоки, такие, как регистр, триггер, комбинационная схема, выполняющая логическое преобразование информации (сумматор, свертка, схема анализа). Дугам графа соответствуют микрооперации.

Вершина графа, в которую поступают тестовые наборы из накопителя тестов, является входной. Вершина графа, из которой результаты поступают на сравнение с эталоном, является выходной. Как отмечалось ранее, при микродиагностировании сравнение результатов проверки с эталонными может выполняться либо с помощью имеющейся в процессоре аппаратуры, такой как, например, сумматор или схема анализа, либо с помощью дополнительной аппаратуры опроса состояния ЭВМ и сравнения ее с эталонной.

В первом случае выходная вершина является входом аппаратуры анализа и сравнение с эталоном выполняется с помощью стандартной микропрограммы сравнения и ветвления, а во втором случае выходных вершин может быть несколько (в принципе каждая вершина графа может быть выходной, если все средства ЭВМ имеют выход на опрос по сериализации), а сравнение с эталоном выполняется аппаратурой блока диагностики или сервисного процессора при помощи специальных диагностических операций.

Рассмотрим ряд понятий теории графов, используемых далее при описании алгоритма проектирования микродиагностики.

Путь из вершины v_i в вершину v_j есть упорядоченная последовательность дуг, связывающих v_i и v_j . Циклом называется путь, начинающийся и заканчивающийся в одной и той же вершине. Вершина v_j достижима из вершины v_i , если имеется путь из вершины v_i в вершину v_j . Если G — направленный граф с n вершинами, то его матрица связно-

сти есть матрица размерности $n \times n$, элемент C_{ij} которой равен 1, если имеется дуга $v_i v_j$. В противном случае $C_{ij} = 0$.

Диагностическая микропрограмма, проверяющая какой-либо блок ЭВМ, выполняет следующие действия:

чтение тестового набора из накопителя микродиагностики;

транспортировку набора на вход проверяемого блока;

опрос состояния проверяемого блока;

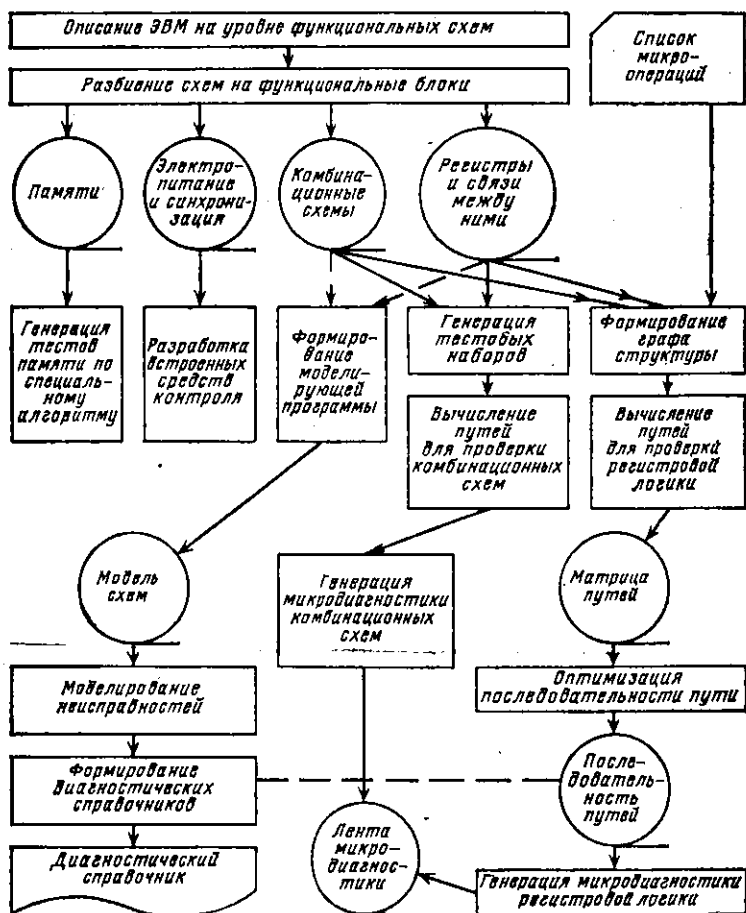


Рис. 9.6. Система проектирования диагностических тестов для метода микродиагностирования

чение эталона из накопителя микродиагностики;
передачу эталона в блок сравнения;
анализ результата и сообщение о неисправности.

Основной задачей при проектировании микродиагностики является нахождение путей в графе G , обеспечивающих условия транспортировки и транспортировку тестовых наборов из входной вершины на входы проверяемой вершины, а также транспортировку результатов проверки в выходную вершину. Последовательность найденных путей должна обеспечивать минимальное приращение проверенных вершин и дуг по отношению к предыдущему пути. Совокупность всех путей должна обеспечивать проверку всех связей и функциональных блоков.

Алгоритм проектирования микродиагностики в общем случае состоит из следующих этапов (рис. 9.6):

- определения путей по матрице связности графа G ;
- вычисления микропрограммы по найденному пути;
- дополнения вычисленной микропрограммы стандартными микропрограммами сравнения, ветвления и сообщения о неисправности.

Проектирование диагностических справочников для микродиагностики может выполняться вручную путем разработки таблиц соответствия между функциональными блоками и связями между ними и конструктивными адресами ТЭЗ, в которых расположены эти блоки и связи между ними, либо с помощью логического или физического моделирования.

Поскольку вся структура ЭВМ состоит из двух типов функциональных блоков и связей между ними: комбинационных схем преобразования информации, таких как сумматоры, свертки, дешифраторы и т. д.; регистровой логики и передач между регистрами, процедура вычисления микродиагностики состоит из двух этапов [92]:

- 1) вычисления микродиагностики регистров и передач между ними;
- 2) вычисления микродиагностики комбинационных схем.

Вычисление путей для проверки регистровой логики

Алгоритм вычисления путей для проверки регистровой логики заключается в следующем.

Шаг 1. Формируется матрица связности S направленного графа G .

Шаг 2. На матрице связности обозначаются входная и выходная вершины. При наличии нескольких входных или

выходных вершин вводится входная или выходная псевдовершина.

Шаг 3. Входная вершина принимается за исходную и выбираются все вершины, связанные со входной вершиной. Проводятся линии, соответствующие дугам между входной вершиной и другими выбранными вершинами. Входная вершина считается принадлежащей уровню 0, а инцидентные ей вершины — уровню 1.

Шаг 4. Все вершины, выбранные на шаге 3, принимаются за исходные. Процедура шага 3 итеративно повторяется до тех пор, пока не будут покрыты все вершины графа *G*. Если какая-либо вершина уже принадлежала предыдущему или этому же уровню, то она принимается за псевдовершину и продвижение по этой линии прекращается. Граф, сформированный по описанной выше процедуре, называется псевдодеревом.

Шаг 5. Для завершения пути от псевдовершины к выходной вершине среди уже найденных путей выбирается кратчайший, соединяющий псевдовершину с выходной вершиной. Этот путь присоединяется к псевдовершине. Таким образом формируется дерево путей.

Если граф содержит максимальные сильносвязанные подграфы, процедура шага 5 разбивается на следующие шаги:

шаг 5.1 — выбираются пути между входной и выходной вершинами, не содержащие псевдовершин;

шаг 5.2 — к псевдовершине графа присоединяются части путей, найденных на предыдущем шаге и соединяющих псевдовершины с выходной вершиной;

шаг 5.3 — формируются пути, содержащие 1, 2, ..., *n*-кратные циклы, обеспечивающие покрытие всех вершин и дуг графа. Эти пути заканчиваются псевдовершинами и для продолжения их до выходной вершины дополняются найденными на шаге 5.1 частями путей, соединяющих псевдовершины с выходной вершиной.

Шаг 6. Возможные пути выбираются из полного дерева. Множество путей, полученных по приведенному алгоритму, обеспечивает покрытие всех вершин и дуг графа.

На основании полученных путей формируется матрица путей, показывающая принадлежность вершин и дуг тому или иному пути.

В качестве примера приведем граф, показанный на рис. 9.7. На шаге 1 формируется его матрица связности. На шаге 2 определяется в качестве входной вершина 1, а в качестве выходной — вершина 4. На шагах 3 и 4 форми-

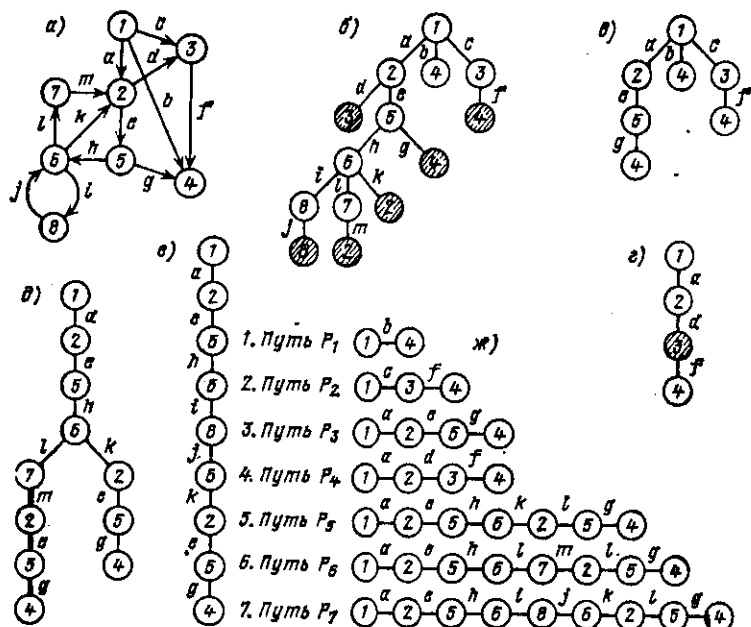


Рис. 9.7. Вычисление путей по графу структуры ЭВМ:

а — исходный граф; б — граф типа «псевдодерево» (заштрихованы псевдовершины); в — шаг 5.1; г — шаг 5.2; д — шаг 5.3; е — шаг 5.3; ж — возможные пути (шаг 6); з — иллюстрирует шаг 5, представляющий собой процедуру получения полного графа

руется псевдодерево (см. рис. 9.7, б). На шаге 5.1 находится путь между входной и выходной вершинами, не содержащий псевдовершин (рис. 9.7, в). На шаге 5.2 дуга 3—4, принадлежащая пути 1—3—4, соединяется с псевдовершиной 3 (рис. 9.7, г). На шаге 5.3 вычисляется путь, содержащий один цикл (рис. 9.7, д). На шаге 5.3 вычис-

	1	2	3	4	5	6	7	8	a	b	c	d	e	f	g	h	i	j	k	l	m
P_1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P_2	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
P_3	1	1	0	1	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0
P_4	1	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
P_5	1	1	0	1	1	1	0	0	1	0	0	0	1	0	1	1	0	0	1	0	0
P_6	1	1	0	1	1	1	1	0	1	0	0	0	1	0	1	1	0	0	0	1	1
P_7	1	1	0	1	1	1	0	1	1	0	0	0	1	0	1	1	1	1	1	0	0

Рис. 9.8. Матрица путей

ляется путь, содержащий двойной цикл, и формируется полное дерево (рис. 9.7, е). На шаге 6 составляются все возможные пути по дереву (рис. 9.7, ж). Матрица путей показана на рис. 9.8. Как видно из рисунка, неисправности в элементах k и 4 , e и g , l и t неразличимы.

Генерация микродиагностики для проверки регистровой логики

Для генерации микродиагностики по найденным реальным путям необходимо построение таблицы соответствия между микрооперациями (МО) и вершинами и дугами графа системы. Пример таблицы соответствия показан на рис. 9.9. Каждой микрооперации соответствуют входная и выходная вершины.

Микроопера- ция	Дуга графа	Микроопера- ция	Дуга графа
МО ₁	1→2	МО ₈	5→6
МО ₂	1→4	МО ₉	6→8
МО ₃	1→3	МО ₁₀	8→6
МО ₄	2→3	МО ₁₁	6→2
МО ₅	2→5	МО ₁₂	6→7
МО ₆	3→4	МО ₁₃	7→2
МО ₇	5→4		

Рис. 9.9. Таблица микроопераций

Таблица микроопераций строится следующим образом.

Шаг 1. Микрооперации классифицируются по трем группам:

- МО передач;
- арифметические;
- МО анализов условий.

Шаг 2. Выбираются микрооперации передач между регистрами.

Шаг 3. Последовательность логических блоков с учетом микроопераций представляется в виде направленного графа.

Шаг 4. Если на шаге 3 выяснилось, что микрооперация имеет более двух входных и выходных вершин, необходимо перечислить все комбинации последовательностей логических блоков в таблице микроопераций.

Процедура генерации микродиагностики по найденным путям состоит в следующем.

Шаг 1. Последовательность логических блоков в тестовом пути заменяется соответствующей микрооперацией из таблицы микроопераций.

Шаг 2. Шаг повторяется до тех пор, пока весь путь не будет заменен последовательностью микроопераций.

Шаг 3. К последовательности микроопераций, соответствующей пути, в начале добавляется микропрограмма считывания тестового набора из накопителя микродиагностики, а в конце — микропрограмма сравнения с эталоном.

Шаг 4. Описанная выше процедура применяется ко всем найденным путям. Например, пути P_5 (рис. 9.7, ж) соответствует микропрограмма

$$MO_1 \rightarrow MO_5 \rightarrow MO_8 \rightarrow MO_{11} \rightarrow MO_5 \rightarrow MO_7.$$

Последовательность микроопераций в начале дополняется микропрограммой чтения тестового набора ЧТН, а в конце — микропрограммой сравнения с эталоном СРВ. На рис. 9.10 показаны микропрограммы, соответствующие найденным путям.

Путь	Микропрограммы
P_1	ЧТН $\rightarrow$ $MO_2 \rightarrow$ СРВ
P_2	ЧТН $\rightarrow$ $MO_3 \rightarrow$ $MO_6 \rightarrow$ СРВ
P_3	ЧТН $\rightarrow$ $MO_1 \rightarrow$ $MO_5 \rightarrow$ $MO_7 \rightarrow$ СРВ
P_4	ЧТН $\rightarrow$ $MO_1 \rightarrow$ $MO_4 \rightarrow$ $MO_6 \rightarrow$ СРВ
P_5	ЧТН $\rightarrow$ $MO_1 \rightarrow$ $MO_5 \rightarrow$ $MO_8 \rightarrow$ $MO_{11} \rightarrow$ $MO_5 \rightarrow$ $MO_7 \rightarrow$ СРВ
P_6	ЧТН $\rightarrow$ $MO_1 \rightarrow$ $MO_5 \rightarrow$ $MO_8 \rightarrow$ $MO_{12} \rightarrow$ $MO_{13} \rightarrow$ $MO_5 \rightarrow$ $MO_7 \rightarrow$ СРВ
P_7	ЧТН $\rightarrow$ $MO_1 \rightarrow$ $MO_5 \rightarrow$ $MO_8 \rightarrow$ $MO_9 \rightarrow$ $MO_{10} \rightarrow$ $MO_{11} \rightarrow$ $MO_5 \rightarrow$ $MO_7 \rightarrow$ СРВ

Рис. 9.10. Таблица микропрограмм, соответствующих путям.

Генерация микродиагностики для проверки комбинационных схем

Для проверки многовходовых комбинационных схем необходимо найти совокупность путей, обеспечивающих подачу тестовых наборов на входы комбинационной схемы, т. е. на регистры, являющиеся входами комбинационной схемы и соединяющие выходной регистр комбинационной схемы с выходной вершиной графа системы.

С этой целью используются фрагменты путей для проверки регистровой логики. Например, если вершины 2 и 3 (рис. 9.7, а) являются входами сумматора, а вершина 5 — его выходом, то микропрограмма проверки сумматора на

одном наборе будет состоять из следующих фрагментов, найденных на предыдущем этапе микропрограмм: $ЧТН \rightarrow MO_3$, $ЧТН \rightarrow MO_1$, $MO_7 \rightarrow CPB$ и выполнения микрооперации сумматора, т. е. MO_c . Полная микропрограмма будет иметь вид

$ЧТН, MO_3, ЧТН, MO_1, MO_c, MO_7, CPB$.

Для проверки сумматора на N наборах данная микропрограмма должна быть повторена N раз.

9.5. ПРОЕКТИРОВАНИЕ ТЕСТОВ ДЛЯ СИСТЕМ ДИАГНОСТИРОВАНИЯ, РЕАЛИЗУЮЩИХ МЕТОД ПОСЛЕДОВАТЕЛЬНОГО СКАНИРОВАНИЯ

Данный метод ориентирован на отдельную проверку регистровой и комбинационной логики. Поэтому в системах проектирования диагностических тестов для ЭВМ, использующих данный метод, много общего с системами проектирования, реализованных с помощью метода двух-эталонного диагностирования. Различие заключается в этапе редактирования, т. е. когда выполняется формирование окончательных машинных форматов тестов после вычисления тестовых наборов.

9.6. ПРОЕКТИРОВАНИЕ ТЕСТОВ ДЛЯ ПРОВЕРКИ ТЭЗ

На рис. 9.11 приведена структура системы автоматизированного проектирования тестов для ТЭЗ. Поскольку ТЭЗ содержит как комбинационные схемы, так и схемы памяти (триггеры), одной из основных задач является эквивалентное представление схем памяти в модели, по которой выполняется синтез проверяющих тестов. Одним из способов представления является комбинационный эквивалент схем с памятью, который получается путем условного обрыва обратных связей [53]. При числе эквивалентов больше двух такие эквиваленты (копии) соединяются последовательно в итеративную сеть (рис. 9.12).

Нижние индексы при символах векторов входных X , внутренних Y и Y' и выходных Z переменных являются номерами тактов работы устройства. Для копии v ($v=1, 2, \dots, p$) вектор Y_v задает состояние схемы в момент времени v , вектор Y'_v — в момент времени $v+1$, $Y'_v = Y_{v+1}$. Для итеративной сети внешними входами являются входы X всех копий и входы Y_1 первой копии, а внешними выходами — выходы Z всех копий и выходы Y'_p последней копии.

К построенной таким образом итеративной сети могут быть применены методы построения тестов для комбинационных схем, описанные в § 9.7.

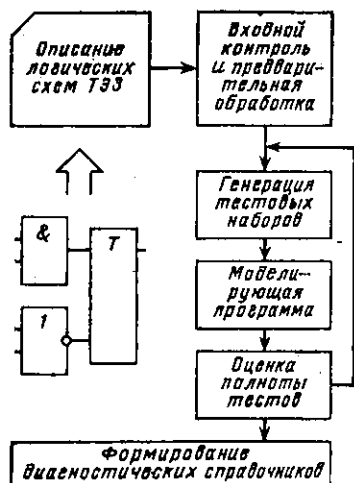


Рис. 9.11. Система проектирования диагностических тестов для метода, ориентированного на проверку ТЭЗ

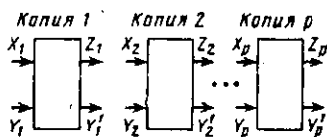


Рис. 9.12. Представление схем с памятью в виде итеративной комбинационной сети

Для синтеза диагностических тестов по таким моделям применяются упрощенные алгоритмы, которые делают необходимым дополнительный

анализ полученных тестов с помощью цифрового моделирования.

В процессе моделирования схем ТЭЗ на очередном полученном тестовом наборе выявляются случаи состязаний в цифровых схемах, и если тестовый набор приводит к состязаниям, то он не включается в множество синтезированных наборов. Обычно для моделирования используется метод триничного моделирования, хотя в некоторых работах используется многозначное моделирование, повышающее достоверность синтезированных тестов.

Большой объем вычислений при моделировании тестовых наборов требует поиска методов ускорения моделирования.

Для ускорения используется параллельное и событийное моделирование.

9.7. ПОСТРОЕНИЕ ДИАГНОСТИЧЕСКИХ ТЕСТОВ ДЛЯ КОМБИНАЦИОННЫХ СХЕМ

Неисправности, возникающие в комбинационных логических схемах, могут вызываться различными причинами и носить различный характер. Будем рассматривать неисправности только одиночные логического типа, т. е. такие, которые соответствуют появлению в какой-либо точке схемы устойчивого значения сигнала, соответствующего логическим 1 или 0.

Для обнаружения и локализации таких неисправностей используются тесты-наборы значений входных переменных, подаваемых на схему, и соответствующие им наборы значений выходных переменных, по которым можно обнаруживать неисправность схемы.

Методы построения тестов комбинационных схем изложены во многих работах, но только некоторые из них удобны и эффективны

при решении практических задач локализации неисправностей схем ЭВМ. Наибольшее распространение в настоящее время получили два метода: чувствительных путей и *D*-алгоритма [98, 99]. Оба эти метода позволяют строить совокупность тестов для комбинационной схемы, имеющей одиночный отказ логического типа. Рассмотрим сущность этих методов.



Рис. 9.13. Проверка логической схемы И

Пусть имеется произвольная комбинационная схема, содержащая схему И, в которой предполагается наличие неисправности. Нетрудно заметить, что работоспособность схемы И (рис. 9.13) может быть проверена с помощью следующих тестов:

<i>a</i>	<i>b</i>	<i>c</i>	<i>f</i>
0	1	1	0
1	0	1	0
1	1	0	0
1	1	1	1

Действительно, любой одиночный отказ, соответствующий появлению устойчивого логического 0 (устойчивый 0) на выходе схемы И, приведет к тому, что на наборе значений переменных $a=b=c=1$ значение f будет равно 0 вместо 1, а любой одиночный отказ, соответствующий устойчивому появлению логической 1 (устойчивая 1), приведет к тому, что хотя бы на одном из наборов значений переменных значение f будет равно 1 вместо 0.

При распространении такого подхода на любую комбинационную схему возникает трудность, связанная с тем, что в реальных схемах доступен выход не каждого элемента. В общем случае необходимо проверять работоспособность схемы, состоящей из большого числа внутренних недоступных элементов. Следовательно, проверяя некоторый элемент, необходимо сделать так, чтобы состояние этого элемента определяло значение выходного сигнала всей схемы. Посмотрим на примере, как это можно сделать (рис. 9.14).

Предположим, что проверяется наличие отказа на выходе элемента I. Тогда, если на схему поданы указанные на рис. 9.14 значения переменных, для проверки элемента I на наличие устойчивого 0 достаточно придать переменной a значение 1, а для проверки элемента I на наличие устойчивой единицы придать переменной a значение 0.

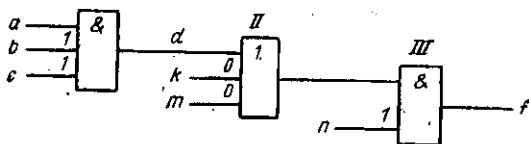


Рис. 9.14. Пример чувствительного пути

Таблица 9.1

x	$\bar{x}$
0	1
1	0
d	$\bar{d}$
$\bar{d}$	d

Таблица 9.2

x	y	xy	$x \vee y$
0	0	0	0
0	1	0	1
0	$\bar{d}$	0	$\bar{d}$
0	d	0	d
1	0	0	1
1	1	1	1
1	$\bar{d}$	$\bar{d}$	1
1	d	d	1
$\bar{d}$	0	0	$\bar{d}$
$\bar{d}$	1	$\bar{d}$	$\bar{d}$
$\bar{d}$	$\bar{d}$	$\bar{d}$	$\bar{d}$
$\bar{d}$	d	0	1
d	0	0	d
d	1	d	1
d	$\bar{d}$	0	1
d	d	d	d

Таблица 9.3

a	b	c
d	1	1
1	d	1
1	1	$\bar{d}$
d	d	1
d	1	d
1	d	d
d	d	d

Другими словами, выходной сигнал схемы определяется значением входного сигнала проверяемого элемента.

Основная идея метода чувствительных путей заключается в выборе пути от подозреваемого места отказа до выхода и поиске такого набора входных переменных, для которого значение выходной переменной определяется состоянием проверяемого элемента.

Процесс построения теста для предполагаемой неисправности можно разделить на два этапа. Первый этап заключается в прохождении от проверяемого элемента к выходу схемы и фиксации входных переменных каждого из элементов схемы, так чтобы значение состояния проверяемого элемента передавалось к выходу схемы по выбранному пути. Этот этап называется проталкиванием. Второй этап заключается в подборе таких входных значений переменных схемы, при которых фиксируются значения входных переменных элементов, выбранные на первом этапе. Этот этап называется «спуском» [26].

Чтобы иметь возможность одновременно рассматривать отказы любого вида (устойчивый 0 или устойчивая 1), состояние проверяемого элемента будем обозначать буквой d и определим логические операции с d следующим образом (табл. 9.1 и 9.2).

Здесь символ d рассматривается как обобщенная переменная, которая может принимать значение 1 или 0.

Вернемся к примеру на рис. 9.14. Фиксируя на выходе проверяемого элемента значения d , после проталкивания получаем значение d на выходе элементов II и III ($k=m=0, n=1$). На этапе спуска можно зафиксировать один из вариантов, приведенных в табл. 9.3.

Метод чувствительных путей оказывается значительно более эффективным, если подбор значений входных переменных производить не только среди значений 0 или 1, но и среди значений d и $\bar{d}$. Действительно, рассмотрим схему на рис. 9.15.

Проверим выход элемента I и выберем путь через элементы II и III. На этапе проталкивания зафиксируем для x_1 значение 1 и для

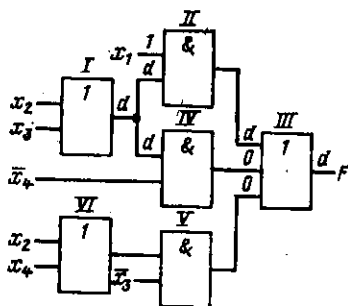


Рис. 9.15. Пример схемы к методу чувствительных путей

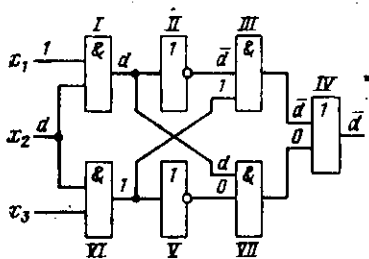


Рис. 9.16. Пример схемы к методу чувствительных путей

выходов элементов IV и V — значение 0. Так как чувствительным делается только один путь, то на этапе спуска будем последовательно требовать для входа $\bar{x}_4$ элемента IV значение 0, а следовательно, для входа x_4 элемента VI значение 1 и для входа $\bar{x}_3$ элемента V значение 0, что приводит к значению x_3 элемента I, равному 1. Если выбрать $x_3=1$ для элемента I, то получить на выходе d невозможно.

В то же время значение 0 на выходе элемента IV может быть получено при $x_4=d$ и значение 0 на выходе элемента V — при $x_3=d$ и $x_2=0$. В итоге получим тест вида

x_1	x_2	x_3	x_4	F
1	0	d	d	d

т. е. при проверке отказа в элементе I вида устойчивый 0 для $x_1=1, x_2=0, x_3=1, x_4=1$ выходной сигнал $F \neq 1$ и при проверке отказа вида устойчивая 1 для $x_1=1, x_2=0, x_3=0, x_4=0$ выходной сигнал $F \neq 0$.

Метод чувствительных путей удобен для практического применения благодаря его простоте. Будучи запрограммированным для ЭВМ, он

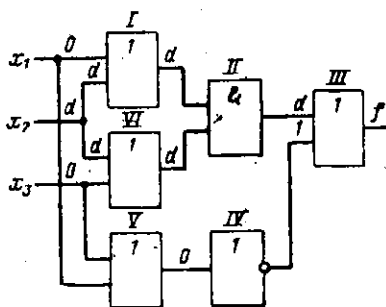


Рис. 9.17. Пример схемы к D-алгоритму

оказывается довольно экономичным с точки зрения времени выполнения программы и требуемого объема памяти.

Основной недостаток этого метода состоит в том, что при формальном его использовании не всегда удается получить тесты. Это объясняется тем, что не в любой схеме можно осуществить проталкивание d по любому пути. Действительно, пусть имеем схему, представленную на рис. 9.16, для которой необходимо протолкнуть d от входа x_2 через элементы I—IV.

После проталкивания d получаем, что выходной сигнал элемента VI должен иметь значение 1, а это невозможно. Объясняется это тем, что d проталкивается через элемент I только по пути I, VII, IV.

Эту трудность можно обойти, отыскивая каждый раз путь, по которому можно протолкнуть d .

Другой выход из затруднения заключается в том, что в схеме, для которой невозможно осуществить проталкивание d к выходу по некоторому пути, делается дополнительный внешний выход из комбинационной схемы — из той ее точки, до которой проталкивание оказалось возможным (в данном примере такой точкой является выход элемента II).

Таким образом, недостатком метода «чувствительных путей» можно считать то, что каждый раз отыскивается только один путь от места предполагаемой неисправности к выходу схемы.

Этот недостаток устраняется при использовании метода D-алгоритма. Основная идея метода заключается в одновременном проталкивании d от места неисправности по всем возможным путям схемы. При этом на каждом шаге проверяется возможность проталкивания d по всем путям. Иногда это невозможно, и тогда d не проталкивается по некоторым из путей.

Пусть, например, имеем схему, представленную на рис. 9.17. В этой схеме невозможно протолкнуть d от x_2 к выходу по всем возможным путям, так как для этого x_1 и x_3 должны иметь значение 0, а при этом выходной сигнал элемента IV имеет значение 1, что недопустимо. В то же время d можно протолкнуть на выход по любому одному пути схемы от x_2 .

После завершения этапа проталкивания выполняется этап спуска, завершающийся получением тестового набора значений переменных для предполагавшейся неисправности.

Таким образом, для построения тестов необходимо фиксировать одну за другой все предполагаемые неисправности схемы и строить для них тесты. Практически поступают несколько иначе. Как правило, с помощью теста, построенного для обнаружения некоторой неисправности, можно обнаружить и ряд других неисправностей. Так, например, с помощью теста, построенного для схемы на рис. 9.17, на пути от x_2 к выходу через I, II и III можно проверить не только x_2 , но и выходы I, II и III.

Учитывая это, получаем совокупность тестов для проверки неисправностей схемы, причем известно, какой тест какую неисправность обнаруживает.

Зная это, можно построить таблицу неисправностей, каждый столбец которой отмечается некоторой неисправностью, каждая строка — тестом, а в клетке на пересечении i -й строки и j -го столбца ставится единица, если i -й тест обнаруживает j -ю неисправность.

С помощью таблицы неисправностей можно завершить построение совокупности тестов. Совокупность тестов может иметь различную длину (число наборов). Очевидно, желательно выбирать совокупность минимально возможной или близкой к ней длины. Задача поиска совокупности тестов минимальной длины для обнаружения (контроля) всех возможных неисправностей состоит в поиске минимального набора строк, имеющего единицы в каждом столбце таблицы. Эта задача совпадает с задачей кратчайшего покрытия (минимизации) в теории логических схем и может решаться такими же методами.

Задача поиска совокупности тестов минимальной длины для диагностики (локализации) отказов решается несколько иначе. Удобный метод решения этой задачи сводится к следующему. Каждому тесту сопоставляется весовой коэффициент, равный числу пар различаемых неисправностей. Это число получается как произведение числа отмеченных и числа неотмеченных клеток в строке таблицы неисправностей, соответствующей данному тесту. Тест с максимальным весовым коэффициентом выбирается как входящий в строящуюся совокупность. При этом таблица разделяется на две новые. В одну входят все неисправности, обнаруживаемые выбранным тестом, а в другую — все необнаруживаемые. Вычисляем весовые коэффициенты тестов $w_1^1, w_2^1, \dots, w_k^1$ для первой таблицы и $w_1^2, w_2^2, \dots, w_k^2$ для второй, а затем выбираем тест T_j , для которого величина $w_j^1 \times w_j^2$ максимальна.

При этом число рассматриваемых таблиц увеличивается. С ними проделываем такую же процедуру, и для k таблиц выбираем тест T_p с максимальным значением $w = \prod_{i=1}^k w_p^i$. Выбор тестов для строящейся совокупности прекращается после того, как любой вычисляемый весовой коэффициент теста для любой из таблиц равен нулю. В результате получается диагностическая совокупность тестов. Эта процедура может быть запрограммирована и реализована на ЭВМ.

9.8. ФУНКЦИОНАЛЬНОЕ ДИАГНОСТИРОВАНИЕ КОМБИНАЦИОННЫХ СХЕМ

Построение диагностических тестов комбинационных схем по их математической модели является сложной и трудоемкой процедурой, а само диагностирование требует специальных устройств с большой памятью для хранения тестовых воздействий и эталонных результатов на каждом тестовом воздействии. С ростом степени интеграции комбинационные схемы, как правило, размещаются в большой интегральной микросхеме, и их диагностирование сводится к контролю работоспособности. В связи с этим становятся интересными методы диагностирования комбинационных схем, не требующие построения тестов и хранения большого числа тестовых воздействий и эталонных результатов. К подобным методам относятся рассмотренный в § 8.14 метод сигнатурного анализа и рассматриваемый ниже *метод синдромного диагностирования*. Последний основан на переборе всех входных наборов схемы и сжатом представлении результатов диагностирования на всем множестве входных наборов.

Синдромом булевой функции (комбинационной схемы) называется отношение $S = K/2^n$, где K — число входных наборов, на которых функция имеет значение 1, а n — число входов схемы; $0 \leq S \leq 1$ [102].

Синдром является функциональным свойством комбинационной схемы, так как все реализации одной функции имеют один и тот же синдром.

На рис. 9.18 приведены синдромы простых n -входовых комбинационных схем, а на рис. 9.19 — зависимость между синдромом схемы и образующих ее подсхем, объединенных инвертором, схемами И, ИЛИ и сложения по модулю 2.

Вычислим в качестве примера синдром схемы, показанной на рис. 9.20:

$$S_1 = 1 - 2^{-2} = 3/4;$$

$$S_2 = 1 - 2^{-2} = 3/4; \quad S_3 = 2^{-3} = 1/8;$$

$$S_4 = 1 - (S_2 + S_3 - S_2 S_3) = 7/32;$$

$$S = S_1 S_4 = 21/128, \quad K = 21.$$

Схема может быть диагностирована по синдрому, если синдром неисправной схемы отличается от синдрома исправной.

Тестовая процедура заключается в подаче на вход схемы всех входных наборов, определении синдрома (обычно с помощью счетчика) и сравнении его с эталонным синдромом (требуется всего один эталон).

Устройство для диагностирования комбинационных схем приведено на рис. 9.21.

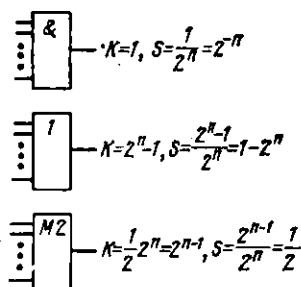


Рис. 9.18. Синдромы простых n -входных функций

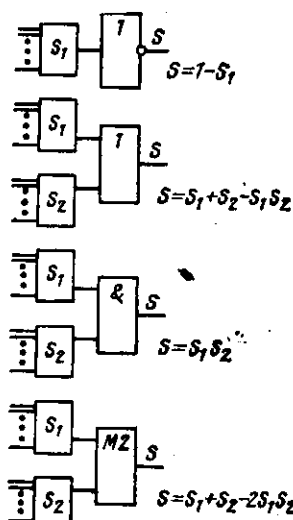


Рис. 9.19. Синдромы сложных функций

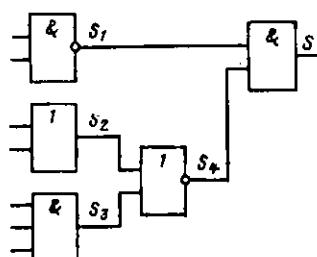


Рис. 9.20. Пример схемы для вычисления синдрома

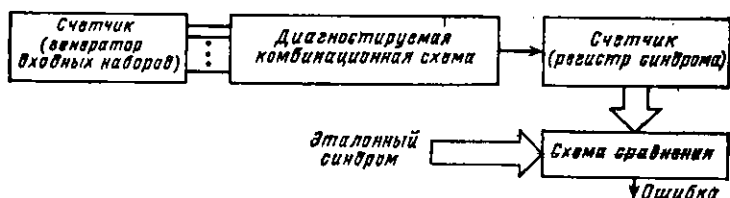


Рис. 9.21. Схема диагностирования комбинационных схем

На проверяемую схему от счетчика подаются все входные наборы. Выход проверяемой схемы соединен со счетчиком синдрома, который подсчитывает число единиц на выходе проверяемой схемы. После перебора всех выходных наборов производится сравнение полученного и эталонного синдромов.

Заметим, что единственным различием между синдромом и числом единиц является неясная запятая в регистре (счетчике) синдрома. Если она считается стоящей слева от числа, находящегося в регистре

синдрома, то число является синдромом, если справа — то числом единиц.

Для уменьшения длины тестов, равной 2^n , комбинационная схема разбивается на подсхемы, спроектированные таким образом, что каждая из них проверяется по своему синдрому.

Для реализации метода синдромного диагностирования комбинационные схемы должны проектироваться таким образом, чтобы синдром исправной схемы отличался от синдрома неисправной.

Глава десятая

ДИАГНОСТИРОВАНИЕ ПЕРИФЕРИЙНЫХ УСТРОЙСТВ

10.1. СПОСОБЫ ДИАГНОСТИРОВАНИЯ ПЕРИФЕРИЙНЫХ УСТРОЙСТВ

Большинство периферийных устройств (ПУ) представляет собой электромеханические устройства, надежность которых значительно ниже надежности центрального процессора.

В электромеханических ПУ большинство отказов приходится на долю их электромеханических частей. Локализация неисправностей логической части ПУ и их управления не представляет затруднений, основана на использовании традиционных методов диагностирования дискретных объектов и может быть автоматизирована. Локализация неисправностей электромеханических узлов, основанная на методах определения работоспособности непрерывных объектов, с трудом поддается автоматизации и в значительной мере остается ручной процедурой, заключающейся в наблюдении сигналов с помощью осциллографа и сравнении их с эталонными.

Часто под ПУ понимают совокупность собственно периферийного аппарата и его блока управления (УПУ), тем более что в ряде случаев ПУ и УПУ представляют собой одно целое в логическом и конструктивном отношении. Собственно ПУ (периферийный аппарат) — это механизм или электронное устройство, непосредственно выполняющее основные функции по вводу или выводу информации с помощью различных носителей информации, преобразование физических сигналов или действий в цифровой код и обратно (лентопротяжный механизм, печатающий механизм, клавиатура пишущей машинки, аналого-цифровой преобразователь и т. д.).

Электронный блок управления периферийным аппаратом выполняет специфическую для данного ПУ функцию преобразования информации и управления механизмами аппарата.

На рис. 10.1 приведена общая схема подключения УПУ/ПУ к ЭВМ, в которой можно выделить два типа интерфейсов: периферийных устройств (интерфейс ввода-вывода) и периферийных аппаратов (малый интерфейс).

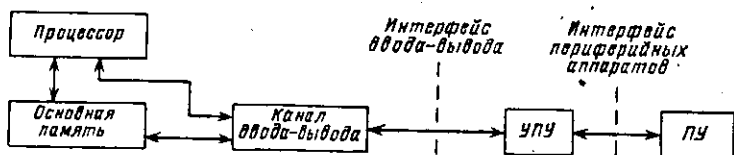


Рис. 10.1. Общая схема подключения УПУ/ПУ к ЭВМ

Интерфейс ввода-вывода стандартизован, а интерфейс периферийных аппаратов нестандартизован, так как сами периферийные аппараты весьма разнообразны по принципу действия, по выполняемым приказам и операциям и по используемым форматам информации и сигналам.

Через интерфейс ввода-вывода от канала ввода-вывода к УПУ передаются байты данных, признаки и номер (адрес) ПУ. Передача осуществляется с помощью сигналов, которые называются *сигналами канала*. От УПУ в канал ввода-вывода поступают вводимые байты данных, номер (адрес) ПУ, байты основного и уточненного состояний. Передача осуществляется с помощью сигналов, которые называются *сигналами абонента*.

Диагностирование УПУ/ПУ может осуществляться либо во время работы с каналами ввода-вывода ЭВМ, либо путем имитации этих режимов. Во втором случае средства диагностирования имитируют сигналы канала ввода-вывода и анализируют полученные от диагностируемого устройства сигналы абонента.

В зависимости от средств, используемых для диагностирования, различают следующие способы диагностирования УПУ/ПУ:

- с помощью процессора;
- с помощью тестеров;
- с помощью имитаторов канала;
- с помощью встроенных в УПУ средств.

10.2. ДИАГНОСТИРОВАНИЕ УПУ/ПУ С ПОМОЩЬЮ ПРОЦЕССОРА

При этом способе диагностирования (рис. 10.2) тестовые воздействия в УПУ/ПУ подает процессор/канал и результаты элементарных проверок через канал постунают в процессор. Воздействиями элементарных проверок являются приказы и данные, а результатами элементарных проверок — данные, байты основного и уточненного состояний [24].

Для управления вводом-выводом в ЕС ЭВМ I очереди используются следующие команды: НАЧАТЬ ВВОД-ВЫВОД, ПРОВЕРИТЬ ВВОД-ВЫВОД, ПРОВЕРИТЬ КАНАЛ, ОСТАНОВИТЬ ВВОД-ВЫВОД.

В ЕС ЭВМ II очереди помимо указанных выше первых трех команд имеются следующие команды: НАЧАТЬ ВВОД-ВЫВОД С БЫСТРЫМ ОТКЛЮЧЕНИЕМ, ОСВОБОДИТЬ ВВОД-ВЫВОД, ОСТАНОВИТЬ УСТРОЙСТВО, ЗАПИСАТЬ ИДЕНТИФИКАТОР КАНАЛА.

Команда НАЧАТЬ ВВОД-ВЫВОД инициирует операцию ввода-вывода. Операция, выполняемая УПУ/ПУ, задается в коде операции (приказе) управляющего слова канала (УСК). В результате выполнения операции канал записывает слово состояния канала (ССК), содержащее информацию о состоянии канала и ПУ. Основная информация о состоянии устройства в виде байта состояния автоматически посылается устройством в канал при пуске операции ввода-вывода, ее окончании или при возникновении каких-либо ненормальных ситуаций в работе ПУ, а также по приказу, содержащемуся в УСК.

Форматы команд ввода-вывода, адресного слова канала, слова состояния канала и управляющего слова канала в ЕС ЭВМ показаны на рис. 10.3.

При обнаружении ошибок во время операций ввода-вывода в байте состояния записываются соответствующие признаки. Разряды байта состояния характеризуют занятость устройств ранее начатой операцией, окончание операции, обнаружение ошибки в работе устройства, обнаружение определенных особых условий. Значения отдельных разрядов (указателей) байта состояния приведены ниже.

Номер разряда		Указатель
К		Контрольный разряд
0		Внимание
1		Модификатор
2		УПУ кончило
3		Занято
4		Канал кончил
5		Периферийное устройство кончило
6		Сбой в устройстве
7		Особый случай

Указатели формируются ПУ или УПУ.

Указатель «Сбой в устройстве» устанавливается при различных ошибках в ПУ и УПУ.



Рис. 10.2. Диагностирование УПУ/ПУ с помощью процессора

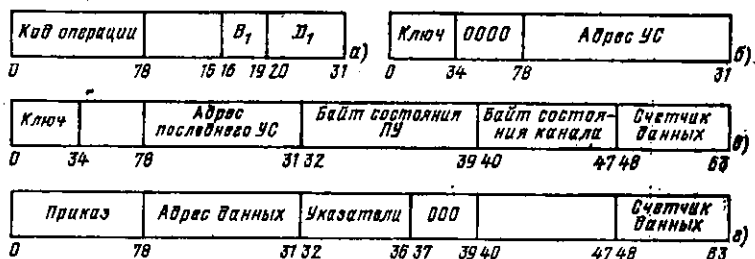


Рис. 10.3. Форматы команд ввода-вывода (а), адресного слова канала (б), слова состояния канала (в), управляющего слова канала (г)

Более подробную информацию о состоянии ПУ содержит так называемый код уточненного состояния, длина и объем которого зависят от типа ПУ. Для получения кода уточненного состояния используется специальный приказ «уточнить состояние». Число передаваемых при этом байт и значение информации зависит от типа ПУ. Собственно информация уточненного состояния обычно содержится в первых двух байтах, а в остальных байтах находится диагностическая информация. Независимо от типа ПУ в первых шести разрядах начального (нулевого) байта расположены указатели, приведенные в табл. 10.1.

Повышение глубины поиска дефекта в современных УПУ/ПУ достигается увеличением количества диагностической информации о состоянии УПУ/ПУ в коде уточненного состояния, а также добавлением специальных диагностических приказов, таких, как например, *диагностическая запись* и *диагностическое чтение*, конкретное содержание которых зависит от типа ПУ.

Диагностирование УПУ/ПУ в составе ЭВМ выполняется путем запуска операций ввода-вывода и анализа их результатов. Эти функции выполняют программные средства тестового диагностирования. В ЕС ЭВМ имеются два комплекта тестовых программ, выполняемых соответственно во время профилактических и ремонтных работ и в мультипрограммном режиме параллельно с решением задач пользователя под управлением ОС ЕС. Первый комплект называется комплектом

Таблица 10.1

Номер разряда	Указатель	Условия установки указателя
0	Команда отвергнута	Периферийное устройство не может выполнить приказ (приказ запрещен для данного ПУ, неправильная последовательность приказов, оборудование, выполняющее приказ, не включено и т. п.)
1	Требуется вмешательство	Необходимы действия оператора (замятие перфокарты, нет бумаги в ПУ и т. п.)
2	Ошибка на ШИН-К	Периферийное устройство получило из канала байт с неправильным контрольным разрядом
3	Сбой в оборудовании	Сбой в логических схемах ПУ или УПУ
4	Ошибка в данных	Ошибка в данных на носителе ПУ
5	Переполнение	Канал не успел обслужить ПУ за отведенное время (пропускная способность памяти или канала меньше общей скорости выполняемых операций ввода-вывода)

программ технического обслуживания (КПТО), а второй — комплексом неавтономных тестов устройств (КНТУ).

Комплекты программ технического обслуживания и неавтономных тестов устройств, их состав, назначение и процедуры использования рассмотрены в гл. 16.

10.3. ПРИНЦИП ПРОВЕРКИ УПУ/ПУ С ПОМОЩЬЮ ДИАГНОСТИЧЕСКИХ ПРИКАЗОВ

Как отмечалось ранее, конкретная реализация диагностических приказов зависит от типа УПУ/ПУ. Вместе с тем в ряде устройств используется универсальный принцип проверки УПУ/ПУ с помощью диагностических приказов. Этот принцип основан на том, что исполнительные (недиагностические) приказы ПУ можно представить в виде совокупности функций, выполняемых различным оборудованием УПУ/ПУ и зависящих от параметров приказа. Кроме того, благодаря специальным диагностическим признакам, устанавливаемым с помощью приказа *диагностическая запись*, появляется возможность запрещать (разрешать) выполнение тех или иных функций. И поскольку та или иная функция выполняется определенным оборудованием, появляется возможность управлять количеством работающего оборудования и фиксировать последовательность выполнения функций.

Каждой функции, совокупность которых определяет ход выполнения приказа, ставится в соответствие характерный для данного этапа

сигнал, отсутствие которого говорит о выполнении (невыполнении) этапа (функции) и неисправности определенного оборудования.

Эти сигналы устанавливают соответствующие биты диагностического регистра. Совокупность функций и соответствующая ей последовательность диагностических бит, установившихся при выполнении приказа, называются *трассой*. *Трасса* определяется параметрами исполнительного приказа и диагностическими признаками. Сочетание последних задает ход выполнения приказа и позволяет предсказать конечный результат (трассу) при нормальном выполнении приказа.

Диагностическая процедура выполняется следующим образом:

по приказу *диагностическая запись*, устанавливается диагностический режим, который запрещает/разрешает выполнение определенных функций с помощью последующего исполнительного приказа. Таким способом задается количество работающего оборудования и фиксируется трасса выполнения исполнительной команды;

выполняется исполнительный приказ, и его трасса накапливается в диагностическом регистре;

по окончании выполнения приказа содержимое диагностического регистра считывается с помощью приказа *диагностическое чтение* и передается в процессор;

диагностическая программа производит сравнение полученной трассы с ожидаемой. При успешном выполнении диагностической процедуры (трассы сравнились) выполняется очередная процедура с расширением функций исполнительных приказов (увеличение количества работающего оборудования) по отношению к предыдущей, успешно выполненной процедуре. Таким образом производится последовательное включение оборудования с проверкой его работоспособности. Если результат сравнения трасс отрицателен (трассы не сравнились), программа производит анализ полученной трассы для определения неисполненной функции и ставит ей в соответствие согласно словарю неисправностей определенное оборудование, расположенное в конкретных ТЭЗ.

Описанный принцип диагностирования используется в устройстве ввода с перфокарт и устройстве управления дисплеями.

В устройстве ввода-вывода перфокарт ЕС-6015 используется только один диагностический приказ, а именно *диагностическое чтение*, т. е. совокупность выполняемых устройством функций может быть задана только параметрами исполнительного приказа. В результате выполнения заданного из ЭВМ приказа получается последовательность диагностических бит (трасса), которая опрашивается приказом *диагностическое чтение*, передается в ЭВМ и сравнивается с эталонной трассой [52].

В устройстве управления дисплеями имеется больше возможностей управления совокупностью выполняемых функций благодаря наличию регистра диагностических приказов, устанавливаемого приказом

Диагностическая запись. Этот регистр позволяет управлять аппаратурой и улучшить локализацию неисправностей.

В остальном диагностирование выполняется так же, как и в предыдущем устройстве.

10.4. ДИАГНОСТИРОВАНИЕ УПУ/ПУ С ПОМОЩЬЮ ТЕСТЕРОВ

Тестеры периферийных устройств представляют собой ручные или полуавтоматические устройства, позволяющие диагностировать УПУ/ПУ в автономном режиме, т. е. вне ЭВМ.

Существуют два типа тестеров: для диагностирования совмещенных УПУ/ПУ и для диагностирования ПУ.

Обобщенные схемы соединения таких тестеров с устройством показаны на рис. 10.4, а и б соответственно.

Тестеры позволяют выполнить автономные тестовые процедуры для проверки УПУ/ПУ, индцировать результаты проверки на своем пульте, создать режимы заикливания тестовых процедур для наблюдения временной диаграммы с помощью осциллографа. Тестовые процедуры задают последовательность действий, которую необходимо выполнить на тестере и в устройстве для получения реакции устройства на эти действия, а также указывают состояние ламп индикации тестера. На основании анализа реакции устройства, сравнения состояния ламп индикации, полученного и заданного тестовой процедурой, делается заключение о неисправности устройства.

Совмещенные УПУ/ПУ обычно содержат блоки сопряжения с каналом (БСК) и управления механизмом — адаптер и сам механизм. Соответственно тестовые процедуры делятся на процедуры проверки функционирования БСК, адаптера и механизма.

Примером тестера УПУ/ПУ является тестер, предназначенный для диагностирования устройств ввода с перфокарт, ввода с перфоленты, вывода на перфоленту. Тестер обеспечивает проверку БСК, адаптера и механизма считывания. Проверка БСК проводится в шаговом режиме. Все последовательности сигналов интерфейса, задаваемые с тумблеров, подаются в блок сопряжения непосредственно через усилители приемника устройства, т. е. достигается полная проверка сопряжения. При этом на тестере индицируются сигналы интерфейса и состояние триггеров, что позволяет легко определить неисправность.

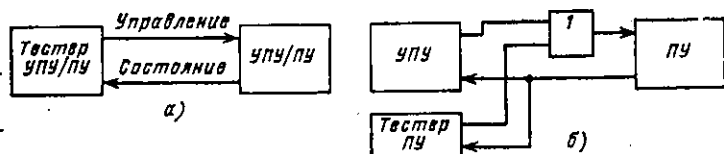


Рис. 10.4. Схема соединения тестера

Существуют следующие виды проверок адаптера и механизма считывания: проверка правильности считывания информации путем использования специальных макетов перфокарт, контроль кодопреобразования, проверка реакции схем контроля на неисправности.

Примером тестера периферийных устройств является тестер накопителя на сменных магнитных дисках (НСМД) ЕС-5061. Этот тестер обеспечивает проверку НСМД ЕМ-5061 с помощью следующих операций:

- позиционирования головок накопителя между двумя адресами, предварительно заданными оператором;

- последовательного позиционирования головок путем приращения адреса цилиндра от нулевого до максимального и возврата снова к нулевому адресу;

- последовательного позиционирования головок от максимального адреса до нулевого и возврата снова к максимальному адресу;

- позиционирования головок накопителя по случайным адресам;

- записи нулей;

- записи единиц;

- чтения информации с пакета дисков;

- индикации состояния интерфейсных линий;

- сброса регистра головок;

- приращения регистра головок.

Каждая операция может быть выполнена однократно и многократно. Скорость повторения операции может регулироваться потенциометром «время цикла». В документации тестера приводятся временные диаграммы наиболее важных сигналов накопителя при выполнении различных операций.

10.5. ДИАГНОСТИРОВАНИЕ УПУ/ПУ С ПОМОЩЬЮ ИМИТАТОРОВ КАНАЛА

Имитаторы канала (ИК) представляют собой автономные устройства, имитирующие различные режимы работы каналов ввода-вывода для полной проверки УПУ/ПУ.

Наиболее совершенные имитаторы канала представляют собой программируемые устройства с возможностью загрузки программы диагностирования конкретного УПУ/ПУ в оперативную память имитатора, выполнение этой программы с помощью блока управления.

В качестве примера рассмотрим ИК, изображенный на рис. 10.5 [23].

Имитатор канала подключается к УПУ/ПУ через блок стандартного сопряжения. Блок управления формирует сигнал управления в соответствии с заданным режимом работы. Блок формирования данных предназначен для генерации тестового кода для передачи в ПУ.

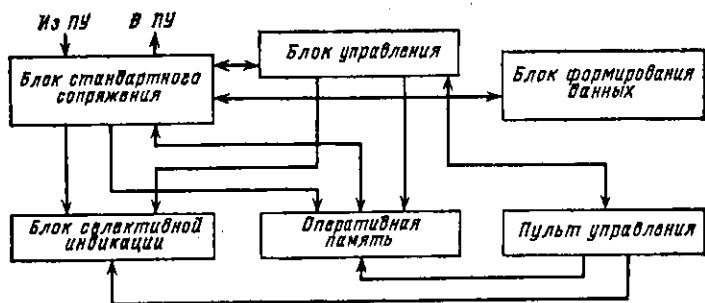


Рис. 10.5. Структурная схема имитатора канала

После загрузки в оперативную память ИК программы диагностирования ПУ и задания с пульта управления ИК режима работы начинается проверка ПУ. Блок управления выбирает очередной приказ программы и вводит в ПУ последовательность начальной выборки, в процессе которой из ИК в ПУ поступает адрес ПУ и код приказа. В ответ ПУ передает в ИК адрес, который сравнивается с заданным.

При совпадении адресов ИК передает в ПУ код приказа, на который оно отвечает байтом состояния. Если байт состояния ПУ не содержит указателей сбоя, ИК переходит к передаче или приему данных. При передаче данных последние поступают в ПУ через блок стандартного сопряжения либо из блока формирования данных, либо из памяти чисел. После каждого переданного и принятого байта модифицируется состояние внутреннего счетчика и сравнивается со значением счетчика байт, заданным в управляющем слове. При приеме каждый байт контролируется по модулю 2. После передачи последнего байта ИК вводит в ПУ последовательность сигналов окончания операции и, получив конечный байт состояния, заканчивает выполнение приказа.

Для осциллографирования неисправностей в ИК предусматривают режим циклического перезапуска по сигналу ошибки. Удобным средством является также блок селективной индикации, который позволяет индицировать на пульте произвольный байт любого вида (байт адреса, состояния или данных). Для этого с пульта задается число и вид байта. На вход счетчика, входящего в блок селективной индикации, поступает сигнал, идентифицирующий адрес, байт состояния или байт данных, поступающие от ПУ. Когда значение счетчика совпадает с кодом, заданным с пульта, схема сравнения формирует сигнал, разрешающий запись выбранного байта в регистр индикации.

10.6. ДИАГНОСТИРОВАНИЕ УПУ/ПУ С ПОМОЩЬЮ ВСТРОЕННЫХ В УПУ СРЕДСТВ

Для ПУ типа накопителей на магнитных дисках и магнитных лентах устройства управления представляют собой сложные устройства, приближающиеся по своим возможностям к универсальным процессорам. В этих случаях они имеют встроенные средства тестового диагностирования, обеспечивающие диагностирование как самого устройства управления, так и периферийных устройств. В микропрограммных УПУ используется встроенная или загружаемая микродиагностика.

В качестве примера рассмотрим встроенные средства тестового диагностирования устройства управления НМД. Микродиагностика этого устройства обеспечивает проверку арифметическо-логического блока, общих регистров, регистра адреса микрокоманд, регистров канального переключателя и другой аппаратуры.

При обнаружении неисправностей на пульте управления устройства индицируются коды ошибок, по которым в инструкции по эксплуатации находится идентификатор листа микропрограммной логики (алгоритма микродиагностики), где поясняется тип неисправности.

Некоторые устройства управления содержат встроенные имитаторы каналов и имитаторы ПУ, позволяющие максимально полно проверить УПУ в автономном режиме. Таким устройством, в частности, является мультитексор передачи данных, средства диагностирования которого описаны в гл. 11.

Микродиагностика НМД, выполняемая с помощью встроенных в устройство управления средств, позволяет проверить следующие средства и режимы работы НМД:

- регистр адреса цилиндра, состояние накопителя, частота вращения диска накопителя;

- возврат к нулевому цилиндру, установку на заданный цилиндр по набранному на переключателях коду. Проверка обеих операций выполняется путем чтения и сравнения собственного адреса нулевой головки;

- установку между двумя цилиндрами. Каждое выполнение установки проверяется путем считывания и сравнения собственного адреса нулевой головки;

- последовательную установку на каждый из цилиндров пакета дисков. Процедура начинает выполняться с произвольного адреса цилиндра. Адрес увеличивается на единицу до достижения максимального номера цилиндра. После этого адрес уменьшается на единицу до достижения нулевого адреса;

- последовательную установку на каждый из цилиндров с проверкой каждой установки выбранной головкой путем сравнения собственного адреса;



Рис. 10.6. Параллельная работа УУ НМД и блока диагностирования НМД

установку на различные цилиндры. После каждой установки считывается собственный адрес;

проверку возможности накопителя считывать предварительно сделанную запись определенной длины. При этом выполняется установка на цилиндр 001, сравнивается собственный адрес и считывается RO. Может быть задана однокорректная и многокорректная запись. При задании с пульта многокорректной записи считывание производится всеми дорожками, начиная с 00 и кончая 19. Однокорректная запись производится заданной с пульта головкой;

проверку регулировки головки (заданной с пульта) на конкретном цилиндре. Используется для обнаружения неисправности при регулировке.

Поиск неисправностей с помощью указанных микропрограммных тестов ведется по приведенным в документации временным диаграммам.

Обычно встроенные средства тестового диагностирования УПУ выполняют диагностирование присоединенных ПУ в автономном режиме, т. е. УПУ монополизировано средствами тестового диагностирования и не может выполнять каких-либо других операций с системой.

Для устройств управления НМД и НМЛ такой режим нежелателен, так как он фактически приводит к останову ЭВМ. Для устранения этого недостатка встроенные в устройство управления средства диагностирования разрабатываются так, чтобы они могли использовать общие шины интерфейса периферийного аппарата в режиме разделения времени. При таком решении устройство управления может выполнять приказы процессора, работая с исправными устройствами, а блок диагностирования (БД) — проверку неисправного устройства в режиме разделения времени в периоды незанятости устройства управления. Благодаря этому обслуживающий персонал может диагностировать и устранять неисправности устройства без прерывания нормальной работы ЭВМ. На рис. 10.6 показано устройство управления НМД (УУ НМД) с встроенным блоком диагностирования. Адаптер НМД декодирует и выполняет приказы, преобразование параллельного кода в последовательный и наоборот, обеспечивает формирование стандартного формата для записи на диск. Входные шины адаптера используются как самим устройством управления, так и блоком диагностирования.

Поскольку интерфейс УУ — адаптер используется двумя блоками, устанавливается приоритет обращения к адаптеру. До начала использования интерфейса блок диагностирования запрашивает устройство управления о возможности использования интерфейса. Для обеспечения совместной работы вводятся две дополнительные линии, осуществляющие взаимную блокировку доступа УУ и БД к общему тракту. Линия от БД к УУ активизируется, когда БД запрашивает использование интерфейса. Устройство управления опознает запрос и после достижения соответствующей точки выполнения приказа разрешает БД захватить управление интерфейсом. Активизация линии от УУ к БД означает, что УУ работает с НМД. Блок диагностирования не может прервать работу УУ, пока эта линия активна.

Конфликтная ситуация может возникнуть, когда УУ и БД одновременно пытаются получить доступ к интерфейсу. Для предотвращения подобной ситуации БД после получения разрешения на доступ к интерфейсу начинает работу только после определенной паузы. Если в течение этой паузы по линии УУ—БД поступил сигнал, БД не начнет работу, пока снова не представится такая возможность. После запуска теста БД не прервет работу до его окончания.

Глава одиннадцатая

ДИАГНОСТИРОВАНИЕ СРЕДСТВ ТЕЛЕОБРАБОТКИ ДАННЫХ

11.1. АППАРАТУРНЫЕ СРЕДСТВА ТЕЛЕОБРАБОТКИ ДАННЫХ

Развитие вычислительной техники привело к интеграции ее со средствами связи. Все большее распространение получают распределенные вычислительные системы, системы телеобработки, как одомашинные, так и сетевые, в которых пользователь имеет доступ к сети ЭВМ. Аппаратурные средства телеобработки данных включают в себя мультиплексоры передачи данных (МПД), аппаратуру передачи данных (АПД), абонентские пункты (АП), телефонные и телеграфные линии связи. На рис. 11.1 приведена структурная схема типичной системы телеобработки с передачей данных по телефонному каналу связи.

Мультиплексор передачи данных предназначен для двустороннего поочередного или одновременного обмена информацией между ЭВМ и АП, а также межмашинного обмена информацией по телефонным и телеграфным каналам связи.

Аппаратура передачи данных обеспечивает сопряжение логических дискретных устройств МПД и АП с каналами

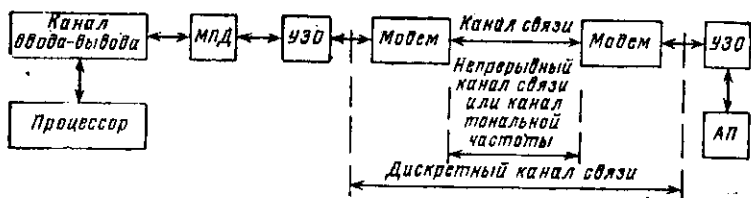


Рис. 11.1. Структурная схема системы телеобработки данных

связи. *Абонентские пункты* представляют собой одно или несколько периферийных устройств со специальным устройством управления, которое обеспечивает обмен информацией с МПД по каналу связи.

Аппаратура передачи данных включает в себя модемы, вызывные устройства (ВУ) для коммутируемых линий связи и устройства защиты от ошибок (УЗО).

Модем называют устройство для преобразования дискретного сигнала в непрерывный и наоборот в соответствии с принятым видом модуляции (амплитудной, частотной, фазовой и др.). *Вызывное устройство* предназначено для автоматического установления соединения на коммутируемых каналах связи тональной частоты и подключается к соединительной линии абонентского телефона совместно с модемом, имеющим в своем составе устройство автоматического ответа.

Устройство защиты от ошибок предназначено для обнаружения и исправления ошибок в переданной по каналу связи информации.

Канал передачи данных, включающий в себя модемы и телефонную линию, условно называют *дискретным каналом связи*. Телефонную линию называют также *непрерывным каналом связи* или *каналом тональной частоты* (ТЧ).

11.2. ДИАГНОСТИРОВАНИЕ МУЛЬТИПЛЕКСОРА ПЕРЕДАЧИ ДАННЫХ

Мультиплексор передачи данных состоит из блока сопряжения с каналом ввода-вывода (БСК), блока подсоединения адаптеров (БПА), блока управления индикацией (БУИ), блока диагностирования (БД), адаптеров данных (АД) и фактически представляет собой электронный коммутатор, поочередно подсоединяющий адаптеры данных к каналу ввода-вывода ЭВМ. *Адаптер данных* представляет собой устройство для преобразования параллельного кода в последовательный при передаче данных в модем и при приеме данных из модема.

1
Диагностирование МПД может осуществляться в двух режимах: автономном с помощью встроенных средств тестового диагностирования;

системном с помощью диагностических программ, использующих специальные диагностические приказы.

Рассмотрим особенности диагностирования МПД в автономном и системном режимах.

Диагностирование МПД в автономном режиме выполняется следующим образом.

Блок диагностирования, с помощью которого выполняется поиск неисправностей МПД в автономном режиме, формирует сигналы, имитирующие сигналы канала ввода-вывода. Эти сигналы подаются в блок сопряжения с каналом (минуя входные/выходные усилители). Блок сопряжения с каналом ввода-вывода работает с адаптером также, как при работе с ЭВМ. Однако выход адаптера в реальный канал связи в автономном режиме закрыт и заведен обратно в блок диагностирования. Блок диагностирования передает команды и данные через БСК в адаптер и одновременно контролирует ответные сигналы БСК. Информация, передаваемая адаптером в блок диагностирования, сравнивается с тестовой информацией.

Блок диагностирования МПД содержит местное устройство управления (МУУ), имитатор канала (ИК), имитатор линий связи (ИЛС) и блок анализа (БА).

Местное устройство управления обеспечивает управление блоком диагностирования. *Имитатор канала* обеспечивает выдачу приказов, прием и передачу информации, прием состояния. *Имитатор линии связи* выдает в проверяемый адаптер по цепи «информация приема» (ИНФ—ПМ) сигналы, идентичные сигналам, принимаемым адаптером из линии связи. Имитатор линии связи принимает также из адаптера информацию, поступающую по цепи «информация передачи» (ИНФ—ПД). Основной частью ИЛС является 10-битный регистр сдвига.

Блок анализа предназначен для определения правильности прохождения информации по тракту БД—БСК—адаптер—БД, а также для проверки правильности реакции БСК и адаптеров на выдаваемые блоком диагностирования сигналы. Анализ выполняется путем сравнения полученных результатов с эталонными. Для этого блок анализа содержит буферный регистр, схемы сравнения информации при записи и чтении, а также схему сравнения адреса. Информация об ошибке индицируется на пульте управления МПД.

На рис. 11.2 показана схема прохождения тестовой информации при работе МПД в автономном режиме диагностирования. Все внешние связи МПД при этом заблокированы. Тестовая информация, вырабатываемая имитатором канала, проходит по шинам канала через БСК, соответствующий адаптер и по цепи передаваемой информации

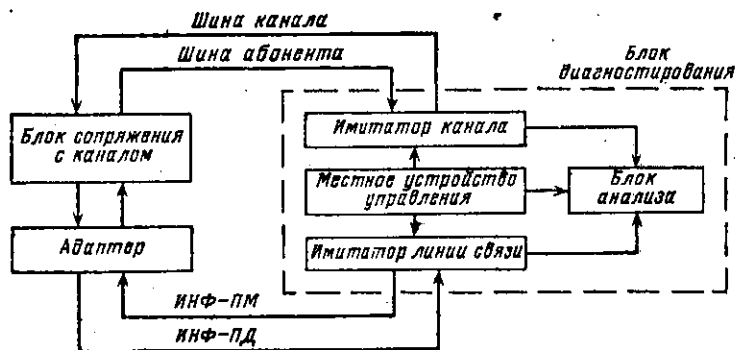


Рис. 11.2. Схема прохождения тестовой информации при автономном режиме диагностирования МПД

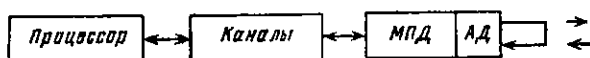


Рис. 11.3. Схема диагностирования МПД в системном режиме

(ИМФ—ПД) возвращается в ИЛС блока диагностирования, откуда поступает в БА, где сравнивается с исходной информацией (режим «Запись»). В режиме «Чтение» работа происходит аналогично, только меняется направление потока информации.

Диагностирование МПД в системном режиме выполняется с помощью диагностических программ под управлением процессора ЭВМ. С целью исключения аппаратуры передачи данных и канала связи при диагностировании самого МПД используются два специальных приказа — *диагностическая запись* и *диагностическое чтение*, выполняемые адаптером данных.

Схема диагностирования МПД в системном режиме приведена на рис. 11.3. Приказы *диагностическая запись* и *диагностическое чтение* выполняются так же, как и *запись* и *чтение*, за исключением того, что адаптер, не работает с каналом связи. Для этой цели в адаптере имеется регистр диагностирования (РгДг), представляющий собой регистр сдвига, в котором хранится последний знак, переданный в канал связи по приказу *запись* или *диагностическая запись*. При выполнении приказа *диагностическая запись* данные из адаптера поступают не в модем, а на РгДг. Если передается несколько знаков, на РгДг остается последний. При выполнении приказа *диагностическое чтение* данные в регистр данных адаптера поступают не из канала связи, а из

РгДг. После получения информации процессор сравнивает ее с эталонной и на основании этого определяет неисправность МПД и адаптера данных. На рис. 11.4 показаны регистр данных адаптера, обеспечивающий прием и передачу информации в канал связи и из канала связи, диагностический регистр и его связи с регистром данных адаптера.

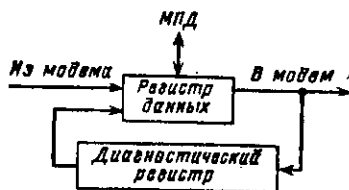


Рис. 11.4. Диагностирование адаптера данных

Другим примером диагностического приказа является приказ *писать по шлейфу*, используемый в мультимплексоре МПД-2. При выполнении этого приказа информация, выдаваемая по какому-либо подканалу ввода-вывода, передается всегда обратно в ЭВМ по нулевому подканалу.

11.3. ДИАГНОСТИРОВАНИЕ КАНАЛА ПЕРЕДАЧИ ДАННЫХ

Традиционным методом проверки аппаратуры передачи данных, дискретного канала связи и канала ТЧ является проверка с созданием шлейфового режима. *Шлейф* представляет собой перемычку для создания обратной связи для возврата сигнала к месту его передачи. Шлейфы бывают следующих типов:

- дискретного канала и непрерывного канала;*
- местный и удаленный;*
- с местным или дистанционным управлением;*
- одно- и двунаправленный.*

На рис. 11.5 показаны местный и удаленный шлейфы. Следует отметить, что местный шлейф может создаваться дистанционным управлением и наоборот. Местный шлейф позволяет по внутреннему тракту аппарата подать передаваемую информацию и тактовый сигнал передатчика соответственно на информационный и тактовый входы при-

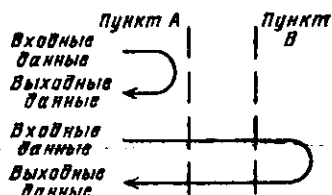


Рис. 11.5. Местный и удаленный шлейфы

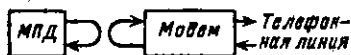


Рис. 11.6. Двунаправленный шлейф дискретного канала

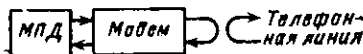


Рис. 11.7. Двунаправленный шлейф непрерывного канала

емника. Удаленный шлейф позволяет проверить всю линию. В каждом случае имеющаяся в источнике данных программа позволяет сравнить принятые символы с символами, которые были переданы источником, чтобы проверить правильность работы тракта.

На рис. 11.6 показан двунаправленный шлейф дискретного канала. Шлейф является дискретным, так как он образован на дискретной стороне модема, и двунаправленным, так как в месте разрыва цепи сформированы два противоположно направленных шлейфа, причем, как видно из рисунка, левый шлейф является местным, а правый — удаленным.

На рис. 11.7 показан шлейф непрерывного канала, также двунаправленный. Левый шлейф является местным шлейфом непрерывного канала, а правый удаленным. Обычно чаще применяются однонаправленные шлейфы. В этом случае тракт после шлейфа отключается во время тестирования [87].

Диагностирование дискретного канала связи

При возникновении каких-либо неполадок в канале связи обычно сначала выполняется тест модема с помощью встроенных в модем средств диагностирования либо с помощью специальных тестеров передачи данных (ТПД). Встроенные средства диагностирования позволяют вырабатывать сигналы от специального генератора, позволяющего вести передачу с разной скоростью, в виде серий сигналов заданной длины [2].

Тестеры передачи данных также подают на дискретный вход модема сигналы с различной скоростью и различных кодовых комбинаций, т. е. выполняют роль источника данных, подключенного к модему, а на стороне приема в роли устройства обработки данных проверяют принятые модемом данные. Тестеры передачи данных позволяют также на основании сравнения с эталонной последовательностью вести подсчет и индентифицировать число ошибочных бит или блоков.

Встроенные средства диагностирования модема имеют более ограниченные возможности по сравнению с тестерами передачи данных.

Для проверки модема в самом модеме устанавливается шлейф, и выходные данные модема поступают на его же вход по шлейфу непрерывного канала, имитирующего телефонную линию, как показано на рис. 11.8.

Следует, однако, отметить, что такая схема не обеспечивает полноты проверки, так как тестовые наборы, как правило, не обеспечивают 100 %-ной проверки схем модема, а имитатор линии не может имитировать реальные условия линии связи и в большинстве случаев представляет собой аттенюатор, хотя иногда он имитирует и частотные характеристики линии.

Следующим этапом проверки канала связи является проверка работы модема с реальной линией (рис. 11.9). При этом в модеме на

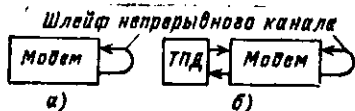


Рис. 11.8. Схема проверки модема:

а — с помощью встроенных средств; б — с помощью ТПД

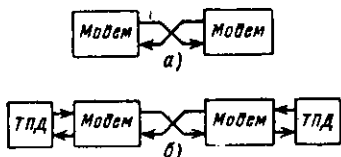


Рис. 11.10. Схема проверки модемов с линией связи в обоих направлениях без шлейфового режима:

а — с помощью встроенных средств; б — с помощью ТПД

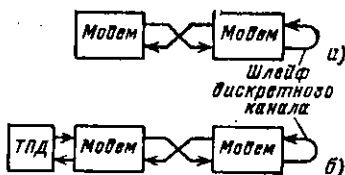


Рис. 11.9. Схема проверки модема с линией связи:

а — с помощью встроенных средств; б — с помощью ТПД

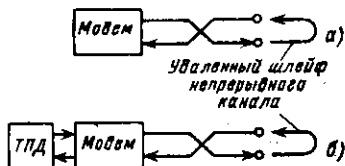


Рис. 11.11. Схема проверки модема с линией связи:

а — с помощью встроенных средств; б — с помощью ТПД

противоположном конце линии связи устанавливается удаленный шлейф дискретного канала. В этой схеме тестовый набор, сформированный в одном модеме, проходит через линию связи, другой модем, затем через линию связи в первый модем.

Если встроенные средства проверки модема не обнаруживают неисправности, то желательно выполнить проверку с помощью ТПД, который имеет больше возможностей для проверки.

Еще одной схемой тестирования аппаратуры передачи данных и дискретного канала связи без шлейфового режима является схема, приведенная на рис. 11.10. При этом методе тестирования на модемах концов включаются режимы генерации наборов и сравнения с эталоном. По данной схеме можно определить направление передачи с которым связаны неполадки, что может помочь в локализации неисправности линии связи.

Одной из схем тестирования дискретного канала, позволяющих исключить из тракта удаленный модем и, таким образом, сократить подозреваемое в неисправности оборудование, является удаленный шлейф непрерывного канала (рис. 11.11). Здесь, однако, следует учесть, что при использовании удаленного шлейфа дискретного канала для проверки линии связи наличие модема на конце линии позволяет восстановить форму сигналов перед их передачей обратно. При удаленном шлейфе непрерывного канала протяженность линии удваива-

ется, что может привести к такому затуханию сигналов, что сделает невозможным использование этого метода.

Если установлено, что неисправность имеет место в линии связи, для более точной локализации неисправности необходимо проверить характеристики линии связи, т. е. канала тональной частоты.

Диагностирование абонентских пунктов

Диагностирование АП может выполняться с помощью встроенных аппаратурных или микропрограммных средств тестового диагностирования или с помощью внешних средств, например тестеров АП.

Тестеры АП представляют собой устройства, позволяющие задавать программу работы и имитировать реальную работу путем задания с пульта скорости приёма и передачи данных. Тестеры АП позволяют накапливать поступающую от АП информацию и сравнивать ее с эталонной.

Тестеры АП позволяют проверить АП (рис. 11.12, а), комплекс «АП—модем» (рис. 11.12, б), комплекс «АП—модем—линия связи—модем—АП» (рис. 11.12, в), комплекс «модем—линия связи—модем» со шлейфом дискретного канала на удаленной станции (рис. 11.12, г), комплекс «модем—линия связи» (рис. 11.12, д), модем по шлейфу непрерывного канала на выходе модема (рис. 11.12, е), МПД (рис.

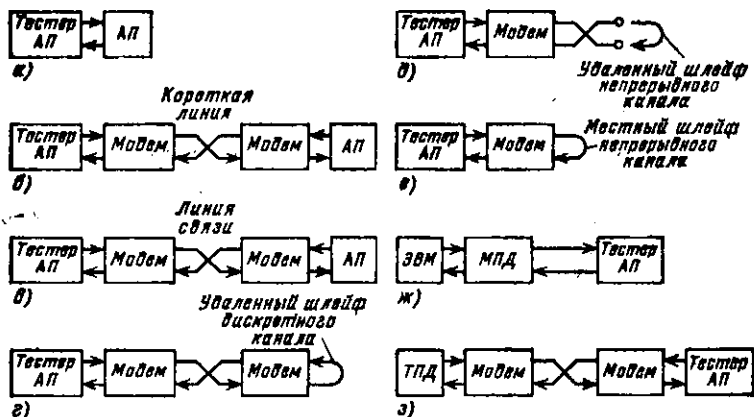


Рис. 11.12. Схемы проверки различных средств АПД с помощью тестера АП:

а — проверка комплекса «АП—модем»; б — проверка комплекса «АП—модем—линия связи—модем—АП»; в — проверка комплекса «модем—линия связи—модем»; г — проверка комплекса «модем—линия связи—модем» со шлейфом дискретного канала на удаленной станции; д — проверка комплекса «модем—линия связи»; е — проверка модема по шлейфу непрерывного канала на его выходе; ж — проверка МПД; з — совместная проверка комплекса «ТПД—модем—линия связи—тестер АП»

11.12, ж). Дополнительные возможности проверки дает совместное использование устройств проверки АП и ТПД (рис. 11.12, з). Первый из них выполняет роль генератора данных, а второй — анализатора ошибок.

11.4. ПРОВЕРКА КАНАЛА ТОНАЛЬНОЙ ЧАСТОТЫ

Техническое состояние канала ТЧ определяется в основном следующими характеристиками [87]: затуханием, частотной характеристикой, уровнем помех.

При передаче сигналов по каналам ТЧ соотношения между мощностями, напряжениями и токами электрических сигналов измеряются в децибелах. Ниже приводится математическое определение уровней мощности, напряжения и тока в децибелах:

$$N_P = 10 \lg (P_1/P_2); \quad N_U = 20 \lg (U_1/U_2); \quad N_I = 20 \lg (I_1/I_2),$$

где P_1, U_1, I_1 — параметры на входе; P_2, U_2, I_2 — параметры на выходе линии. В технике проводной связи за абсолютный нулевой (начальный) электрический уровень принята активная мощность синусоидального сигнала P_0 , равная 1 мВт и выделяемая на сопротивлении, равном 600 Ом [9].

Одним из параметров канала является затухание сигнала в децибелах, которое определяется путем замера уровней сигнала эталонной частоты на выходе модема передающей стороны и на входе модема принимающей стороны, как показано на рис. 11.13. Для замера затухания в линию выдается сигнал мощностью 1 мВт на эталонной частоте.

Измерение уровней сигнала на передающей и приемной сторонах производят измерителем уровня (ИУ) [9].

При затухании в линии, находящемся в пределах нормы, проверяются другие характеристики линии. Если затухание превышает допустимую норму, то выполняется поиск участка тракта, в котором затухание недопустимо велико, и устраняется вызывающая его причина.

Другим параметром канала ТЧ, который необходимо проверить при нарушении его работоспособности, является частотная характеристика. Канал ТЧ должен иметь частотную характеристику, удовлетворяющую определенным стандартам, которые задают максимально допустимую разницу (в децибелах) между затуханием на эталонной и других частотах. Частотную характеристику строят, откладывая по оси ординат в децибелах абсолютное затухание сигнала или относительное к затуханию на эталонной частоте, а по оси абсцисс — частоту. На

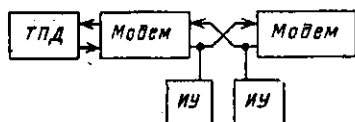


Рис. 11.13. Определение затухания сигнала

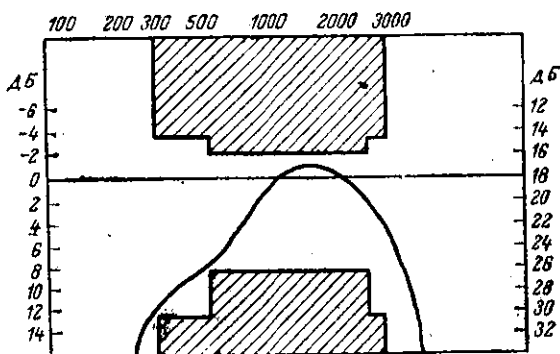


Рис. 11.14. Частотная характеристика линии связи (в допустимых пределах)

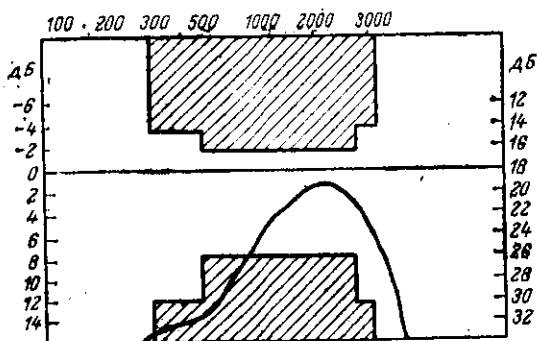


Рис. 11.15. Частотная характеристика линии связи (вне допустимых пределов)

рис. 11.14 приведена типичная частотная характеристика канала ТЧ (заштрихованы области недопустимых значений). Частотная характеристика, показанная на рис. 11.15, говорит о наличии неисправности. На ординате справа показано абсолютное затухание сигнала в децибелах, а на ординате слева — относительное к затуханию на эталонной частоте. Сигналы на частотах менее 300 или более 3400 Гц затухают настолько сильно, что непригодны для передачи. Поэтому считается, что полоса пропускания канала ТЧ равна 3100 Гц [43].

Измерения частотной характеристики выполняются специальной аппаратурой, которая позволяет изменять частоту передаваемого сигнала и измерять его затухание на приемном конце линии.

Частотная характеристика может быть получена по точкам с помощью звукового генератора и измерителя уровня с последующим подсчетом затухания на каждой частоте или визуальным наблюдением

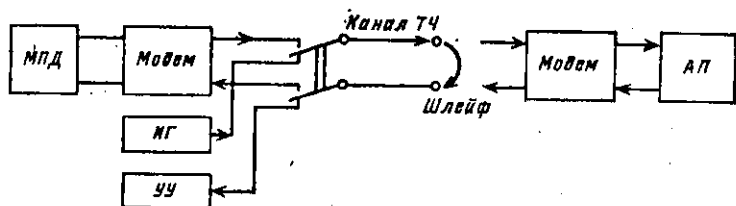


Рис. 11.16. Определение остаточного затухания линии связи в диапазоне тональных частот:

ИГ — импульсный генератор; УУ — устройство управления

формы характеристики на экране ЭЛТ измерителя частотных характеристик. Первый метод достаточно прост, но трудоемок, а второй позволяет только качественно оценить частотную характеристику канала ТЧ.

Иногда частотная характеристика снимается с помощью прибора, устанавливаемого на передающем конце канала ТЧ [9] и содержащего измерительный генератор и указатель уровня. Этот прибор измеряет остаточное затухание канала при фиксированных частотах диапазона 300—3400 Гц. С этой целью организуется шлейф на приемном конце канала ТЧ. Измерительный генератор включается в цепь передачи, а указатель уровня — в цепь приема, как показано на рис. 11.16. Шлейф канала ТЧ (шлейф непрерывного канала) может быть включен следующим образом:

а) под местным или дистанционным управлением схемами модема. Шлейфы непрерывного канала могут включать или не включать в себя усилители. В последнем случае шлейф непрерывного канала можно сравнить с двойной линией. И если затухание в один конец равно 18 дБ, то при шлейфовом режиме оно будет равно 36 дБ. Такой шлейф обычно используют для проверки обрывов линии. Если шлейф включает усилитель, который настраивается на 16 дБ, то обратный сигнал будет иметь затухание 20 дБ ($18 - 16 + 18$);

б) в телефонном центре. Такой шлейф называется центральным. Различают три типа центральных шлейфов, которые, как правило, включают в себя усилитель на 16 дБ [87]:

ручной, включаемый с помощью тумблера;

дистанционно управляемый, управляемый из ближайшего телефонного центра с помощью сигнала постоянного тока;

дистанционно управляемый, управляемый сигналом специальной частоты.

Поскольку обычно параметры линии связи задаются для одного конца линии, с помощью измерений со шлейфом непрерывного канала трудно определить несоответствие параметров. Например, если измерение уровня со шлейфом непрерывного канала показало, что затухание

превышает номинальное на 6 дБ, то это может означать увеличение затухания на 3 дБ в каждом направлении (что находится в пределах допустимых значений) или на 5 и 1 дБ (что находится вне допустимых пределов).

Шлейф непрерывного канала в основном используют при сравнении параметров линии с эталонными характеристиками линий, замеренными при ее вводе в эксплуатацию. И если в режиме шлейфа непрерывного канала результаты не совпадут с эталонными, это будет означать, что параметры линии изменились. Если линия неработоспособна, а измерения в режиме шлейфа непрерывного канала дают совпадение с эталонными значениями, то измерения производятся в нормальном режиме без шлейфа на приемном конце линии и сравниваются с эталонными.

Причиной ухудшения передачи может быть также шум. Для измерения шумов необходимо, чтобы прибор для измерения шумов воспринимал шумы, частота которых лежит в полосе пропускания канала ТЧ, т. е. от 300 до 3400 Гц. С этой целью прибор для измерения шумов подключается к линии через фильтр 300/3400 Гц.

Замер шумов в канале ТЧ без передачи сигналов неполностью соответствует реальной картине, так как в телефонных цепях есть схемы, которые генерируют шум только в присутствии сигнала. С этой целью на противоположном конце линии генерируют сигнал, но при замере шумов этот сигнал исключают из измерения, так как в противном случае он может быть интерпретирован как шум. Это достигается с помощью специального фильтра с провалом частотной характеристики на частоте сигнала. При этом, конечно, теряются и соответствующие составляющие шума, однако их влияние на общий шум незначительно.

Глава двенадцатая

ПРИНЦИПЫ АВТОМАТИЗАЦИИ ПРОФИЛАКТИЧЕСКИХ ИСПЫТАНИЙ

12.1. ТИПЫ ПРОФИЛАКТИЧЕСКИХ ИСПЫТАНИЙ

Как отмечалось в гл. 1 и 2, целью профилактических испытаний ЭВМ является уменьшение вероятности возникновения сбоев и отказов при нормальной эксплуатации машины путем выявления и замены элементов с граничными значениями параметров. Помимо этого профилактические испытания предназначены для выявления отказов в оборудовании, не охваченном контролем или не участвующем в вычислительном процессе.

В ЭВМ широкого назначения невозможно обеспечить 100 %-ный охват оборудования контролем, так как это по-

требовало бы дублирования аппаратуры многих узлов. Кроме того, невозможно обеспечить самоконтролируемость всех схем контроля. Таким образом, могут иметь место и не обнаруживаемые во время работы ЭВМ отказы, что приводит к ухудшению достоверности вычислений.

Различают следующие типы профилактических испытаний:

- для выявления элементов с параметрами, близкими к предельным, являющихся потенциальными источниками сбоев;

- для выявления отказов в неконтролируемой части аппаратуры;

- для выявления отказов в аппаратуре ЭВМ или в отдельных устройствах, не участвующих в течение какого-то времени в вычислительном процессе.

Обычно профилактические испытания выполняются с помощью диагностических тестов с остановом процесса решения задач на машине. Для уменьшения простоя ЭВМ из-за профилактических испытаний возможны следующие решения:

- параллельное (мультипрограммное) выполнение тестов, обнаруживающих отказы в не охваченной контролем части аппаратуры;

- параллельное (мультипрограммное) выполнение тестов аппаратуры или устройств, не участвующих в вычислительном процессе;

- уменьшение времени тестирования отказов в аппаратуре, не охваченной контролем и не участвующей в вычислительном процессе, путем разработки специальных тестов, рассчитанных только на их выявление.

Поскольку основной задачей таких тестов является обнаружение отказов за минимально короткое время, эти тесты должны охватывать по возможности больший объем аппаратуры в отличие от диагностических тестов, использующих для улучшения разрешающей способности метод раскрутки, т. е. постепенного наращивания проверяемой аппаратуры.

Однако в ЭВМ всегда имеется аппаратура, которую невозможно проверить в режиме параллельного тестирования. Так, например, если схемы контроля ЭВМ не имеют свойства самоконтролируемости, то неисправности типа «постоянный логический ноль» на выходе схем контроля не обнаруживаются. Для выявления таких неисправностей схем контроля необходима автоматическая имитация ошибок и проверка реакции схем контроля. Аналогичная аппаратура

имеется и в оборудовании обслуживания ЭВМ, а также во внешних устройствах. Подобные проверки невозможно выполнять путем параллельного тестирования, так как это может привести к нарушению нормальной работы ЭВМ. Поэтому подробные проверки выполняются при останове ЭВМ на регламентные профилактические работы. Таким образом, профилактические испытания можно подразделить на два вида:

1) не требующие остановки ЭВМ. Эти испытания проводятся параллельно с решением задач;

2) требующие остановки ЭВМ. Эти испытания проводятся во время регламентных профилактических работ и монополизируют ЭВМ на время испытаний.

Регламентные профилактические работы обычно проводятся по календарному принципу. Так, например, в ЕС ЭВМ рекомендуются следующие регламентные профилактические работы: ежедневные, двухнедельные, ежемесячные, полугодовые.

При определении периода профилактических испытаний, требующих остановки ЭВМ (см. § 2.5), этот период должен согласоваться с графиком регламентных профилактических работ.

12.2. ПРОГРАММНЫЕ СРЕДСТВА ПРОФИЛАКТИЧЕСКИХ ИСПЫТАНИЙ

Программные средства профилактических испытаний делятся на следующие типы: автономные тесты; неавтономные тесты, выполняемые под управлением конкретной операционной системы; универсальные тесты, выполняемые в любой операционной среде и без нее.

Автономные тесты обычно требуют прекращения решения задач, так как они монополизируют ЭВМ на время тестирования. К автономным тестам относится, в частности, комплект программ технического обслуживания (КПТО) ЕС ЭВМ.

Неавтономные тесты позволяют выполнять параллельное тестирование устройств под управлением специальных программ операционной системы одновременно с решением задач, поэтому они иногда называются параллельными. Неавтономные тесты обычно ориентированы на работу в конкретной операционной среде. К неавтономным тестам относится, в частности, комплект неавтономных тестов устройств (КНТУ) ОС ЕС.

Универсальные тесты могут выполнять параллельное тестирование в любой операционной среде и в автономном режиме благодаря наличию нескольких управляющих программ. К универсальным тестам относится, в частности, система оперативного тестирования ЕС ЭВМ.

В настоящем параграфе КПТО, КНТУ и СОТ рассматриваются только как средство для проведения профилактических испытаний. Более подробная информация об этих комплексах программ приведена в гл. 16.

Автономные тесты

Автономные тесты состоят из тестов ядра и тест-секций, выполняемых под управлением управляющей программы. *Тесты ядра* предназначены для проверки основного оборудования ЭВМ и устройств, необходимых для работы управляющей программы и загрузки тест-секций [79]. Успешное выполнение тестов ядра позволяет загрузить управляющую программу и продолжить проверку остальных функциональных средств с помощью тест-секций. Управляющая программа обеспечивает последовательную загрузку тест-секций и их выполнение. *Тест-секция* обычно состоит из заглавия и переменного числа тестовых примеров. *Тестовый пример* образует логически независимый блок и содержит префикс и собственно проверочную часть. Заглавие секции и префикс примера являются областями связи секции и тестового примера с управляющей программой.

Управляющая программа выполняет следующие функции: настраивается на возможности и объем памяти конкретной ЭВМ, готовность устройств ввода-вывода, декодирует задаваемые оператором приказы и выполняет их, производит предварительную последовательность проверок устройств, загрузку тест-секций в оперативную память, а также выполняет ряд функций, общих для всех тест-секций, например печать сообщений, преобразование кодов. Для работы управляющей программы обычно требуется определенный объем оперативной памяти, устройство ввода приказов оператора (пишущая машинка или дисплей), устройство вывода сообщений (пишущая машинка, дисплей или АЦПУ), устройство загрузки управляющей программы (накопители на магнитной ленте или магнитных дисках или устройство ввода с перфокарт).

При проверке ЭВМ сначала с помощью процедуры начальной загрузки в оперативную память загружается управляющая программа и затем по приказам оператора вводится и выполняется тест-секция. После выполнения одной

тест-секции загружается и выполняется следующая до тех пор, пока не будут выполнены все тест-секции, указанные в приказе оператора. Связь оператора с управляющей программой выполняется с помощью приказов, задаваемых оператором. Тест-секция обычно обращается к управляющей программе для вызова стандартных программ обслуживания с помощью команды **ВЫЗОВ СУПЕРВИЗОРА**.

Работа управляющей программы начинается с ее настройки с использованием данных о системной конфигурации. По этим данным управляющая программа формирует справочную таблицу тест-секций, таблицу каналов, таблицу устройств управляющей программы и таблицу устройств ЭВМ. *Справочная таблица тест-секций* содержит информацию о возможностях процессоров, объеме оперативной памяти, адресе таблиц. *Таблица каналов* содержит информацию о каналах и их особенностях. *Таблица устройств ЭВМ* описывает входящие в состав ЭВМ устройства, их адреса и характеристики. *Таблица устройств управляющей программы* содержит информацию об устройствах загрузки и ввода-вывода, предназначенных для управляющей программы.

Тест-секции могут быть как проверочными, т. е. ориентированными на проверку функций, выполняемых устройством, так и диагностическими, т. е. обеспечивающими локализацию неисправностей до группы ТЭЗ.

Диагностические тест-секции используют специальные диагностические команды, позволяющие получить доступ к минимальному объему аппаратуры для обеспечения требуемой глубины поиска дефекта.

Параллельные (неавтономные тесты)

Параллельное или оперативное тестирование означает, что система выполняет тестирующие или диагностические программы одновременно с решением задач пользователя. Это позволяет повысить эффективность использования ЭВМ, а также сократить время восстановления внешних устройств, так как возможность параллельного диагностирования неисправностей внешних устройств позволяет получить большую глубину поиска дефекта по сравнению с автономной диагностикой. В последнем случае ограниченные возможности внешних устройств не позволяют быстро локализовать их неисправность в автономном режиме. Кроме того, если неисправность находится в интерфейсе между центральным процессором и внешним устройством, она также не может быть локализована в автономном режиме.

Операционные системы обычно предусматривают наличие управляющей программы, которая позволяет тестировать внешние устройства параллельно с решением задач пользователя. При этом каждое внешнее устройство обеспечивается набором тест-секций, вызываемых по приказу оператора.

Универсальные тесты

Различие способов подключения программ к разным операционным системам приводит к необходимости иметь разные тесты для одного и того же устройства. Этот недостаток устраняется при разработке тестов по принципу независимости тестов от операционной системы. При этом один и тот же тест может выполняться как под управлением любой из операционных систем, функционирующих на ЭВМ, так и без них [20]. Этот принцип реализуется с помощью специальных административных программ, обеспечивающих программный интерфейс между тестами и операционными системами. Интерфейс между тестами и административными программами обычно унифицирован.

Подобная система тестирования в ЕС ЭВМ носит название *системы оперативного тестирования* (СОТ) ЕС ЭВМ. Она имеет три административные программы, обеспечивающие выполнение тестов в среде, создаваемой в операционных системах ДОС ЕС (АП ДОС), ОС ЕС (АП ОС), или при их отсутствии (СНАП) (рис. 12.1).

Управляющая программа СОТ вызывается либо командой START с консоли оператора ЭВМ, либо операторами языка управления заданиями. В процессе инициализации СОТ выдается сообщение, запрашивающее параметры теста, и оператор задает тестируемые устройства ввода-вывода и режимы тестирования.

Тест разбивается на несколько тест-секций. Каждая тест-секция содержит один модуль, называемый корневым, и подчиненные ему модули.

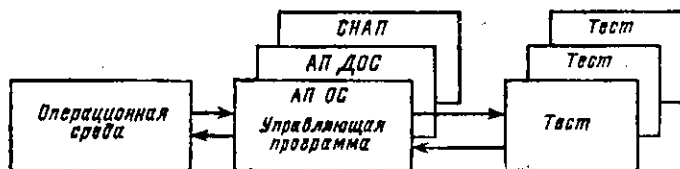


Рис. 12.1. Система оперативного тестирования

Корневой модуль называют ядром теста. В ядре концентрируются основные процедуры (например, обращение к управляющей программе, сравнение результатов с эталонами, формирование печатающей информации и т. д.).

Тест-модули содержат цепочки, сообщения тестов, формирующие эталоны, обращения к ядру.

Административные программы функционально состоят из *супервизорной* и *исполнительной* программ.

Супервизорная программа управляет работой административной программы, организует взаимодействие с оператором, планирует и запускает тесты, обрабатывает прерывание ввода-вывода и внешние прерывания, осуществляет меры по защите данных, обрабатывает аварийные ситуации, возникающие при работе теста.

Исполнительная программа осуществляет запуск и управление выполнением операций ввода-вывода на тестируемых и вспомогательных устройствах, обмен информацией с оператором, подготовку и вывод информации о результатах тестирования, управление структурой теста.

12.3. АВТОМАТИЗАЦИЯ ПРОФИЛАКТИЧЕСКИХ ИСПЫТАНИЙ С ИЗМЕНЕНИЕМ НАПРЯЖЕНИЙ ВТОРИЧНЫХ ИСТОЧНИКОВ ПИТАНИЯ

Вследствие физического старения элементов, которое может выражаться в увеличении числа сбоев, надежные характеристики ЭВМ ухудшаются. Для выявления элементов с ухудшенными параметрами с целью их замены применяют профилактические испытания (профилактический контроль) ЭВМ в утяжеленных режимах. Подобные испытания провоцируют неработоспособность элементов с ухудшенными характеристиками. Локализация и устранение путем замены таких элементов восстанавливает надежные характеристики ЭВМ.

Отказы интегральных микросхем могут быть мгновенными и постепенными. Анализ показывает, что возникновению отказа последнего типа предшествует ухудшение характеристик, проявляющееся в увеличении числа сбоев. Возможные состояния элементов в процессе эксплуатации показаны на рис. 12.2, т. е. процесс развития определенной части отказов элементов ЭВМ можно представить следующим образом: с какого-то момента времени начинается ухудшение характеристик элемента, проявляющееся в увеличении числа сбоев. С течением времени ухудшение характеристик переходит в отказ [65].

Проведение плановых профилактических испытаний в утяжеленных режимах, т. е. при профилактическом изменении напряжений вторичных источников питания, позволяет перевести логические элементы с ухудшенными параметрами в неработоспособное состояние, локализовать их и устранить путем замены на исправные элементы.

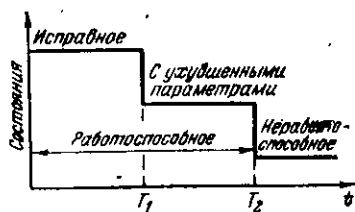


Рис. 12.2. Состояния элементов в процессе эксплуатации

Проведение таких профилактических испытаний позволяет обнаружить и устранить «будущую» неисправность во время плановых профилактических работ и таким образом предупредить возникновение отказа.

Если поток отказов имеет интенсивность λ , то после проведения профилактики возникает «разряженный» поток отказов с параметром λ_p . Это разрежение объясняется тем, что при проведении профилактических испытаний устраняются элементы с ухудшенными параметрами, появившиеся за межпрофилактический срок.

Обычно профилактические испытания целесообразно применять в тех случаях, когда предупредительная замена элементов с ухудшенными характеристиками обходится дешевле последствий отказов [49].

Если предположить затраты, связанные с отказом, равными C_1 , а затраты, связанные с предупредительной заменой еще не отказавших элементов, равными C_2 , то профилактические испытания целесообразны, если $C_1 > C_2$. Затраты C_1 интерпретируются как среднее время потерь и восстановления при аварийном отказе, а C_2 — как среднее время локализации и замены элемента с граничными параметрами.

Эффективность профилактических испытаний для выявления элементов с граничными параметрами повышается, если тестовые программы выполняются при возможно большем числе различных сочетаний отклонений напряжений источников питания.

При ручном профилактическом контроле ограниченное время регламентных работ не позволяет проверить аппаратуру на большом числе сочетаний.

Автоматизация профилактического контроля напряжений питания позволяет увеличить число проверяемых сочетаний и тем самым повысить эффективность профилактики.

манды — через регистр диагностики в регистр микрокоманд ЭВМ. Диагностические операции обеспечивают сравнение состояния регистров ЭВМ после выполнения тестовых микрокоманд с эталонными состояниями.

В конце теста микродиагностики на магнитной ленте записаны диагностические операции установки напряжений источников питания на $\pm 5\%$ выше или ниже номинального. После установки напряжений источников питания выполняется повторный прогон того же теста, но уже с измененными напряжениями источников питания. Установка напряжений источников питания выполняется путем передачи в регистр профилактического контроля кода из регистра диагностики под управлением диагностической операции «Профилактический контроль». Ниже приведено значение бита регистра профилактического контроля (РгПФК):

БИТЫ РгПФК	Назначение
0—7	Адрес ВИП
8—9	Уровень напряжения ($\pm 5\%$)
10	Разрешение профилактического контроля

Поскольку каждый тест микродиагностики выполняется при различных значениях напряжений ВИП, в регистре диагностики содержится также указатель номера очередной установки. Прием кода в регистр профилактического контроля происходит только при совпадении кода номера очередной установки со значением счетчика профилактического контроля.

Выходные сигналы регистра профилактического контроля поступают в автоматизированную систему контроля и диагностики электропитания (АС КДЭ) ЭВМ ЕС-1045, где управляют установкой профилактических уровней источников питания.

Вторым видом профилактических испытаний с изменением напряжений источников питания является выполнение тест-секций под управлением тест-монитора. В этом режиме регистр профилактического контроля устанавливается с помощью команды ДИАГНОСТИКА. Информация для профилактического контроля задается в одном из общих регистров.

Схема взаимодействия тест-секций и тест-монитора в режиме профилактических испытаний приведена на рис. 12.4.

В составе КПТО ЕС-1045 имеется тест-секция автопрофилактики, функциями которой является изменение напря-

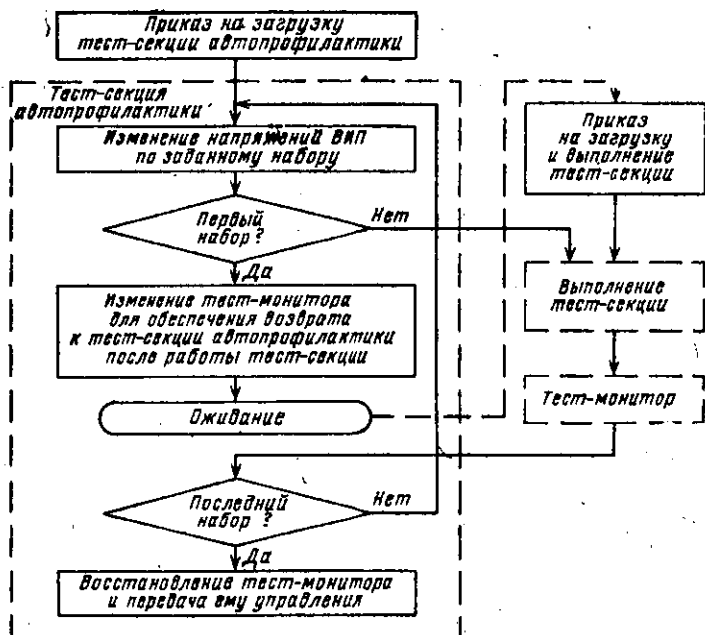


Рис. 12.4. Схема взаимодействия тест-секций и тест-монитора в режиме профилактических испытаний

жений источников питания и управление циклическим повторением каждой вызываемой тест-секции при различных сочетаниях напряжений ВИП. Тест-секция автоматического профилактического контроля состоит из секционного заглавия, подпрограммы выдачи начальных сообщений, подпрограммы изменения напряжений источников питания, подпрограммы связи и подпрограммы восстановления. Подпрограмма выдачи начальных сообщений выдает оператору сообщение о том, что можно начать выполнение теста готовности в режиме автоматического профилактического контроля. Подпрограмма изменения напряжений источников питания последовательно устанавливает требуемые отклонения напряжений источников питания. Подпрограмма связи вносит изменения в тест-монитор, обеспечивая тем самым непосредственную связь между проверяемой тест-секцией и тест-секцией автоматического профилактического контроля. Подпрограмма связи обеспечивает также загрузку проверяемой тест-секции в другую область основной па-

мента. Подпрограмма восстановления восстанавливает исходное состояние тест-монитора после выполнения тест-секцией теста готовности в режиме автоматического профилактического контроля. Тест-секция автоматического профилактического контроля обеспечивает пятикратное выполнение каждой тест-секции с различными напряжениями источников питания [42].

Глава тринадцатая

АВТОМАТИЧЕСКОЕ НАКОПЛЕНИЕ ИНФОРМАЦИИ ОБ ОШИБКАХ, ЕЕ ОБРАБОТКА И ИСПОЛЬЗОВАНИЕ

13.1. ЦЕЛЬ НАКОПЛЕНИЯ ИНФОРМАЦИИ ОБ ОШИБКАХ

Для повышения эффективности обслуживания в современных ЭВМ предусматриваются средства автоматического накопления информации об ошибках при работе машины.

Эта информация позволяет накапливать статистику ошибок с целью выявления наиболее вероятных источников ошибок, а также локализовать причину случайных сбоев и перемежающихся отказов. Последняя задача вследствие невоспроизводимости ситуации представляет собой одну из наиболее сложных проблем эксплуатации ЭВМ.

Автоматическое накопление информации об ошибках основано на регистрации состояния ЭВМ в момент ошибки. В операционных системах предусматриваются специальные средства обработки различных типов ошибок и накопления информации о них в специальных системных журналах ошибок (обычно область на резидентном магнитном диске). Обычная процедура обработки ошибок показана на рис. 13.1.

Средства накопления информации о сбоях и отказах развиваются в направлении охвата всех возможных ошибок в системе, сохранения этой информации для последующего анализа, а также сокращения затрат времени операционной системы на указанные действия путем аппаратурной реализации некоторых ее функций.

13.2. ПРОГРАММНЫЕ СРЕДСТВА ОБРАБОТКИ ОШИБОК

Существуют следующие программные средства обработки ошибок:

- обработчик машинных ошибок;
- обработчик канальных ошибок;

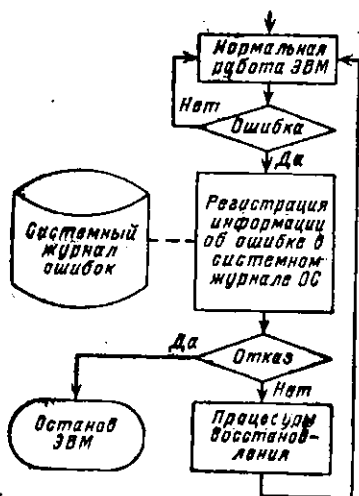


Рис. 13.1. Процедура обработки ошибок

регистратор сбоев периферийных устройств;
 регистратор отказов периферийных устройств;
 регистратор перезагрузок операционной системы;
 регистратор потерь прерываний ввода-вывода;
 регистратор реконфигурации;
 регистратор программных ошибок.

Обработчик машинных ошибок предназначен для обработки ошибок, обнаруженных схемами контроля процессора. Он обеспечивает регистрацию состояния ЭВМ в момент ошибки процессора в системном журнале и восстановление системы [32].

Обработчик канальных ошибок предназначен для обработки канальных ошибок. Он регистрирует данные об ошибке в системном журнале и подготавливает повторение отказавшей команды ввода-вывода с помощью программы восстановления, имеющих для каждого периферийного устройства.

Для регистрации состояния ЭВМ в момент ошибки оно замораживается, после чего записывается в оперативную или специальную память, а затем — в системный журнал ошибок. Первый этап регистрации, т. е. запись состояния аппаратных средств ЭВМ в оперативную или специальную память, называется *аппаратурной регистрацией*. Ее объем определяет точность анализа, локализации ошибок.

В зависимости от структуры ЭВМ аппаратурная регистрация может быть объединенной или раздельной. Обычно способ регистрации зависит от совмещения аппаратуры процессора и каналов.

При совмещении аппаратуры регистрация объединенная, а при независимой аппаратуре — раздельная.

В зависимости от памяти регистрации различают:

- регистрацию в оперативную память;
- регистрацию в специальную буферную память.

Регистрация в оперативную память выполняется при разрешении прерывания от схем контроля. Если в момент

ошибки прерывание запрещено, то состояние ЭВМ в момент ошибки теряется. Регистрация в специальную буферную память удобна тем, что позволяет сохранить информацию об ошибке даже при запрете прерывания от схем контроля. Перепись зарегистрированного состояния ЭВМ в системный журнал ошибок осуществляется обработчиком машинных ошибок при наличии разрешения на прерывание от схем контроля.

Для регистрации информации об ошибках используются следующие носители данных:

- стандартные магнитные диски;
- кассетные магнитные ленты или гибкие магнитные-диски пультаовых накопителей;
- магнитные ленты «накопления»;
- магнитные ленты «истории».

При работе операционной системы состояние ЭВМ в момент ошибки обычно записывается в системном журнале ошибок на стандартных магнитных дисках. Однако вследствие того, что регистрация в системный журнал ошибок требует работоспособности большей части ЭВМ, существует опасность, что система не сможет записать состояние ЭВМ в системный журнал ошибок. С целью исключения возможных потерь информации об ошибках наряду с записью в системный журнал практикуется также регистрация состояния ЭВМ на магнитные носители пультаовых накопителей с помощью аппаратурных средств. При этом используется только аппаратура диагностического ядра ЭВМ.

Накопление и обработка записей о состоянии ЭВМ в момент ошибки очень важны для разработки организационно-технических мероприятий по повышению эксплуатационной надежности. Поэтому записи об ошибках накапливаются и во время выполнения профилактических испытаний ЭВМ. Информация об этих ошибках обычно записывается на лентах накопления, предназначенных для хранения информации об ошибках за небольшой промежуток времени.

Информация об ошибках за длительный период эксплуатации накапливается на лентах истории. Для постоянного пополнения информации об ошибках на лентах истории системе придаются программные средства для переписи на ленту истории информации об ошибках, накопленной в системном журнале, магнитных носителях пультаовых накопителей и лентах накопления.

Регистратор сбоев периферийных устройств предназначен для определения статистики сбоев для каждого пери-

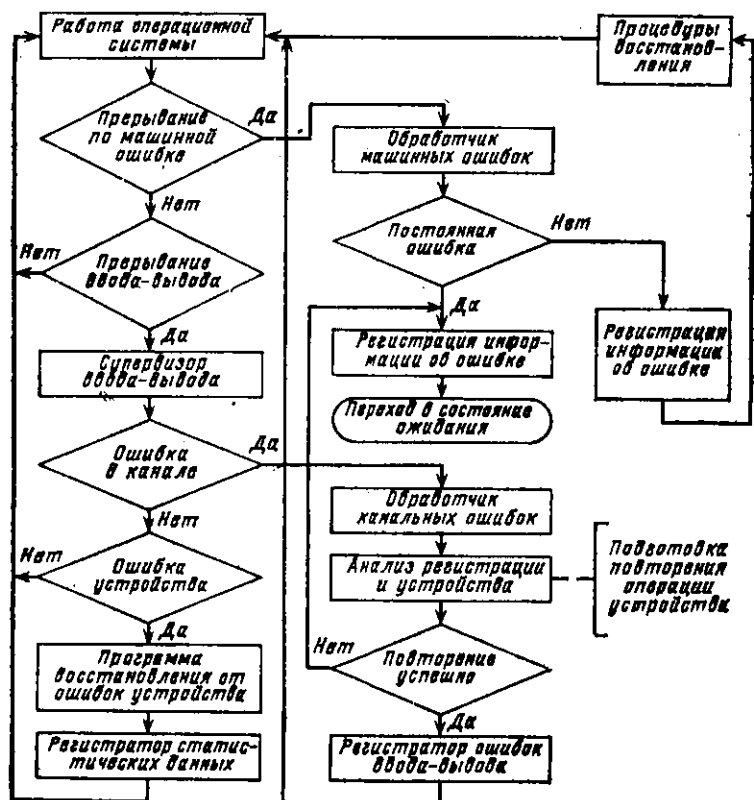


Рис. 13.2. Блок-схема функционирования обработчиков машинных, канальных ошибок и регистраторов ошибок ввода-вывода

ферийного устройства ЭВМ. Регистратор сбоев получает управление от программ восстановления от сбоев периферийных устройств, имеющих в составе операционной системы. Каждый сбой периферийного устройства приводит к увеличению на единицу соответствующего счетчика сбоев.

При переполнении счетчика сбоев регистратор сбоев формирует запись об ошибке периферийного устройства в системный журнал ошибок.

Регистратор отказов периферийных устройств предназначен для записи в системный журнал ошибок информации о неисправимой ошибке (отказе) периферийного устройства.

Диаграмма, иллюстрирующая работу обработчиков машинных и канальных ошибок, а также регистраторов сбоев и отказов периферийных устройств, приведена на рис. 13.2.

Регистратор перезагрузок операционной системы предназначен для записи в системный журнал ошибок информации о причине перезагрузки. Число перезагрузок операционной системы является одним из показателей надежности системы. В некоторых случаях перезагрузка операционной системы считается даже отказом системы. Поэтому во время начальной загрузки операционной системы в ответ на запрос системы предусматривается ввод причины загрузки. В качестве причин перезагрузки операционной системы могут быть указаны такие, как неисправность питания, ошибка системных программ, ошибка аппаратуры, ошибка носителя данных, ненормальное завершение начальной загрузки, ошибка оператора, ошибка в программе пользователя и др.

Статистика причин перезагрузок операционной системы позволяет установить наиболее вероятные среди них и принять меры к их устранению.

Регистратор потерь прерываний ввода-вывода предназначен для записи в системный журнал информации об отсутствии прерываний ввода-вывода, свидетельствующих об окончании начатых операций ввода-вывода. Эта информация включает тип периферийного устройства, его адрес и время.

Регистратор реконфигураций предназначен для записи в системный журнал информации о случаях динамической реконфигурации устройств. Средства динамической реконфигурации операционных систем позволяют при ошибках НМЛ, НМД переустановить пакет или ленту на другое устройство и продолжить нормальную работу. Учет этих случаев также позволяет получить данные о надежности накопителей.

Регистратор программных ошибок предназначен для записи в системный журнал информации о программных ошибках, некоторых ошибках оператора, неудачных попытках программного восстановления при ошибках в аппаратуре.

13.3. ПРОГРАММНЫЕ СРЕДСТВА НАКОПЛЕНИЯ, РЕДАКТИРОВАНИЯ И ПЕЧАТИ ИНФОРМАЦИИ ОБ ОШИБКАХ

Программные средства накопления, редактирования и печати информации об ошибках включают в себя независимую (НПНРП) и системную (СПНРП) программы на-

копления, редактирования и печати информации об ошибках. Независимая программа работает в автономном режиме, а системная — под управлением операционной системы.

Независимая программа выполняет перепись регистраций с магнитных носителей пультовых накопителей на ленту накопления, редактирование и печать регистраций.

Системная программа выполняет редактирование и печать, обобщение ошибок по типам и устройствам, перепись записей об ошибках на ленту истории и позволяет получить:

обобщенные данные о сбоях и отказах за заданный интервал времени;

обобщенные данные об отказах и сбоях периферийных устройств;

тенденцию изменения надежности ЭВМ во времени в течение заданного интервала с указанием числа сбоев и отказов за каждый день заданного интервала времени;

обобщение статистики по всем накопителям на магнитных дисках и магнитных лентах;

распечатку истории сбоев и отказов за заданный промежуток времени с краткой характеристикой задания, адреса периферийного устройства, команды, на которой произошла ошибка, серийного номера тома, а также действий системы;

распечатку по типам записей.

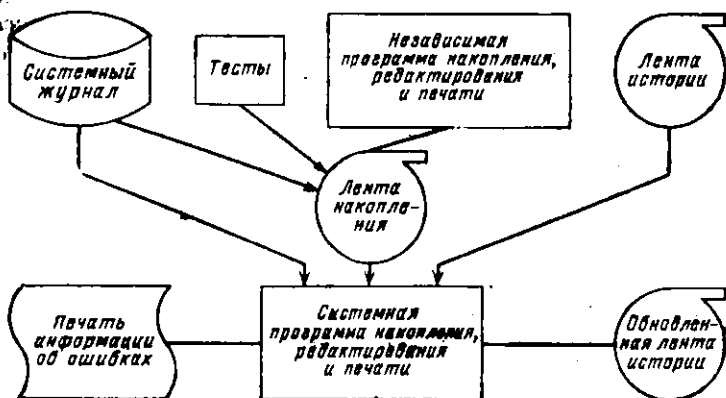


Рис. 13.3. Формирование обновленной ленты истории

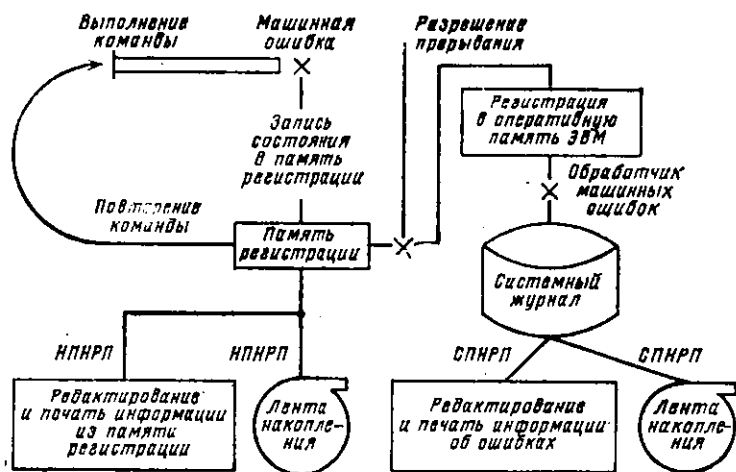


Рис. 13.4. Функции программ НПНРП и СПНРП

Процесс формирования обновленной ленты истории показан на рис. 13.3. На рис. 13.4 показаны некоторые функции программ НПНРП и СПНРП.

13.4. АППАРАТУРНО-МИКРОПРОГРАММНЫЕ СРЕДСТВА НАКОПЛЕНИЯ ИНФОРМАЦИИ ОБ ОШИБКАХ

С целью сокращения непроизводительных затрат времени и упрощения интерфейса между аппаратурой и программными средствами в современных ЭВМ некоторые функции операционной системы реализуются аппаратурно-микропрограммным способом.

К этим функциям относятся регистрация состояния ЭВМ в момент сбоя и восстановление ЭВМ при сбоях путем повторения прерванной операции.

Другой функцией операционной системы, которая может быть передана аппаратурно-микропрограммным средствам, является накопление информации об ошибках периферийного устройства. Один из примеров такой системы, реализованной в устройстве управления накопителями на магнитных дисках, показан на рис. 13.5.

Микропрограммы устройства управления обрабатывают ошибки накопителей на магнитных дисках, выполняют регистрацию, накапливают статистические данные, собирают информацию об ошибках, выполняют процедуры восстановления.

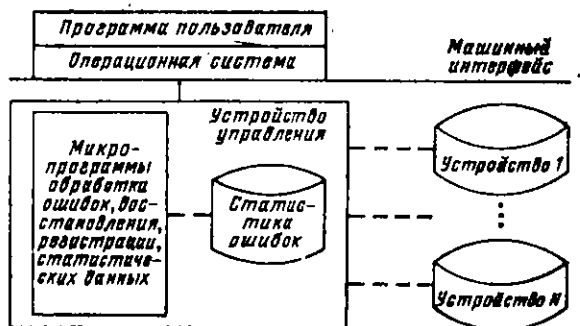


Рис. 13.5. Устройство управления НМД с внутренней регистрацией ошибок

Перепись накопленной в устройстве управления информации об ошибках в оперативную память ЭВМ выполняется при переполнении памяти регистрации устройства управления или при обработке состояния типа «Сбой устройства».

13.5. СБОР, ОБРАБОТКА, ОБОБЩЕНИЕ И НАКОПЛЕНИЕ ИНФОРМАЦИИ О НАДЕЖНОСТИ ЭВМ

С целью усовершенствования ЭВМ, повышения надежности и эффективности функционирования пользователи ЭВМ представляют разработчику и заводу-изготовителю информацию об отказах и их причинах. Эта информация позволяет установить эффективную обратную связь между разработчиками, изготовителями и пользователями ЭВМ.

Целью сбора, обработки, обобщения и накопления информации о причинах ошибок является:

- усовершенствование конструкции и технологии изготовления ЭВМ;
- проведение мероприятий, направленных на повышение качества и эффективности обслуживания;
- создание банка симптомов ошибок.

В процессе этих работ решаются следующие задачи:

- определение наименее надежных изделий;
- определение причин и механизмов возникновения отказов;
- выявление влияния условий и режимов эксплуатации на надежность;

определение влияния частоты и продолжительности профилактических работ на надежность.

Первичные формы учета предназначены для записи первичной не-систематизированной информации о надежности. Первичными формами

учета информации о надежности являются аппаратный журнал и сводная карточка учета неисправностей. Аппаратный журнал предназначен для ведения всех записей, необходимых обслуживающему персоналу при проведении работ, и записей, характеризующих наработку на отказ и неисправность ЭВМ, внешнее проявление неисправности, время ее возникновения, обнаружения и устранения, неисправное устройство.

При сборе информации о надежности особое внимание уделяется информации по анализу причин отказа изделий.

Информация о надежности, вносимая в первичные формы учета, должна быть изложена четко и кратко, иметь достаточную полноту и достоверность с целью обеспечения возможности последующего ее кодирования с помощью таблиц шифров для обработки на ЭВМ.

При индивидуальном обслуживании ЭВМ пользователи заполняют сводную карточку учета неисправностей и высылают ее на предприятие-изготовитель с определенной периодичностью.

При централизованном обслуживании сбор и обработка информации о надежности сосредотачиваются в руках одной организации, что повышает эффективность этой системы.

Для автоматизированной обработки информации о надежности данные с первичных форм учета переносятся на специальные карты учета неисправностей, разработанные с учетом автоматизированной обработки.

Формы учета должны обеспечивать возможность решения задач по сбору, обработке, обобщению и накоплению информации о надежности. Они должны быть достаточно просты и не должны быть перегружены информацией, с тем чтобы их заполнение не требовало много времени.

Для автоматизированной обработки информации о надежности имеются пакеты прикладных программ, которые на основании имеющейся информации позволяют получить выходные документы, включающие в себя качественные и количественные характеристики надежности, сведения о:

- неисправностях изделия в целом;
- наработке на отказ;
- техническом уровне разрабатываемого изделия;
- неисправностях, принятых мерах и времени, затраченном на устранение неисправностей;
- параметрах потока отказов;
- вероятности безотказной работы;
- среднем времени восстановления;
- коэффициенте готовности.

Наличие систем автоматизированной обработки информации о надежности ЭВМ позволяет создать банк симптомов ошибок и методов их устранения. Наличие большого и постоянно обновляемого банка

симптомов ошибок является надежным источником статистики ошибок и рекомендаций по устранению неисправностей. Кроме предоставления аналитических данных для усовершенствования выпускаемых ЭВМ этот банк содержит информацию для программы предсказания надежности. На основе банка симптомов ошибок может быть составлен и рекомендуемый список запасных частей ЭВМ.

Глава четырнадцатая

МЕТОДЫ ПОВЫШЕНИЯ ЭКСПЛУАТАЦИОННОЙ НАДЕЖНОСТИ СИСТЕМ ЭЛЕКТРОПИТАНИЯ ЭВМ

14.1. ОСНОВНЫЕ ПРОБЛЕМЫ ЭКСПЛУАТАЦИИ СИСТЕМ ЭЛЕКТРОПИТАНИЯ ЭВМ

Практика эксплуатации ЭВМ показывает, что основными проблемами, возникающими при эксплуатации систем электропитания ЭВМ, являются:

- оперативный контроль напряжения и локализация отказов источников питания ЭВМ;
- защита ЭВМ от возмущений в системе электропитания;
- защита ЭВМ от длительных перерывов электропитания.

Контроль напряжения и локализация отказов вторичных источников питания

В больших современных ЭВМ число вторичных источников питания достигает нескольких десятков и контроль их выходных напряжений, а также локализация отказов представляют собой достаточно трудоемкие процедуры.

Для облегчения эксплуатации системы электропитания в современных ЭВМ предусматриваются средства оперативного контроля напряжения и локализация отказов вторичных источников питания ВИП с индикацией напряжения на табло цифрового вольтметра или экране дисплея сервисного процессора.

Примером ЭВМ с оперативным контролем и измерением выходных напряжений является ЭВМ ЕС-1045.

Автоматизированная система контроля и диагностики электропитания (КДЭ) машины ЭВМ ЕС-1045 выполняет следующие функции:

- измерение и контроль напряжений ВИП;
- установку профилактических уровней напряжений ВИП по командам на ЭВМ или вручную с пульта КДЭ;

локализацию места и типа аварийных неисправностей, таких как температурные перегревы, отказ вентиляторов, авария блоков управления питанием с выдачей аварийного звукового и светового сигналов.

Структурная схема КДЭ приведена на рис. 14.1.

Режим работы КДЭ задается блоком выбора режима БВР. Код адреса ВИП из БВР поступает на дешифратор Дш, сигнал с выхода которого в зависимости от режима работы управляет коммутаторами измерения Ки, профилактического контроля Кп или аварии Ка.

При измерении выходных напряжений источников питания Ки подключает выход выбранного ВИП к аналого-цифровому преобразователю АЦП, от которого информация в двоично-десятичном коде поступает на схему сравнения СС, где сравнивается с допустимыми отклонениями. При выходе напряжения ВИП за заданные допуски на пульте КДЭ появляется сигнал отклонения от нормы. В качестве АЦП используется цифровой вольтметр. Кроме того, напряжение адресованного ВИП индицируется на табло цифрового вольтметра. При возникновении температурного перегрева, зафиксированного датчиками температуры (ДТ), останова вентилятора (ВЕНТ) или отключения блока управления питанием (БУП), вырабатывается сигнал «Авария», включающий звуковую и световую сигнализацию. На пульте КДЭ высвечивается тип аварии и двоичный адрес неисправного блока.

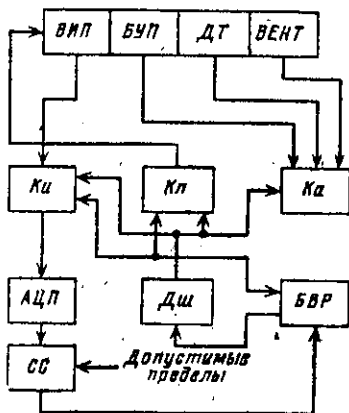


Рис. 14.1. Структурная схема КДЭ

14.2. ЗАЩИТА ЭВМ ОТ ВОЗМУЩЕНИЙ В СИСТЕМЕ ЭЛЕКТРОПИТАНИЯ

Характеристики возмущений

Стандартным требованием к электропитанию ЭВМ является питание от сети переменного тока $220 \text{ В}^{+10\%}_{-15\%}$ с частотой $50 \pm 1 \text{ Гц}$. Нормы на качество электроснабжения не регламентируют такие возмущения в

сети электропитания, как провалы и всплески напряжения [15]. В то же время подобные возмущения напряжений сети являются источником сбоев, приводящих к значительным потерям полезного времени работы ЭВМ. Знание характеристик возмущений напряжения в сети имеет большое значение, так как позволяет разработать специальные меры защиты ЭВМ.

Слежение за напряжением первичной сети выполняется специальными устройствами — *регистраторами помех* [15], которые обнаруживают, подсчитывают и регистрируют в первичной сети такие события, как провалы или всплески напряжения, импульсные помехи и отклонения частоты от нормы. Поскольку эти регистраторы имеют различные характеристики, очень важен правильный их выбор.

Имеются данные о том, что в результате электрических аварий, неkoordinированных срабатываний различных защитных устройств могут происходить провалы напряжения длительностью до 20 периодов, в то время как для нарушения нормального функционирования большинства ЭВМ достаточно потери только одного периода напряжения сети.

Всплески и провалы напряжения происходят также при коммутации электрических цепей с различной нагрузкой, пуске мощных электродвигателей и другого электрооборудования. Кроме того, на частотах 1 МГц и выше наблюдаются кратковременные, но значительные по амплитуде помехи, возникающие при работе различных релейных устройств. Исследования показали [15], что значительная часть импульсных помех имела длительность менее 0,2 мкс.

Провалы и перенапряжения имеют, как правило, длительность от 2 до 100 мкс. Амплитуда провалов 70–90 В. Возможны также отключения питания. Поскольку в сети питания могут быть помехи, то помехоустойчивость должна быть свойством ЭВМ.

В [15] предложена следующая интерпретация импульсной помехозащищенности.

Импульсная помехозащищенность ЭВМ $\beta_{и.п}$ определяется как отношение числа зарегистрированных сбоев m к числу n поданных за время испытаний импульсов помех с нормированной амплитудой $U_{н.п}$ и частотой следования $f_{н.п}$:

$$\beta_{и.п} = m/n.$$

Предполагается, что экспериментальное определение импульсной помехозащищенности выполняется с помощью специальных имитаторов импульсных помех.

Аналогичным способом определяется помехозащищенность ЭВМ от длительных помех (провалов и перенапряжений).

Имитаторы импульсных и длительных помех должны имитировать помехи по амплитуде и длительности, соответствующие реальным характеристикам сети питания.

Методы защиты ЭВМ от помех

Методы защиты ЭВМ от помех можно разделить на электрические и логические. К электрическим методам защиты от импульсных помех относятся сетевые фильтры, устанавливаемые на входе системы электропитания, и электростатические экраны между первичными и вторичными обмотками трансформаторов, существенно уменьшающие паразитные межобмоточные емкости.

К электрическим методам защиты от длительных помех относится применение феррорезонансных стабилизаторов, преобразователей напряжения и мотор-генераторов.

Феррорезонансные стабилизаторы обеспечивают изменение выходного напряжения в пределах 1—5 % при амплитудах длительных помех, не превышающих 20 % номинального напряжения.

Преобразователи частоты и мотор-генераторы обеспечивают частичное резервирование первичного питания ЭВМ при длительных помехах за счет инерции вращающихся частей.

К логическим методам защиты ЭВМ от помех относятся программно-аппаратурные средства восстановления ЭВМ от сбоев. Эти средства обеспечивают замораживание состояния ЭВМ при сбое, обнаруженном схемами контроля, выдержку определенного времени (как правило, несколько миллисекунд) и повторение микрокоманды, команды или части программы. Выдержка времени до начала процесса повторения обычно определяется вероятной длительностью помех и в принципе может варьироваться в зависимости от характеристик помех в сети.

14.3. ЗАЩИТА ЭВМ ОТ ДЛИТЕЛЬНЫХ ПЕРЕРЫВОВ ЭЛЕКТРОПИТАНИЯ

Гарантированное электропитание

ЭВМ, используемые для решения ответственных задач, таких как управление движением, резервирование билетов, коммутация сообщений, задачи в реальном масштабе времени и т. д., как правило, обеспечиваются гарантированным электропитанием. Введем понятие *длительности гарантированного питания*. Будем понимать под этим параметром длительность обеспечения ЭВМ электропитанием при отказе основного источника первичного электропитания, причем обеспечения без нарушения процесса вычислений.

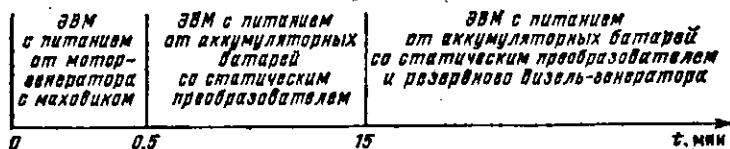


Рис. 14.2. Средства обеспечения гарантированного электропитания

Разным значениям длительности гарантированного электропитания соответствуют различные средства гарантированного электропитания (рис. 14.2).

Поскольку ЭВМ чувствительна к пропаданию первичного электропитания даже на полпериода, а также к изменению фазы при подключении другого источника электропитания, система гарантированного электропитания должна это учитывать.

Системы гарантированного электропитания в порядке возрастания длительности гарантированного электропитания располагаются следующим образом:

мотор-генератор;

аккумуляторные батареи со статическим преобразователем;

аккумуляторные батареи со статическим преобразователем и резервным питанием от дизель-агрегата.

Первые два типа систем гарантированного электропитания обеспечивают кратковременное электропитание при отказе первичной сети и не обеспечивают восстановление вычислительного процесса. Для продолжения вычислений после восстановления электропитания в программном обеспечении ЭВМ предусматриваются дополнительные средства. Третий тип систем гарантированного электропитания обеспечивает длительное гарантированное электропитание.

Гарантированное электропитание от мотор-генератора

Мотор-генератор обеспечивает гарантированное электропитание в течение нескольких секунд. Этого времени, как правило, недостаточно для планомерного отключения ЭВМ с сохранением всех необходимых данных для продолжения решения задач после восстановления электропитания.

При нарушениях в первичной сети вырабатывается предупреждающий сигнал, который вызывает прерывание по нарушению электропитания. Прерывание может обрабатываться операционной системой для планомерного сворачивания операций до отключения электропитания.

Гарантированное электропитание от аккумуляторных батарей со статическими преобразователями

Типичная система гарантированного электропитания состоит из трех основных компонентов [85, 91]:

статического преобразователя постоянного тока в переменный;

дополнительного источника постоянного тока, такого как аккумуляторная батарея;

зарядного устройства для аккумуляторов.

Статический преобразователь вырабатывает прямоугольные импульсы переменного тока с помощью мостовой схемы с четырьмя тиристорами. Переменный ток с выхода статического преобразователя поступает на вход фильтра, который формирует синусоидальное напряже-

ние. В качестве фильтра обычно используется феррорезонансный трансформатор.

Обычно аккумуляторные батареи обеспечивают длительность гарантированного электропитания до 15—30 мин.

В нормальном режиме работы статический преобразователь питается от сети переменного тока через выпрямитель. При пропадании напряжения сети статический преобразователь продолжает питаться от аккумуляторной батареи.

Наиболее простой схемой, используемой в системах, где допустимо кратковременное отключение, является схема с питанием от первичной сети и ручным переключателем на питание от аккумуляторной батареи. Эта схема приведена на рис. 14.3.

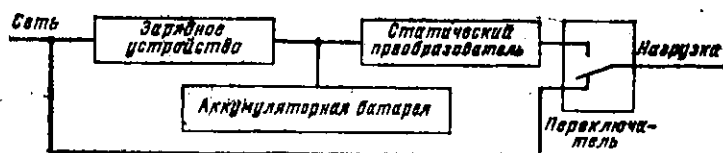


Рис. 14.3. Гарантированное электропитание с ручным переключением на аккумуляторную батарею



Рис. 14.4. Гарантированное электропитание с постоянным питанием нагрузки от аккумуляторной батареи

На рис. 14.4 приведена схема системы гарантированного электропитания с постоянным питанием нагрузки от аккумуляторной батареи. В этом случае зарядное устройство при наличии напряжения сети выполняет две функции: выпрямителя и зарядного устройства аккумуляторной батареи. При отказе системы переключение на первичную сеть может выполняться вручную или автоматически.

Более высокой надежностью обладает схема гарантированного электропитания с выпрямителем переменного тока (рис. 14.5). При такой схеме переход от аккумуляторного питания к питанию от сети осуществляется проще. Напряжение первичной сети поступает на выпрямитель и зарядное устройство. Таким образом, вход статического преобразователя подключен к выпрямителю и аккумуляторной батарее, выходы которых разделены диодом или тиристором. В такой системе не возникает проблем при прекращении питания от сети или от аккумуля-

муляторной батареи, хотя при отказе статического преобразователя возникает проблема перехода к питанию от сети с помощью переключателя, ручного или автоматического.

Сглаживание переходных процессов на шинах питания нагрузки во время переключения достигается путем синхронизации выходного напряжения преобразователя и сети.

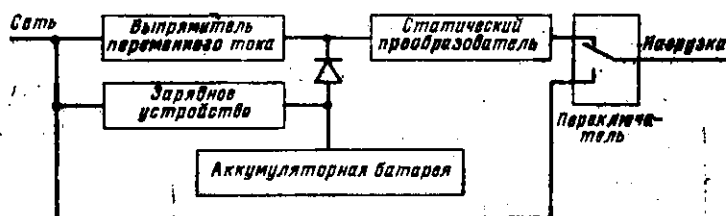


Рис. 14.5. Гарантированное электропитание с постоянным питанием нагрузки от сети и от аккумуляторной батареи

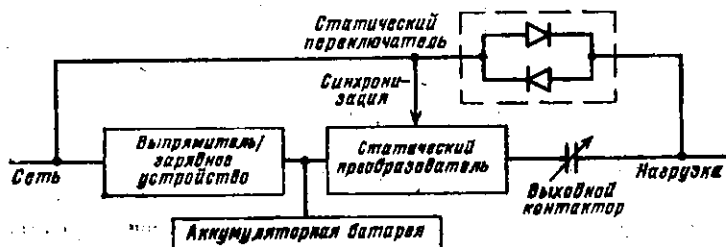


Рис. 14.6. Гарантированное электропитание со статическим переключателем питания

На рис. 14.6 показана система гарантированного электропитания со статическим переключателем, который обеспечивает автоматическое переключение нагрузки на питание от первичной сети при неисправности в самой системе гарантированного электропитания [91].

Переключение нагрузки происходит практически мгновенно и не вызывает переходных процессов.

Защита вычислительного процесса при небольшой длительности гарантированного электропитания

Гарантированное электропитание обеспечивает нормальную работу ЭВМ только в течение нескольких минут, поэтому отключение ЭВМ часто бывает неизбежным. Чтобы отключение электропитания не вызывало существенных потерь в ЭВМ, требующих последующих сложных процедур восстановления, в современных ЭВМ предусматривают

аппаратурно-программные средства подготовки упорядоченного отключения устройств ЭВМ и сохранения ее состояния для последующего продолжения работы при восстановлении питания.

Рассмотрим систему гарантированного питания и программные средства ЭВМ, обеспечивающие упорядоченное отключение ЭВМ при отказе первичного электропитания и последующее восстановление прерванного задания.

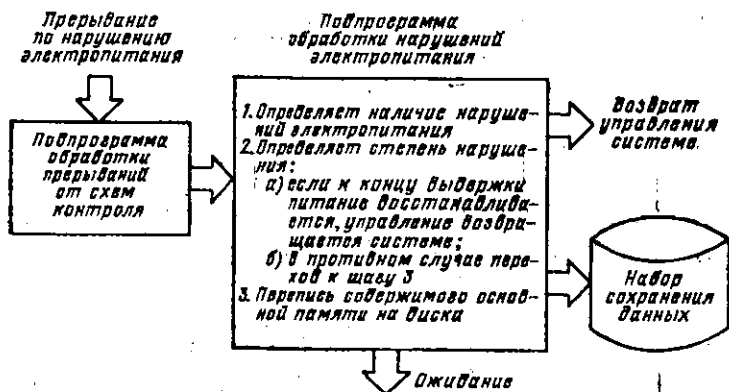


Рис. 14.7. Работа системы при нарушениях питания

Система включает в себя следующие средства:

- 1) аккумуляторные батареи;
- 2) аппаратуру сигнализации о нарушениях первичного электропитания;
- 3) программные средства, обеспечивающие упорядоченное отключение устройств и сохранение содержимого памяти на дисках;
- 4) средства обработки прерываний по нарушению электропитания;
- 5) дополнительные накопители на магнитных дисках;
- 6) программные средства ОС, обеспечивающие восстановление состояния ЭВМ при инициализации системы после восстановления первичного электропитания и продолжение прерванной работы.

Аккумуляторные батареи должны обеспечивать электропитанием критическую часть ЭВМ, включающую центральный процессор, все каналы ввода-вывода, накопители на магнитных дисках и их управление.

При включении в ОС средств обеспечения гарантированного электропитания во время генерации ОС пользователь может задать специальный параметр, определяющий время между получением предупредительного сигнала и началом операций по сбросу содержимого оперативной памяти на диски.

На рис. 14.7 показана работа системы при нарушениях электропитания.

Система работает следующим образом.

1. Нарушение электропитания вызывает передачу управления подпрограмме обработки нарушений электропитания.

2. Время выдержки, заданное при генерации системы, показывает сколько времени система может работать на электропитании аккумуляторных батарей в ожидании восстановления первичного электропитания. Если первичное электропитание восстанавливается до окончания выдержки, управление возвращается системе. Если выдержка времени

заканчивается до восстановления первичного электропитания, управление передается подпрограмме сохранения.

3. Подпрограмма сохранения прекращает все операции ввода-вывода и сбрасывает содержимое основной памяти и памяти ключей защиты на диски и переводит ЭВМ в состояние ожидания.

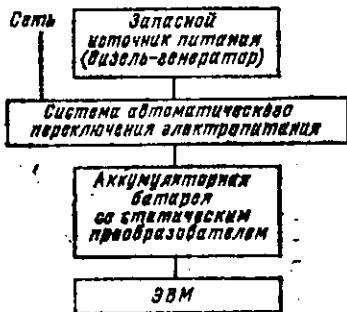


Рис. 14.8. Схема установки с большой длительностью гарантированного электропитания

Восстановление системы и продолжение решения прерванных заданий после восстановления первичного электропитания производится путем повторной загрузки операционной системы. Во время первоначальной загрузки ОС пользователь может восстановить состояние памяти либо стереть наборы данных на дисках сохранения.

Гарантированное электропитание большой длительности

Для особо ответственных применений ЭВМ неприемлемо отключение электропитания, даже если обеспечено продолжение прерванных заданий после восстановления электропитания. К таким системам относятся системы реального масштаба времени с телекоммуникацией, банковские и другие системы. На рис. 14.8 показана схема установки с большой длительностью гарантированного электропитания. Обычно в качестве установки гарантированного питания используются статические преобразователи и дизель-генератор.

Статические преобразователи синхронизированы с выхода по фазе с первичным электропитанием, так как при выходе из строя статического преобразователя ЭВМ переключается на первичную сеть, сигнализируя об аварии статического преобразователя.

При отказе первичного электропитания статические преобразователи обеспечивают ЭВМ питанием в течение времени, достаточного для запуска дизель-генераторов. Обычно батареи обеспечивают питанием в течение 15 мин. Время запуска генераторов составляет несколько минут.

Подобные системы гарантированного электропитания обычно выполняются наращиваемыми, чтобы их можно было расширять при подключении новых ЭВМ, а также модульными для обеспечения их резервирования.

14.4. ПОМЕХИ В ЦЕПЯХ СИГНАЛОВ И СПОСОБЫ ВЫПОЛНЕНИЯ МЕЖСОЕДИНЕНИЙ ИНТЕГРАЛЬНЫХ МИКРОСХЕМ

В данном параграфе рассмотрен вопрос, прямо не относящийся к организации системы электропитания ЭВМ, но тесно с ней связанный. Речь пойдет о довольно тонкой проблеме — источниках помех в цепях сигналов, часто являющихся причиной сбоев в работе машины, и некоторых методах снижения уровня помех. Эти вопросы, безусловно учитываемые при проектировании машин, являются актуальными и для персонала, занятого техническим обслуживанием ЭВМ.

Высокое быстродействие интегральных микросхем и большая плотность их компоновки и монтажа, с одной стороны, и наличие большого числа разнесенных на сравнительно значительные расстояния устройств, с другой, приводят к тому, что в системах, построенных на интегральных микросхемах, имеется значительный уровень помех, сравнимый с запасом помехоустойчивости самих схем.

При скоростях работы, характерных для интегральных микросхем, имеющих времена переключения и задержки порядка единиц наносекунд, электрические характеристики системы межсоединений и системы электропитания оказывают существенное влияние на быстродействие вычислительной системы и искажения сигналов при прохождении их по линиям связи.

При построении вычислительных устройств и систем, использующих в качестве элементной базы быстродействующие интегральные микросхемы, и в частности интегральные микросхемы ТТЛ, ТЛЭС и др., важными вопросами являются способ выполнения монтажа схем и согласования схем с линиями передачи сигналов.

Влияние соединительных линий на передачу сигналов в цифровом оборудовании определяется соотношением длины электромагнитных волн λ , соответствующих спектру частот передаваемых сигналов, и физической длины линии связи l .

Межсоединения можно классифицировать следующим образом:

1) имеющие длину, меньшую $0,0001$ длины волны, соответствующей максимальной частоте активного спектра сигнала; эквивалентны электрической схеме из последовательно и параллельно включенных резисторов;

2) для которых l/λ лежит в диапазоне $0,0001—0,05$; могут рассматриваться как схемы с сосредоточенными сопротивлениями, емкостями и индуктивностями;

3) для которых $l/\lambda > 0,05$; должны рассматриваться как длинные линии передачи с распределенными параметрами.

Соединительные линии длиной более 0,5—1 м в системах, построенных на интегральных микросхемах ТТЛ, должны рассматриваться как длинные линии передач с распределенными параметрами, так как задержка в типичных линиях составляет $\tau = 5 + 6$ нс/м и соизмерима с длительностью переключения схем ТТЛ (5—15 нс). При таком соотношении быстродействия соединяемых микросхем и длины межсоединений в соединительных линиях передачи возникают помехи:

1) вследствие явлений отражения из-за несогласования волнового сопротивления линии и входных и выходных сопротивлений интегральных микросхем, что может приводить к неправильному функционированию и снижению быстродействия системы;

2) из-за перекрестных наводок между соседними проводниками, которые также могут приводить к неправильной работе.

При проектировании устройств и систем на интегральных микросхемах необходимы оценка уровня помех из-за отражений и перекрестных наводок и определение условий, при которых обеспечивается достаточный для надежной работы запас от помех, возникающих в линиях передачи.

Снижение помех из-за отражений и перекрестных наводок является одной из основных задач, решаемых при выборе системы монтажа. При этом вопросы искажения сигналов из-за отражений и взаимных наводок в соединительных линиях приобретают наибольшее значение при увеличении длины линии и особенно важны при построении систем, объединяющих большое число устройств, при создании систем связи, интерфейсов.

Для быстродействующих вычислительных систем необходимы: 1) система монтажа, обеспечивающая непосредственное (без каких-либо дополнительных мер) соединение интегральных микросхем с ограниченной длиной линий передачи, и 2) система монтажа для передачи сигналов на расстояния, превышающие допустимую длину линий, обеспечиваемую первой системой монтажа, с контролируемым уровнем помех в линиях передачи.

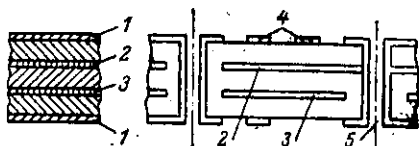
Межсоединения интегральных микросхем могут выполняться одиночными проводниками или с помощью специальных соединительных линий с контролируемыми параметрами. Однако при использовании одиночных проводников вследствие их высокой индуктивности допускаемая длина межсоединения при том же уровне помех из-за монтажа оказывается в 2—3 раза меньше, чем при использовании линий, имеющих обратные провода, расположенные на близком расстоянии от сигнальных проводов. Такие линии с контролируемыми параметрами могут быть выполнены в виде микрополосковых линий, получаемых с помощью многослойного печатного монтажа с внутренними слоями земли и питания; коаксиальных кабелей, витых пар проводов и т. п. Из-за небольшой индуктивности таких соединительных линий их длина по условиям помехоустойчивости может быть значительно больше, чем

длина одиночных проводников. Параметры этих линий (емкость, индуктивность и определяемое ими характеристическое сопротивление линии) постоянны в любом сечении линии, сохраняются неизменными в процессе эксплуатации, не зависят от расположения в общей системе и могут быть легко рассчитаны. Это облегчает определение влияния линий связи на искажения сигналов, что необходимо при проектировании быстродействующих систем.

Для первой системы монтажа целесообразно использовать микрополосковые линии, изготавливаемые с помощью многослойного печатного монтажа с внутренними слоями земли и питания, который обеспечивает высокую плотность компоновки и монтажа интегральных микросхем. Поперечное сечение четырехслойной печатной панели с микрополосковыми линиями передачи показано на рис. 14.9. Первая система монтажа используется обычно в панелях устройств. Размеры

Рис. 14.9. Четырехслойный печатный монтаж с микрополосковыми линиями передачи:

1 — слой сигнальных проводников;
2 — экранирующий слой земли;
3 — экранирующий слой питания;
4 — сигнальные проводники;
5 — сквозное отверстие с металлическим покрытием



панели, правила проложения линий и длина линий выбираются таким образом, чтобы помехами из-за отражений в линиях можно было пренебречь.

При построении систем, включающих ряд вычислительных устройств, длина соединительных линий между устройствами, как правило, превышает длину линии, отражениями в которой можно пренебречь. Поэтому здесь необходимо использовать вторую систему монтажа и выполнять линии передачи с помощью коаксиальных или других специальных кабелей или витых пар проводов.

Оценка искажений сигналов в таких линиях должна проводиться с учетом распределенных параметров линий передачи.

Пример типичных искажений сигналов в несогласованной длиной линии передачи, соединяющей схемы ТТЛ, показан на рис. 14.10. При передаче спада сигналов в результате отражений в линиях передачи на выходе линий возникают колебания с отрицательными и положительными выбросами значительной амплитуды. Отрицательные выбросы на входе приемных схем, включенных на выходе линии, могут вывести их из строя, а положительные выбросы привести к ложному срабатыванию этих классов схем.

При передаче фронта сигналов на входе линии имеется ступенчатое нарастание сигнала, что может вызвать увеличение задержки распространения сигнала через присоединенные схемы, включенные на входе линии, или многократное срабатывание этих схем, если ступень напряжения находится на ее пороге срабатывания.

В результате таких искажений сигналов снижается запас от помех передаваемых сигналов и, следовательно, надежность системы. Поэтому в этом случае необходимо вводить специальные цепи для согласо-

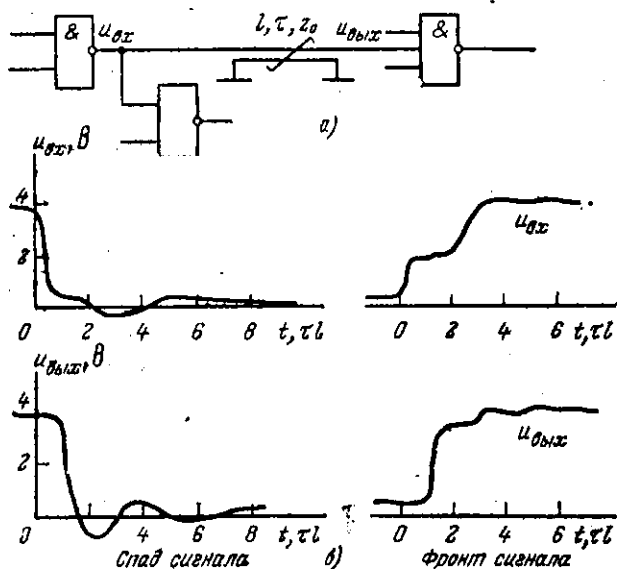


Рис. 14.10. Искажение сигналов при передаче по длинным линиям, соединяющим схемы ТТЛ:

а — соединения схем ТТЛ длинной линией передачи (l — длина линии, τ — задержка распространения сигнала на единицу длины линии; z_0 — характеристическое сопротивление); б — сигналы на входе и выходе линии

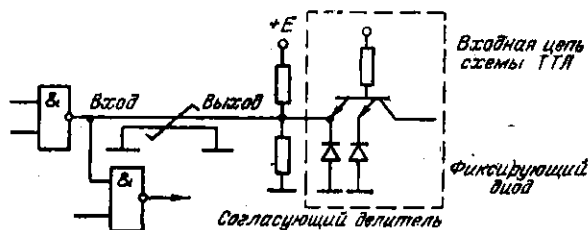


Рис. 14.11. Цепи согласования линии передачи

вания волнового сопротивления линии с входными сопротивлениями соединяемых схем в виде согласующих резисторов на концах линий и фиксирующих диодов на входах приемных схем (рис. 14.11).

Для передачи сигналов при этом необходимо использовать специальные схемы передачи, обеспечивающие достаточную мощность и за-

щиту от коротких замыканий, и специальные приемные схемы с высоким входным сопротивлением, что позволяет производить произвольное подключение приемных схем вдоль линии передачи.

Глава пятнадцатая

КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНАЯ АППАРАТУРА ДЛЯ ЭКСПЛУАТАЦИОННОГО ОБСЛУЖИВАНИЯ ЭВМ

15.1. ТИПЫ КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫХ ПРИБОРОВ

Эффективность поиска неисправностей и ремонта средств вычислительной техники значительно повышается при наличии удобных и простых в эксплуатации контрольно-измерительных приборов. Неудивительно, что номенклатура контрольно-измерительных приборов и их возможности постоянно расширяются. Широкое распространение в практике эксплуатации средств вычислительной техники получили такие приборы, как одноконтактные и многоконтактные логические пробники, компараторы, импульсные генераторы, измерители тока.

Широкое внедрение микропроцессоров и микро-ЭВМ и проблемы, возникающие при их наладке и эксплуатации, потребовали создания качественно новых типов контрольно-измерительных приборов, таких как логические и сигнатурные анализаторы, портативные стенды для ремонта печатных плат с микросхемами.

В данной главе дается описание наиболее распространенных контрольно-измерительных приборов, их возможности и способы использования.

15.2. ОДНОКОНТАКТНЫЙ ЛОГИЧЕСКИЙ ПРОБНИК

Одноконтактный логический пробник представляет собой прибор для индикации двоичного состояния элементов дискретных схем.

Основные преимущества логических пробников — компактность, возможность работы в труднодоступных местах, питание от источника проверяемого логического устройства, удобство работы.

Задача логических пробников — упростить проверку логических схем, давая пользователю возможность наблюдать логические уровни без настройки и калибровки, которые необходимы при измерениях с помощью осциллографов.

Для индикации состояния элементов схемы применяются лампочки накаливания или светодиоды, число которых может быть различным.

В одноламповом пробнике включенное состояние лампочки означает, что в проверяемой точке схемы имеется сигнал, соответствующий логической единице, выключенное — логическому нулю, а мигание с

определенной частотой — переключению уровней сигнала. Постоянное свечение с половинной яркостью может означать отсутствие сигнала [63].

В любом случае наиболее важным качеством пробника являются четкость и однозначность показаний. Во многих пробниках применяются лампочки накаливания ввиду их большой яркости.

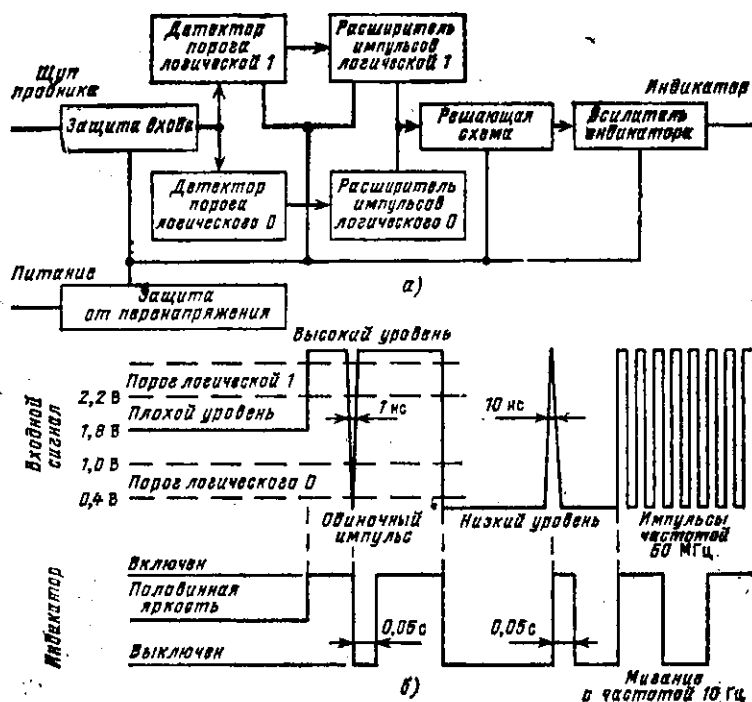


Рис. 15.1. Одноламповый логический пробник:

а — структурная схема; б — показания пробника при различных значениях измеряемых сигналов

Очень важным достоинством логических пробников является возможность работы с различными комплексами ИС, например ЭСЛ, ТТЛ и др. Это очень удобно при эксплуатации вычислительных систем, где, как правило, используются различные комплексы ИС. Обычно эта возможность реализуется наличием на корпусе пробника переключателя, устанавливаемого в положение, соответствующее комплексу ИС, с которым предполагается работа в данный момент.

В логические пробники встраивается также схема расширения импульсов, позволяющая оператору проверять наличие импульсных сиг-

налов высокой частоты. В таких пробниках при наличии на входе последовательности импульсов лампочка начинает мигать с частотой несколько герц.

На рис. 15.1 приведены структурная схема однолампового логического пробника, предназначенного для работы с ТТЛ-схемами, и его показания при различных значениях измеряемых сигналов.

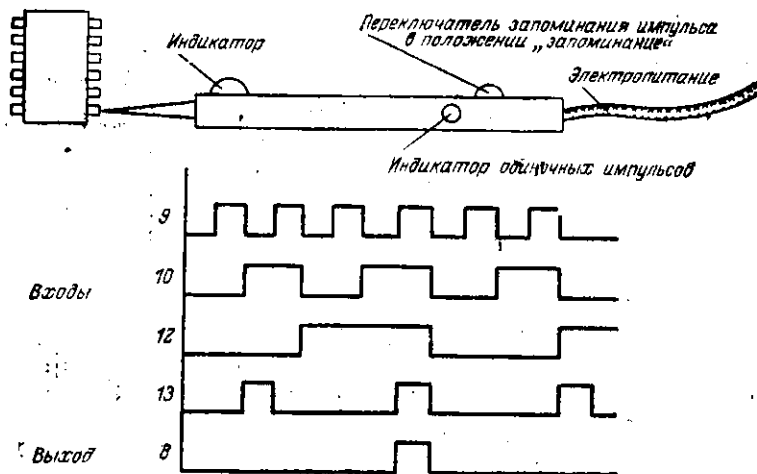


Рис. 15.2. Режим запоминания одиночных импульсов

В некоторых логических пробниках имеется встроенная схема запоминания импульсных сигналов и индикатор, позволяющий обнаружить наличие нерегулярных импульсов. Когда в контролируемой точке схемы после подсоединения пробника происходит изменение логического состояния и запоминающий элемент сброшен, то он запоминает импульс и индицирует его, пока переключатель запоминания не будет возвращен в исходное состояние. На рис. 15.2 показана последовательность импульсов на входе четырехвходовой схемы И, которая вызывает одиночный импульс на выходе схемы, запоминаемый пробником. Этот импульс включает индикатор импульса, который остается во включенном состоянии и после отвода щупа пробника от контакта схемы.

15.3. МНОГОКОНТАКТНЫЙ ЛОГИЧЕСКИЙ ПРОБНИК

Многоконтактный логический пробник предназначен для одновременной индикации состояния всех контактов интегральной микросхемы.

Он имеет отдельный индикатор для каждого контакта и позволяет, таким образом, наблюдать значения сигналов на всех контактах ИС.

Многоконтактный логический пробник надевается как клипса на микросхему. Точные пластмассовые направляющие гарантируют надежное соприкосновение между контактами пробника и выводами ИС. Обычно в этих приборах имеется схема поиска питающего напряжения, которая автоматически определяет соответствующие выводы ИС и подключает их к схемам пробника [63].

Многоконтактные логические пробники имеют защиту от повышенного напряжения, предотвращающую перегрузку схемы или перегорание светодиодов. В комплекте с некоторыми пробниками поставляются эталонные схемы-карточки, вставляемые в индикаторную секцию для облегчения чтения показаний пробника.

Многоконтактные логические пробники позволяют наблюдать не только статические сигналы, но, например, и работу разрядов счетчика в динамике, хотя, конечно, получить с их помощью полезную информацию затруднительно, если используемая схема работает на частотах, превышающих пороговые частоты визуального восприятия.

15.4. ЛОГИЧЕСКИЙ КОМПАРАТОР

Логический компаратор предназначен для индикации различий сигналов на контактах проверяемой и эталонной интегральных микросхем. Конструктивно он выполнен так же, как и многоконтактный логический пробник, т. е. надевается на проверяемую ИС как клипса. Число ин-

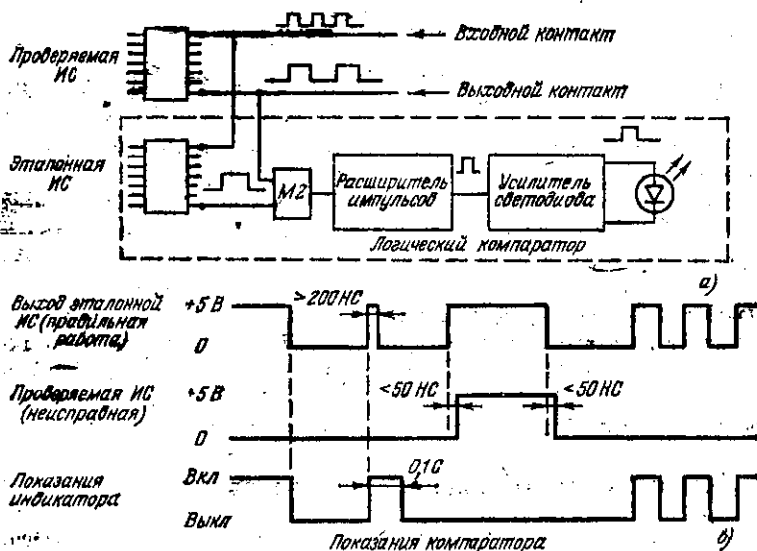


Рис. 15.3. Логический компаратор:

а — схема подключения компаратора; б — показания компаратора при различных значениях входных сигналов

дикаторов (светодиодов) компаратора равно числу контактов ИС. Включенное состояние индикатора соответствует несовпадению сигналов проверяемой и эталонной ИС, т. е. ошибке.

Эталонная ИС, установленная в компаратор, должна соответствовать типу проверяемой ИС. Компараторы обнаруживают не только постоянные неисправности, но и кратковременные динамические ошибки.

На рис. 15.3 приведена схема соединения компаратора и форма входных/выходных сигналов для различных случаев. Компараторы подсоединяют проверяемому и эталонную ИС параллельно к одним и тем же входным линиям, т. е. эталонная ИС приводится в действие теми же сигналами, что и проверяемая.

Выходы обеих ИС сравниваются с помощью схемы сложения по модулю 2. Разница в значениях выходных сигналов обеих ИС длительностью больше 200 нс воспринимается как ошибка и вызывает загорание индикатора.

Поскольку логические компараторы построены по принципу использования тех же входных сигналов, что и подаваемые на ИС, реально они могут определить только неисправный узел, а не дефектную ИС. Например, внутреннее короткое замыкание входного контакта некоторой микросхемы проявится как ошибка ИС, работающей на эту неисправную ИС, поскольку как неработоспособная ИС, так и компаратор будут иметь один и тот же входной сигнал.

15.5. ЛОГИЧЕСКИЙ ИМПУЛЬСНЫЙ ГЕНЕРАТОР

Для работы большинства логических пробников требуются те или иные средства изменения состояния входов логической схемы при одновременном контроле ее выхода.

Для выполнения этой функции предназначены логические импульсные генераторы [63]. Они подают на узлы схемы кратковременные сигналы, переключающие эти узлы в логические состояния, обратные имеющимся. Длительности импульсов составляют обычно десятые доли микросекунды и достаточно малы, чтобы испортить схему. Логические генераторы не могут переключить в противоположные состояния линию, непосредственно связанную с источником питания или землей; такие короткие замыкания в схеме можно обнаружить, если к одной и той же точке подключить генератор и пробник и не получить изменения логического состояния при подаче импульса.

В некоторых импульсных генераторах имеются ПЗУ, обеспечивающие возможность программного управления видом выходного сигнала при помощи кнопок. Например, выходные сигналы могут быть следующих типов: одиночные импульсы, непрерывная последовательность импульсов с частотой повторения 100 Гц, пачки из 100 импульсов, непрерывная последовательность импульсов с частотой повторения 10 Гц, пачки из десяти импульсов и непрерывная последовательность импуль-

сов с частотой повторения 1 Гц. Вид выходного сигнала устанавливается с пульта. На таких приборах обычно предусматривают светодиодный индикатор для контроля выходного сигнала генератора.

15.6. ИЗМЕРИТЕЛИ ТОКА

Измерители тока, выполненные в виде зондов, являются удобным прибором, позволяющим определить неисправный компонент схемы путем выявления цепей, по которым протекают токи, значения которых выходят за установленные пределы. Так, например, если сигнальная шина замкнута на землю, прибор может определить, где имеет место короткое замыкание — в источнике или в приемнике сигнала, даже в монтажных схемах И, ИЛИ. Прибор позволяет также обнаруживать замыкающие переключки из припоя и разрывы в печатных схемах.

Измеритель тока обычно используется вместе с импульсным генератором. Измерители тока используют принцип измерения магнитного поля, создаваемого сигналами, вырабатываемыми в проверяемой схеме или поступающими от внешнего импульсного генератора. Для индикации изменения логических уровней, одиночных импульсов и серии импульсов прибор имеет индикатор. Поскольку измеритель тока нечувствителен к напряжению, им можно пользоваться для проверки любых логических схем, в которых импульсы тока не превышают заданный предел по амплитуде и частоте.

В состав измерителей тока входят экранированный индуктивный чувствительный элемент тока и широкополосный усилитель с большим коэффициентом усиления, которые обеспечивают чувствительность, необходимую для обнаружения магнитных полей, создаваемых изменениями тока в печатных проводниках схемы. Чувствительность прибора бывает достаточно высокой для отслеживания сигналов в многослойных печатных платах. При работе прибор подносится сначала к опорной точке (обычно к выходу источника сигнала) и его чувствительность устанавливается такой, чтобы загорелся индикатор. Затем прибор перемещается вдоль цепи сигнала. При прохождении закороченной ветви индикатор гаснет.

15.7. ОСЦИЛЛОГРАФЫ

Осциллографы продолжают оставаться наиболее распространенным контрольно-измерительным прибором для цифровых схем. Ввиду все возрастающей частоты цифровых устройств разрабатываются осциллографы с расширенной полосой рабочих частот и уменьшенных габаритов. Появление новых малогабаритных осциллографов изменило представление о них — вместо стационарных они стали портативными.

Кроме того, осциллографы комбинируют с другими измерительными приборами для повышения их эффективности при эксплуатации, например с мультиметром, приставкой для подсчета логических переключений.

чений, цифровым индикатором для отсчета значений напряжений и временных параметров.

Осциллографы совершенствуют также путем добавления режима регулируемого послесвечения, что позволяет наблюдать одиночные и случайные сигналы путем запоминания осциллограммы в течение 3—15 мин. Регулировка послесвечения позволяет получать четкие осциллограммы трудных для наблюдения низкочастотных сигналов, которые обычно дают мерцающие осциллограммы, и коротких импульсов с крутыми фронтами, но низкой частотой повторения.

Некоторые осциллографы снабжаются сменными блоками переключателя и анализатора логических состояний, позволяющих получить новый режим отображения логических состояний, при котором результаты измерения отображаются на экране осциллографа в виде логических 1 и 0.

15.8. ЛОГИЧЕСКИЕ АНАЛИЗАТОРЫ

При поиске сложных неисправностей в дискретных устройствах возникает необходимость одновременного наблюдения нескольких сигналов, однократных и аperiodических, поведения схемы в моменты времени, предшествующие какому-либо событию или следующие за ним. Такие возможности представляет *логический анализатор* — прибор для сбора и анализа данных о реальных условиях работы дискретных устройств [73, 74].

Логический анализатор представляет собой комбинацию многоканального регистратора двоичных сигналов, построенного на базе быстросрабатывающего ЗУ с развитой системой управления процессом записи

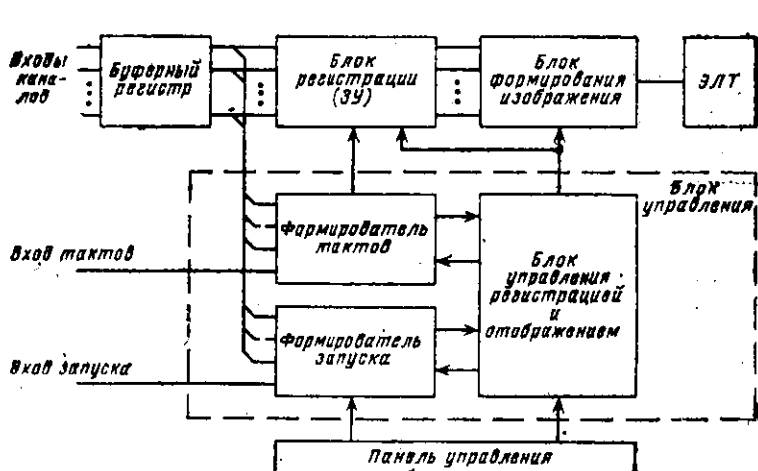


Рис. 15.4. Структурная схема логического анализатора

данных, и экранного пульта-дисплея, отображающего записанную в ЗУ информацию в форме, наиболее удобной для ее анализа.

Структурная схема логического анализатора приведена на рис. 15.4.

Различают два типа логических анализаторов: *анализаторы логических состояний* и *анализаторы временных диаграмм*.

Анализаторы логических состояний фиксируют состояния контрольных точек проверяемой схемы во время тактовых сигналов, задаваемых проверяемым устройством, и записывают процесс изменения состояний синхронно с его работой.

Анализаторы временных диаграмм фиксируют состояния контрольных точек проверяемой схемы в моменты времени, которые задаются независимо работающим тактовым внутренним генератором анализатора.

Состояния контрольных точек фиксируются в дискретные моменты времени (при подаче тактовых сигналов) в двоичной форме: 0 — при отсутствии сигнала, 1 — при его наличии.

Логические анализаторы имеют два основных режима: регистрации и отображения.

Регистрацией называется процесс записи состояния сигналов, поступающих по входным каналам анализатора, в его запоминающие устройства. Регистрация начинается по сигналу запуска регистрации, который может быть либо внешним сигналом, либо кодовым словом, либо последовательностью кодовых слов.

Отображением называется процесс индикации на экране электронно-лучевой трубки временных диаграмм или логических состояний, записанных в ЗУ в процессе регистрации.

Для установки режима, способа запуска регистрации, способа отображения, а также кодовых слов запуска регистрации на панели управления имеются переключатели и гнезда для подключения внешних сигналов.

Основными характеристиками логических анализаторов являются: количество каналов одновременной регистрации состояний сигналов;

уровни входных логических сигналов;

глубина регистрации, т. е. максимальное количество запоминаемой по одному каналу информации;

максимальная частота регистрации, определяющая минимальный интервал времени между двумя последовательными отсчетами состояний входных сигналов.

Глубина регистрации определяется емкостью ЗУ, а максимальная частота регистрации — быстродействием ЗУ анализатора.

Выпускаемые в настоящее время анализаторы имеют от 8 до 48 каналов регистрации, частоту регистрации от 20 до 200 МГц, глубину регистрации от 64 до 2048 бит на канале.

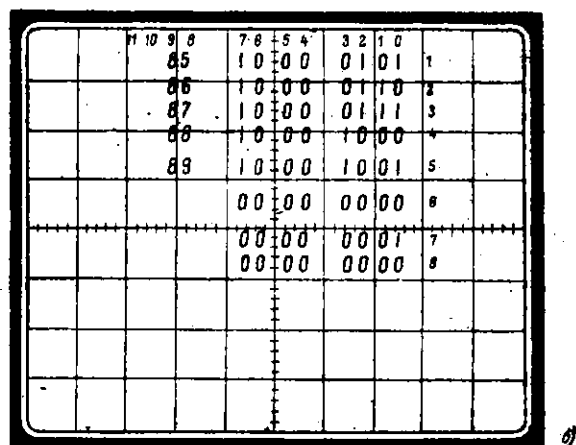
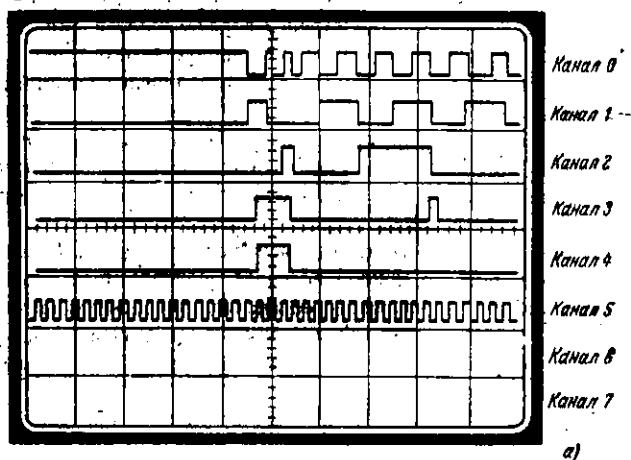


Рис. 15.5. Режимы логического анализатора:
а — отображение временных диаграмм; б — отображение логических состояний

Процесс регистрации в логическом анализаторе может быть начат при появлении на входах:

- специально заданного внешнего сигнала;
- заданной кодовой комбинации (слово состояния) сигналов;
- заданной последовательности кодовых комбинаций.

Существуют следующие способы запуска регистрации: *прямой, задержанный и с предустановкой* [73].

При прямом запуске сигнал запуска сразу включает процесс регистрации, а при задержанном — через определенное время, задаваемое числом тактов задержки. При запуске с предустановкой анализатор регистрирует состояния в контрольных точках диагностируемого устройства независимо от сигнала запуска и позволяет сохранить и выдать на отображение данные, которые были записаны за N тактов до появления сигнала и $M - N$ тактов после него, где M — максимальная глубина регистрации по одному каналу. Предустановка может иметь значение в пределах $1 < N < M$.

Если для запуска регистрации использовать канал, характеризующий появление сбоя, то прибор позволит записать предысторию появления этого события и на основе ее анализа установить причины сбоя.

Информация ЗУ анализатора может быть выведена на экран в форме логических временных диаграмм, логических таблиц или графического изображения.

В режиме отображения логических временных диаграмм анализатор функционирует как многоканальный цифровой запоминающий осциллограф, индицирующий двоичные сигналы состояний (рис. 15.5, а).

Для облегчения анализа записанной информации и сопоставления сигналов, принятых по различным каналам, на экран индикатора выводятся вспомогательные указатели — курсоры в виде вертикальной линии, которая может перемещаться по указанию оператора вдоль горизонтальных осей.

В режиме отображения логических таблиц информация индицируется в двоичном, восьмеричном, шестнадцатиричном или алфавитно-цифровом коде (рис. 15.5, б).

В графическом режиме отображения каждое из записанных слов состояний записывается на экране ЭЛТ в виде точки, координаты которой определяются разрядами слова состояния.

Глава шестнадцатая

ПРОЦЕССЫ ЭКСПЛУАТАЦИОННОГО ОБСЛУЖИВАНИЯ ЭВМ

16.1. ОБОРУДОВАНИЕ ПОМЕЩЕНИЙ ДЛЯ ЭВМ

В данной главе рассмотрены вопросы размещения ЭВМ, требования к помещению, перечень видов регламентных работ, меры по технике безопасности и др. Рассмотрение ведется применительно к ЭВМ единой системы. При написании главы использованы материалы инструкций по эксплуатации машин единой системы.

Для размещения средних и больших моделей ЭВМ единой системы требуются специально оборудованные помещения.

Собственно машины располагаются в машинном зале. В непосредственной близости от него располагают зал сервисной аппаратуры, аналитическую лабораторию, помещения для подготовки данных, хранения магнитных носителей данных, бумажных носителей данных, ЗИП и технологической документации, помещение для ремонта узлов ЭВМ. Необходимые площади для средней и большой моделей ЭВМ приведены ниже:

Назначение помещения	Площадь, м ² ЕС-1033	ЕС-1050
Машинный зал	110	270—300
Сервисная аппаратура	24	90
Ремонт узлов ЭВМ		20
Аналитическая лаборатория	20	30
Подготовка данных		40
Хранение магнитных носителей	10	20
Хранение бумажных носителей		20
Хранение ЗИП и технической документации		20
	20	

В машинном зале, помещениях сервисного и технического обслуживания, подготовки данных, хранения магнитных носителей параметры окружающей среды должны находиться в следующих пределах:

Температура окружающего воздуха	20±5 °С
Относительная влажность воздуха	65±15 %
Атмосферное давление	От 84 до 106,7 кПа (от 630 до 800 мм рт. ст.)
Уровень шумов	До 75 дБ
Запыленность, не более	1 мг/м ³ при размере частицы не более 3 мкм

Освещение машинного зала выполняется люминесцентными лампами. Освещенность должна быть 350—400 лк на высоте 0,8 м от пола.

В качестве примера на рис. 16.1 показано размещение основного оборудования ЭВМ ЕС-1050 в машинном зале [79]. Здесь использовано так называемое зонное размещение устройств (зона центральных устройств, зона внешних ЗУ, зона устройств ввода-вывода), что облегчает техническое обслуживание и использование резервного оборудования.

Для обеспечения необходимых параметров окружающей среды в машинном зале и других помещениях ЭВМ используются приточно-вытяжные системы кондиционирования.

В машинном зале пол выполняется двойным (зазор между полами равен или более 250 мм) и потолок монтируется подвесным. Между технологическим полом и подвесным

потолком должно быть расстояние не менее 2,8 м. Стены и подвесной потолок покрываются звукопоглощающими плитами.

Пространство между основным и технологическим полами используется для прокладки силовых кабелей, жгутов сигнальных цепей, защитного и схемного заземления, а так-

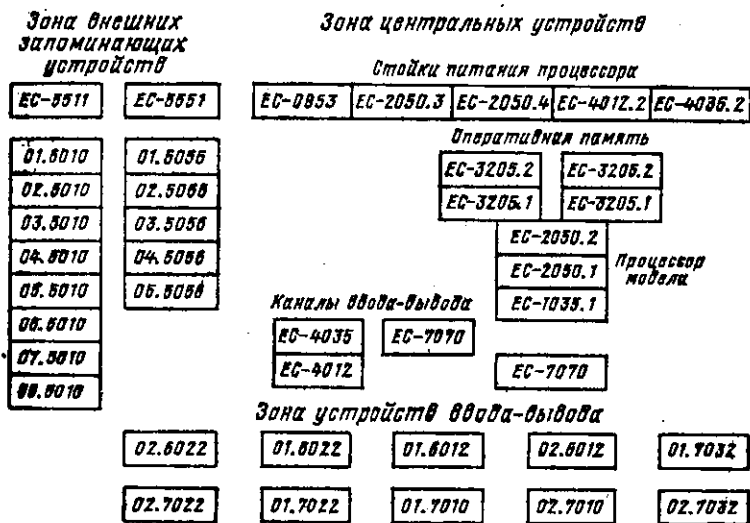


Рис. 16.1. Размещение основного оборудования ЭВМ ЕС-1050 в машинном зале

же в качестве приточного вентиляционного канала. Пространство между основным и подвесным потолками служит в качестве вентиляционного канала. В этом пространстве размещаются воздуховоды, осветительные устройства и противопожарная аппаратура.

При использовании центральной проточно-вытяжной вентиляции нагретый в устройствах воздух удаляется либо путем выброса в машинный зал, либо забором из стоек с помощью подсоединенных к ним специальных воздуховодов. Некоторые варианты распределения охлаждающего воздуха в машинном зале представлены на рис. 16.2, а — в.

Воздух, подаваемый в устройства ЭВМ центральной приточно-вытяжной вентиляции, должен иметь температуру от 14 до 16°C и максимальную запыленность не более 0,75 мг/м³ при размере частиц не более 3 мкм. Отметим,

что при отключенной центральной приточно-вытяжной вентиляции устройства ядра ЭВМ — процессор, каналы, основная память могут работать не более 20 мин (при работающей автономной вентиляции).

Особого внимания требует создание определенных условий для хранения бумажных (перфокарт) и магнитных

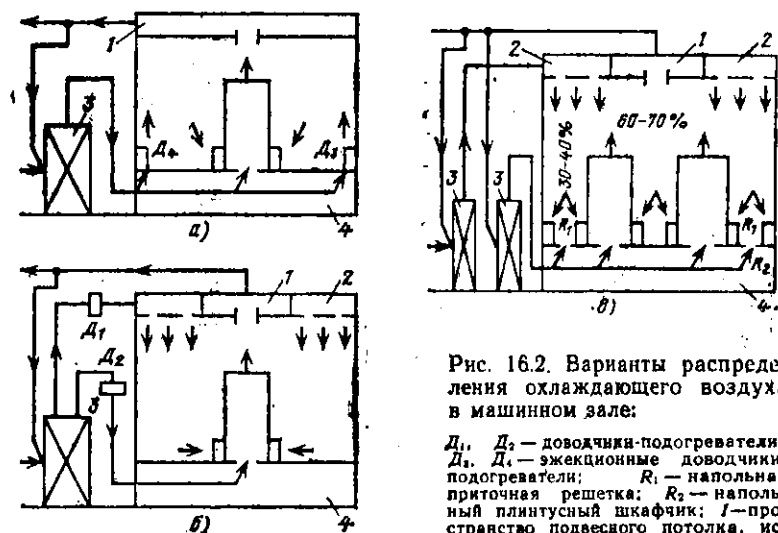


Рис. 16.2. Варианты распределения охлаждающего воздуха в машинном зале:

D_1 , D_2 — доводчики-подогреватели; D_1 , D_2 — эжекционные доводчики-подогреватели; R_1 — напольная приточная решетка; R_2 — напольный плинтусный шкафчик; 1 — пространство подвесного потолка, используемого в качестве вытяжного вентиляционного канала; 2 — пространство технологического пола

пространство подвесного потолка, используемого в качестве вытяжного вентиляционного канала; 3 — конденсатор; 4 — пространство технологического пола

(магнитных лент) носителей. Перфокарты следует хранить под двухкилограммовыми прессами. Коробки с магнитными лентами хранятся в вертикальном положении. В хранилищах магнитных лент следует поддерживать температуру $21-22^{\circ}\text{C}$, влажность 45—55 %.

В качестве первичного источника питания ЭВМ служит силовой трансформатор и трехфазная четырехпроводная сеть 380/220 В, частотой 50 Гц с глухозаземленной нейтралью. Трансформатор допускает регулирование напряжения под нагрузкой. Напряжение первичного источника питания не должно отклоняться от номинала более чем на ± 10 и ± 15 %, а частота не более чем на ± 1 %. Требования к мощности первичного источника питания зависят от модели и комплектации ЭВМ. Например, для ЭВМ ЕС-1050 требуемая мощность составляет 100 кВт.

16.2. ТЕХНИКА БЕЗОПАСНОСТИ ПРИ РАБОТЕ С ЭВМ

В инструкциях по эксплуатации ЭВМ ЕС указываются следующие меры безопасности.

В помещении ЭВМ должна быть проложена шина защитного заземления сечением не менее 120 мм^2 . Участки шин должны свариваться внахлест на длине не менее двойной ширины шины. Шина защитного заземления соединяется с заземлением первичного источника питания — трехфазной четырехпроводной сети 380/220 В, 50 Гц. Нейтраль электроустановки (трансформатора) первичного питания глухо заземляется с помощью заземляющего устройства с сопротивлением заземления не более 600 мОм.

Корпуса всех устройств ЭВМ должны иметь надежное электрическое соединение с шиной защитного заземления в машинном зале. Переходное сопротивление между непосредственно соединенными между собой металлическими деталями не должно превышать 600 мОм, а суммарное переходное сопротивление при заземлении аппаратуры через перемычки, шины и т. п. должно быть не более 2000 мОм.

Сопротивление изоляции электрических цепей с напряжением 380/220 В между собой и относительно корпуса (болта заземления) любого устройства должно быть не менее 20 МОм.

Обслуживающий ЭВМ персонал должен всегда помнить, что опасность представляют первичные цепи блоков питания, подключенные к трехфазному напряжению 380/220 В, 50 Гц.

К работе с ЭВМ допускаются лица, изучившие документацию на систему электропитания и прошедшие инструктаж по технике безопасности при эксплуатации установок с напряжением до 1000 В. Лица, обслуживающие систему электропитания, должны быть обучены приемам освобождения попавшего под напряжение, приемам искусственного дыхания, правильного оказания первой помощи при поражении электрическим током и способам гашения пожара в электроустановках.

При эксплуатации ЭВМ запрещается:

- включать ЭВМ при неисправной защите электропитания;

- подключать и отключать разъемы кабелей электропитания и блоков вентиляции при поданном напряжении электросети;

- заменять съемные элементы под напряжением;

- производить пайку в аппаратуре, находящейся под напряжением;

снимать крышки, щиты, закрывающие доступы к токоведущим частям;

пользоваться электрическими паяльниками и другими электроинструментами с напряжением 36 В и выше с незаземленными корпусами.

При техническом обслуживании аппаратуры ЭВМ обязательным является выполнение следующих правил:

при проверке и обслуживании съемных блоков электропитания их корпуса должны заземляться;

измерение напряжений в токоведущих частях с напряжением более 36 В необходимо производить, пользуясь резиновыми ковриками и изолированными щупами;

при прозвонке электрических цепей необходимо предварительно эти цепи обесточить и проверить отсутствие напряжения с помощью вольтметра. Прозвонку схем ТЭЗ производить прибором с источником постоянного напряжения не более 1,5 В;

металлические корпуса приборов следует заземлять.

16.3. ОБЕСПЕЧЕНИЕ ПОЖАРНОЙ БЕЗОПАСНОСТИ ВЫЧИСЛИТЕЛЬНЫХ ЦЕНТРОВ

Пожарная безопасность вычислительных центров обеспечивается [77]:

выполнением требований пожарной безопасности помещений для ЭВМ, хранилищ информации, установок кондиционирования и систем электропитания;

выполнением правил пожарной безопасности при ремонтно-профилактических работах;

системой автоматической пожарной сигнализации и пожаротушения.

К помещениям для ЭВМ, хранилищам информации, установкам кондиционирования и системам электропитания предъявляются следующие требования пожарной безопасности:

материалы для акустической отделки стен и потолков должны быть по возможности негорючими;

древесностружечные и древесноволокнистые плиты должны применяться только при их глубокой пропитке огнезащитными составами;

стальные несущие и ограждающие конструкции должны быть защищены огнезащитными материалами и красками;

конструкция съемного пола машинного зала должна обеспечивать доступ к кабелям и вентиляционным системам;

плиты съемного пола должны быть выполнены из негоряемых или трудногоряемых материалов;

опоры и стойки съемных полов должны быть негоряемыми;

подпольные пространства под съемными полами должны разделяться негоряемыми диафрагмами;

различные помещения должны отделяться друг от друга по мере возможности противопожарными стенами и перегородками;

в надпотолочном и подпольном пространствах должны устанавливаться пожарные извещатели, система трубопроводов и выпускных устройств для подачи огнетушительного состава;

в помещениях, где нет постоянного присутствия персонала, необходимо устанавливать автоматическую систему пожарной защиты;

для нормальной эвакуации людей во время пожара ширина дверей должна быть не менее 1,5 м, высота не менее 2 м, ширина коридоров не менее 1,8 м;

машинные залы площадью более 250 м² должны иметь не менее двух выходов;

пожар в любом помещении не должен отрезать пути к эвакуационным выходам;

запасы перфолент и перфокарт в машинных залах не должны превышать оперативных норм;

воздуховоды системы вентиляции должны быть выполнены из негорючих материалов, иметь небольшое число поворотов и гладкую поверхность стенок, так как накопленная в них пыль может воспламеняться;

воздуховоды должны периодически очищаться;

в системе вентиляции должны быть предусмотрены клапаны для перекрытия воздуховодов при пожаре;

теплоизоляция воздуховодов систем охлаждения ЭВМ должна быть выполнена из негоряемых материалов;

при возникновении пожара системы вентиляции должны отключаться.

При выполнении ремонтно-профилактических работ необходимо соблюдать следующие правила пожарной безопасности:

перед началом профилактических работ необходимо проводить инструктаж по пожарной безопасности, правилам работы с горючими жидкостями;

съемные узлы ЭВМ необходимо ремонтировать в отдельном, специальном помещении;

необходимо использовать низковольтные паяльники, устанавливаемые на несгораемой подставке;

в качестве моющих средств необходимо выбирать пожаробезопасные вещества;

если для промывки деталей необходимо пользоваться горючими жидкостями, то эти работы должны выполняться в специальном помещении с приточно-вытяжной вентиляцией;

горючие жидкости необходимо хранить в металлической, плотно закрывающейся таре и убирать по окончании работы в сейф;

нельзя устанавливать ящики с жидкостями у входов, выходов, на эвакуационных путях, около установок и приборов отопления;

нельзя оставлять без наблюдения включенную в сеть контрольно-измерительную аппаратуру;

нельзя оставлять на устройствах остатки проводов, вату, марлю и другой обтирочный материал;

запрещается применять электронагревательные бытовые приборы;

запрещается курить, включать и выключать электросеть и приборы во время работы с легковоспламеняющимися жидкостями.

Система автоматической пожарной сигнализации и пожаротушения содержит: пожарные извещатели; приемную станцию; схему подрыва пиропатронов; батарею для хранения и выпуска огнетушащего состава; систему трубопроводов для подачи огнетушащего состава; баллон и распределитель сжатого воздуха. Система работает следующим образом. При повышении концентраций дыма или температуры в помещении извещатели срабатывают и выдают импульс на приемную станцию и далее на схему подрыва пиропатронов клапанов распределительного устройства и автоматической головки-затвора баллона сжатого воздуха. Через вскрывшуюся головку воздух вскрывает головки баллонов с огнетушительным составом, который поступает через систему трубопроводов к выпускным устройствам.

Различают два типа пожарных извещателей: пожарные извещатели, реагирующие на дым; тепловые пожарные извещатели.

Первые отличаются более высокой чувствительностью.

В качестве огнетушащих составов используют неэлектропроводные, не вызывающие коррозии и порчи оборудования составы. Наиболее широко используются составы с углекислотой и фреоном.

Для локального тушения пожаров используются также стационарные и передвижные огнетушители с ручным пуском.

16.4. ПРОЦЕССЫ ПЛАНОВО-ПРОФИЛАКТИЧЕСКОГО ОБСЛУЖИВАНИЯ

Целью планово-профилактического обслуживания является обеспечение эксплуатации ЭВМ с коэффициентом использования, оговоренном в технической документации. Для машин ЕС ЭВМ принят коэффициент использования $K_{т.и} = 0,9$ при круглосуточной работе.

В процессах планово-профилактического технического обслуживания определяются и заменяются элементы, параметры которых вследствие износа находятся на пределе допустимого, проверяются элементы и узлы, не охватываемые системой автоматического контроля ЭВМ.

Процессы планово-профилактического технического обслуживания выполняются периодически через определенные интервалы времени, причем чем больше установленный интервал между повторениями процесса обслуживания, тем больше программа работ по обслуживанию и больше продолжительность самого процесса обслуживания. В табл. 16.1 приведены некоторые сведения о процессах планово-профилактического технического обслуживания ЭВМ.

Т а б л и ц а 16.1. Процессы планово-профилактического технического обслуживания

Наименование процесса планово-профилактического обслуживания	Интервал между обслужи- ваниями	Нормативная продол- жительность процесса обслуживания	
		ЕС-1033	ЕС-1060
Суточный	Сутки	1	1
Двухнедельный	Две недели	4	6
Месячный	Месяц	8	16
Полугодовой	Полгода	72	72

Как следует из таблицы, продолжительность обслуживания зависит от модели ЭВМ.

Продолжительность процессов планово-профилактического обслуживания отдельных устройств (технических средств) ЭВМ составляет для ЕС ЭВМ: суточного — 1 ч, двухнедельного — 4 ч, месячного — 12 ч, полугодового — 24 ч.

При вводе ЭВМ в эксплуатацию выполняется полугодовой процесс технического обслуживания.

Рассмотрим перечень работ, выполняемых в процессах планово-профилактического технического обслуживания.

Суточное техническое обслуживание: 1) внешний осмотр технических средств ЭВМ; 2) чистка устройств; 3) проверка температуры и влажности в помещении; 4) работы суточного регламента отдельных технических средств; 5) проверка ЭВМ с помощью комплекса программ технического обслуживания.

Двухнедельное техническое обслуживание. По сравнению с программой суточного обслуживания добавляется осмотр креплений разъемов и установки ТЭЗ, выполняются работы двухнедельного регламента отдельных технических средств машины. Обслуживание завершается проверкой ЭВМ с помощью комплекса программ технического обслуживания.

Месячное техническое обслуживание. Сверх программы двухнедельного обслуживания проверяется работа органов управления и сигнализации, функционирование ЭВМ при изменении на $\pm 5\%$ напряжений вторичных источников питания. Обслуживание заканчивается проверкой ЭВМ с помощью комплекса программ технического обслуживания при номинальном напряжении.

Полугодовое техническое обслуживание. Сверх программы месячного обслуживания производится осмотр и чистка ТЭЗ, блоков питания, кабелей и жгутов, проверка заземления путем замера переходного сопротивления, выполняются работы полугодового регламента отдельных технических средств, проверка защиты системы электропитания и сигнализации о перегреве, замер и регулировка частоты задающего генератора. Обслуживание завершается проверкой функционирования ЭВМ при изменении на $\pm 5\%$ напряжений вторичных источников питания и прогонкой при номинальном напряжении комплекса программ технического обслуживания.

Процесс технического обслуживания закончен, если не было сбоев при выполнении комплекса программ технического обслуживания.

16.5. ВЕДЕНИЕ ЖУРНАЛА ЭКСПЛУАТАЦИИ ЭВМ

При эксплуатации ЭВМ должен производиться сбор данных о работе ЭВМ, отдельных ее устройств, о возникающих отказах и сбоях, о проведенных профилактических работах и их результатах.

В гл. 13 этот вопрос был рассмотрен под углом зрения автоматизации процесса накопления и обработки информации о машинных

ошибках. В данной главе рассматривается ведение «аппаратурного журнала», являющегося первичной формой информации о работе ЭВМ. Эта информация, зафиксированная по определенной форме в виде сводных карточек учета неисправностей, ежеквартально высылается пользователем-владельцем ЭВМ предприятию — изготовителю машины. Текущие данные об эксплуатации ЭВМ, наработке машины и отдельных устройств, наблюдавшихся отказах и сбоях, установленных причинах отказов и сбоев, о проведенных ремонтных работах и регулировках должны тщательно фиксироваться обслуживающим персоналом машины в аппаратурном журнале установленной формы.

Сбор и накопление данных о поведении ЭВМ в процессе эксплуатации имеет большое значение как для организации (разработчика и предприятия) — изготовителя, так и для пользователя, эксплуатирующего машину.

Разработчики и изготовители ЭВМ на основе данных об отказах и сбоях имеют возможность определить статистические характеристики надежности и обслуживаемости ЭВМ и ее отдельных устройств, определить и соответствующим образом доработать схемы, конструкцию и технологию изготовления узлов и устройств, оказавшихся недостаточно надежными в эксплуатации, скорректировать техническую документацию, в том числе уточнить методику и программы профилактических испытаний.

Пользователь на основе данных о сбоях и отказах ЭВМ имеет возможность планировать проведение ремонтных работ, замену отдельных устройств более надежными, установку резервного оборудования. Данные о сбоях, поведении ЭВМ при профилактических испытаниях с изменением питающих напряжений позволяют прогнозировать возникновение отказов в будущем и принимать профилактически необходимые меры (замену узлов и т. п.).

Для соответствующего учета неисправностей при работе ЭВМ принципиальное значение имеет уточнение классификации возникающих при работе машины событий как сбоев или отказов.

Понятия отказа и сбоя были определены в общеупотребительном смысле в гл. 1. Однако практика организации учета данных о надежности ЭВМ потребовала следующих дополнительных соглашений о том, как классифицировать те или иные события при эксплуатации машины.

Следует различать сбой и отказы ЭВМ и сбой и отказы отдельных ее устройств.

Обрыв магнитной ленты или бумаги, замятие перфокарты, пробивка лишних отверстий на перфокартах и перфолентах, непропечатывание знака классифицируются не как сбой ЭВМ, а как сбой соответствующего устройства. Однако если для устранения указанных явлений необходима регулировка или ремонт устройства, то регистрируется его отказ.

Если при отказе некоторого устройства оказалось возможным подключить к машине резервное устройство (или перейти к работе с резервным устройством) и продолжить решение задачи, то фиксируется не отказ ЭВМ, а отказ устройства. Если указанное переключение влечет за собой новую загрузку операционной системы, то регистрируется отказ ЭВМ.

Сбои ЭВМ, автоматически устраняемые программно-аппаратурными средствами, не учитываются при определении значений наработки на сбой и коэффициента использования ЭВМ. Отметим, что данные о таких сбоях, автоматически накапливаемые и обрабатываемые операционной системой машины, ориентируют пользователя относительно «слабых мест» в эксплуатируемой машине, позволяют по изменению частоты появления таких сбоев прогнозировать надежность работы машины.

Сбои ЭВМ, обнаруженные и устраненные при техническом обслуживании, не учитываются при определении характеристик надежности. Аналогичным образом поступают с отказами и сбоями, произошедшими из-за низкого качества носителей информации, отклонения параметров внешней среды за пределы, оговоренные в технической документации, а также вызванными ошибками в действиях операторов.

16.6. ЭКСПЛУАТАЦИОННАЯ ДОКУМЕНТАЦИЯ

Основными эксплуатационными документами ЭВМ являются:

- формуляр ЭВМ;
- инструкция по эксплуатации ЭВМ и устройств;
- руководства по эксплуатации диагностических и функциональных тестов;
- диагностические справочники;
- таблицы и тесты проверки ТЭЗ;
- техническое описание.

Два других документа: «Требования к помещению» и «Инструкция по монтажу, пуску и регулированию ЭВМ на месте применения», используются при подготовке помещения и вводе ЭВМ в эксплуатацию у пользователя.

Требования к помещению определяют необходимую площадь машинного зала и подсобных помещений, климатические условия, вентиляцию и другие параметры, необходимые для обеспечения нормальных условий эксплуатации.

Инструкция по монтажу, пуску и регулированию ЭВМ на месте применения содержит правила развертывания, установки, монтажа и пуска ЭВМ в эксплуатацию на месте применения. Основным разделом инструкции является «Сдача в эксплуатацию смонтированного изделия», в котором приводится перечень испытаний установленной ЭВМ,

гарантирующих ее работоспособность. В этом разделе даются также определения понятий сбоя и отказа, которыми пользуются при учете результатов испытаний.

В результате успешных испытаний подписывается акт ввода ЭВМ в эксплуатацию, форма которого указана в инструкции.

В инструкции приводятся также правила возмещения стоимости узлов, деталей и электрорадиоэлементов, вышедших из строя в период проведения пуско-наладочных работ и пуска ЭВМ.

Формуляр ЭВМ является документом, содержащим комплектность аппаратурных средств ЭВМ и ее эксплуатационной документации, технические характеристики, данные контрольно-приемочных испытаний и гарантийные обязательства завода-изготовителя.

Инструкция по эксплуатации содержит сведения по установке ЭВМ, включению электропитания, подготовке ЭВМ к работе, составу и численности обслуживающего персонала, инструкции по выполнению всех операций с пульта ЭВМ. Основными разделами инструкции по эксплуатации являются разделы, содержащие описание характерных неисправностей и методы их устранения, рекомендации по планово-профилактическим работам, ремонту ЭВМ, организации ЗИП. В этих разделах содержатся инструкции по запуску диагностических тестов, поиску неисправностей, указания по применению тех или иных тестовых средств. В разделе по организации планово-профилактических работ содержатся рекомендации по периодичности, длительности и составу планово-профилактических работ.

Руководства по эксплуатации диагностических и функциональных тестов содержат инструкции оператору по запуску и интерпретации результатов этих тестов, а также рекомендации по локализации неисправностей с помощью программно-аппаратурных средств.

Диагностические справочники представляют собой документ, в котором устанавливаются соответствия между кодами, получаемыми при выполнении диагностических тестов, и блоками, подозреваемыми на неисправность.

Таблицы и тесты проверки ТЭЗ содержат перечень тестовых наборов, обеспечивающих локализацию неисправности в ТЭЗ, и эталонные результаты. В ЭВМ, снабженных средствами автоматизированной проверки ТЭЗ такими, как автотестер ЭВМ ЕС-1045, дополнительно к таблицам проверки ТЭЗ имеются диагностические таблицы и тесты проверки ТЭЗ, записанные на кассетной магнитной ленте.

Качество эксплуатационной и технической документации, ее форматы и используемый носитель информации оказывают большое влияние на обслуживаемость и ремонт-пригодность устройств ЭВМ, особенно, периферийных.

Существующая техническая документация во многом далека от совершенства. Производители средств вычислительной техники ведут поиск новых форм эксплуатационной документации, позволяющей улучшить и облегчить обслуживание и ремонт.

16.7. ПРОЦЕДУРЫ АУТОНОМНОГО ТЕСТИРОВАНИЯ ЭВМ С ПОМОЩЬЮ ПРОГРАММЫ «ТЕСТ-МОНИТОР» (ЕС ЭВМ)

Комплект программ технического обслуживания (КПТО), включающий тест-монитор (ТМ) и тест-секции (ТС) устройств, используется при автономном тестировании устройств ЭВМ. Комплект программ технического обслуживания поставляется пользователю на магнитной ленте.

Тест-монитор загружается процедурой первоначальной загрузки программ, после чего выполняется его инициализация (настройка) на систему, т. е. определяются системные устройства ввода, вывода сообщений и загрузки тест-секций.

В качестве системных устройств ТМ обычно используются:

- пишущая машинка ПМ или пульт оператора для ввода заданий;

- пишущая машинка (или пульт оператора или АЦПУ) для вывода сообщений;

- накопитель на магнитной ленте для загрузки ТМ и тест-секций.

После определения системных устройств тест-монитор выдает начальное сообщение:

ОЖД НЗП ТМЕС ВЕРСИЯ 0

и переходит в состояние ожидания для приема:

- входных сообщений с клавиатуры ПМ;

- задания с пульта управления системы;

- изменения конфигурации системы с пульта управления.

Если устройство ввода не было определено, ТМ назначает в качестве устройства ввода ПМ, с которой получен запрос «Внимание».

В этот момент выходные сообщения могут использоваться для:

- а) изменения конфигурации путем ввода сообщений U (изменения параметров процессора), A (добавления уст-

роЙств в таблицу определения устройств) и W (исключений устройств из таблицы определения устройств);

б) назначения или изменения системных устройств ТМ путем ввода сообщений ИССУУ (назначение устройств ввода), ИОССУУ (назначение устройств вывода), или ИЛССУУ (назначение устройства загрузки) где СС и УУ — шестнадцатиричные номера канала и устройства;

в) ввода задания.

Задание, описывающее функции, требуемые для выполнения со стороны ТМ, представляет собой набор так называемых *приказов*.

Существуют следующие категории приказов:

а) управляющие работой тест-секций:

С — запрос на зацикливание примера, секции или задания;

F — запрос на прекращение выполнения секции;

G — запрос на прекращение выполнения активной секции только при текущей обработке;

O — запрос на вмешательство оператора;

S — запрос на включение управляющих ключей секции;

Z — запрос на выключение управляющих ключей секции;

б) управляющие работой ТМ:

B — запрос на начало или продолжение;

H — запрос на останов;

I — запрос на инициализацию ТМ;

S/Z — запрос на включение/выключение управляющих ключей ТМ;

в) действующие на таблицы ТМ:

A — запрос на добавление, изменение или восстановление компонентов таблицы определения устройств (UDT);

I — запрос на инициализацию таблиц ТМ или на изменение таблицы определения устройств ТМ (DMIO);

M — запрос на распечатку таблиц ТМ;

U — запрос на определение параметров процессора в таблице тест-секций (SRT);

W — запрос на исключение компонента таблицы определения устройств (UDT);

г) для ввода и запуска задания:

B — запрос на инициализацию ТМ для нового задания;

L — запрос на загрузку тест-секций;

д) выполняющие сервисные функции:

E — запрос на ввод данных в ОП с клавиатуры ПМ;

T — запрос на распечатку области ОП на устройстве вывода.

Пример. I/A50,0380/L520,0380/SO. S 24/V

Запись означает: инициализировать ТМ, добавить в таблицу UDT компонент для НМЛ с адресом 0380; загрузить секцию 520 и проверить НМЛ с адресом 0380; включить управляющий ключ 24ТМ (блокировка печати сообщений).

В процессе работы тест-монитор может выдавать следующие коды сообщений:

ОВВ — ошибка операции ввода-вывода ТМ;

ТВО — требуется вмешательство оператора;

ОВП — ошибочное внешнее прерывание;

ОСТ — останов по требованию секции;

ОВС — ошибочный вызов супервизора;

ПСК — прерывание от схем контроля ЭВМ;

ПГМ — ошибочное программное прерывание;

ПНН — программа не найдена;

ОТВ — секция отвергнута;

ПЗГ — необходима перезагрузка;

НУТ — нет требуемого типа устройства;

ПВС — повторить сообщение;

Н — начало тест-секции;

К — конец тест-секции;

НМТ UDT — таблица UDT заполнена (нет места);

НМТ IMT — таблица IMT заполнена (нет места);

ПНУ — прерывание от неназначенного устройства.

В ТМ и тест-секциях имеются управляющие ключи (УКТ), состояние которых задает различные режимы работы монитора и секции. Ниже приводится описание ключей ТМ, которые удобно использовать при поиске неисправностей и наладочных работах.

Ключи устанавливаются (переключаются) приказами S(Z), вводимыми с ПМ:

УКТ 17 включен (выключен) — ожидание для запуска программы SEREP при прерывании от схем контроля (нормальная обработка прерываний от схем контроля);

УКТ 18 включен (выключен) — блокировка распечатки (распечатка) диагностической области при прерывании от схем контроля;

УКТ 20 включен (выключен) — останов (выполнение) после загрузки каждой тест-секции;

УКТ 21 включен (выключен) — вывод только первой строки (всех строк) сообщений, печатаемых по запросам;

УКТ 22 включен (выключен) — блокировка (разрешение) печати сообщений на ПМ, если она не является системным устройством вывода;

УКТ 24 включен (выключен) — блокировка (разрешение) печати всех сообщений о начале (Н) и окончании (К) тест-секций на системном устройстве вывода;

УТК 25 включен (выключен) — останов (отсутствие останова) после каждой ошибки, обнаруженной монитором или тест-секцией;

УКТ 26 включен (выключен) — задание зацикливается (не зацикливается);

УКТ 27 включен (выключен) — блокировка (разрешение) печати всех сообщений на системном устройстве вывода;

УКТ 28 включен (выключен) — зацикливание (незацикливание) активной (загруженной в ОП) тест-секции;

УКТ 29 включен (выключен) — блокировка (разрешение) печати всех оперативных сообщений тест-секции, печатаемых по запросу SVC DO;

УКТ 30 включен (выключен) — блокировка (разрешение) печати всех сообщений об ошибке.

16.8. ПРОЦЕДУРЫ НЕАВТОНОМНОГО ТЕСТИРОВАНИЯ ПЕРИФЕРИЙНЫХ УСТРОЙСТВ ПОД УПРАВЛЕНИЕМ ОС ЕС

Для эффективного использования машинного времени необходимо обеспечить возможность проверки функционирования ПУ параллельно с выполнением машинной программы пользователей ЭВМ. Таковую возможность предоставляют в ЕС ЭВМ процедуры неавтономного тестирования ПУ. Тестирование под управлением ОС ЕС рекомендуется использовать при проверке готовности устройств, их исправности после ремонта, профилактической проверке устройств и поиске в них неисправностей.

Комплект неавтономных тестов устройств (КНТУ) состоит из управляющей программы и тест-секций, поставляемых пользователю отдельно от операционной системы. Магнитная лента КНТУ содержит отдельные файлы, представляющие собой объектные модули тест-секций ПУ в виде, удобном для помещения их в библиотеку загрузочных модулей.

Для тестирования с помощью программ неавтономного тестирования необходимы библиотеки, содержащие загрузочные модули и процедуры, а также библиотеки конфигураторов.

Конфигуратор ПУ определяет конкретный допустимый режим тестирования, а также некоторые технические характеристики ПУ.

Вызов программы неавтономного тестирования выполняется стандартной процедурой, поэтому она обычно каталогизируется и помещается в системную библиотеку каталогизированных процедур.

Для выполнения тестирования ПУ должно быть переведено в режим, недоступный программе пользователя. С этой целью рекомендуется переводить ПУ в автономный режим поочередно, так как заблаговременный перевод ПУ в автономный режим может затруднить выполнение других заданий.

При включении управляющей программы неавтономного тестирования в библиотеку каталогизированных процедур, она может вызываться командой START.

После вызова управляющей программы неавтономного тестирования, она запрашивает тестовое задание, содержащее адреса тестируемых ПУ, обозначение выполняемых тест-секций и режимы работы при тестировании.

Имеются следующие режимы работы при тестировании: запись всех сообщений тест-секций, выводимых на пульт управления, в выходной набор данных диагностики;

печать сообщений о начале и конце выполнения каждой тест-секции;

вывод на печать подробной диагностической информации о неисправности, обнаруженной в ПУ. Диагностическое сообщение содержит заголовок, признак обнаружения неисправности, имя тест-секции, номер блока, адреса начальной команды тест-секции, адрес программы канала, с помощью которой производилось тестирование, адрес тестируемого ПУ, результаты (ожидаемые и полученные данные в виде кода условия, сформированного в результате выполнения операции ввода-вывода, код приказа ПУ, конечный адрес программы канала, байты состояния ПУ и канала, значение счетчика в управляющем слове канала, байты уточненного состояния, байты данных);

останов теста при обнаружении в ПУ первой ошибки и предоставление оператору возможности изменения тестового задания. В ответ на сообщение оператор может либо продолжить тестирование, либо зациклить выполнение теста на обнаруженной ошибке;

циклическое повторение тестового задания. Этот режим используется для осциллографирования при поиске неисправностей;

л-кратное повторение блока тест-секции, на котором была обнаружена ошибка.

16.9. ОСОБЕННОСТИ ЭКСПЛУАТАЦИИ ОПЕРАЦИОННЫХ СИСТЕМ¹

Современные операционные системы, как правило, являются системами, ориентированными на разнообразные применения. В основе построения операционных систем лежит модульный принцип, позволяющий эффективно приспособлять операционные системы к изменяющимся условиям функционирования, а также обеспечивающий возможность их непрерывного совершенствования.

Модули представляют собой составные части операционной системы, которые могут объединяться в различных сочетаниях и образовывать уникальную операционную систему.

Различают следующие типы модулей:
необходимые в любой операционной системе;
необязательные (дополнительные);
альтернативные, реализующие одинаковые функции с разной степенью эффективности.

Таким образом, для успешной эксплуатации операционных систем необходима реализация следующих мероприятий:

определение назначения вычислительной системы, общего объема и типов выполняемых работ;

выбор конфигурации технических средств и соответствующего типа операционной системы;

осуществление генерации оперативной системы, т. е. настройки ее компонентов на конкретное применение;

проверка работоспособности сгенерированной операционной системы;

адаптация операционной системы к конкретным условиям;

фиксация нарушений в работе ЭВМ, анализ причин нарушений и принятие мер по их ликвидации;

разработка средств, расширяющих возможности стандартных версий операционных систем.

Генерация операционных систем. Операционные системы ЭВМ общего назначения, к которым относятся машины ЕС ЭВМ, представляют собой параметризуемые программные средства, созданные по модульному принципу. В соответствии с возможностями конкретной вычислительной установки (объем основной памяти, состав периферийного оборудования

¹ В написании этого параграфа принимал участие инж. Л. И. Нейштадт.

ния и др.), а также в зависимости от характера выполняемых задач пользователь может выбрать необходимые ему программные средства.

Генерация операционной системы представляет собой процесс объединения выбранных модулей операционной системы в системные наборы данных. Совокупность указанных системных наборов образует операционную систему данной установки. Новая операционная система состоит, как правило, из стандартных средств, присутствующих в любой системе, и некоторых дополнительных средств, задаваемых пользователем.

Рассмотрим процесс генерации системы на примере ОС ЕС ЭВМ.

В ОС ЕС пользователь описывает необходимый ему вариант операционной системы с помощью специальных макрокоманд генерации, в которых отражаются конфигурация вычислительной установки и ее программные возможности. При осуществлении генерации используются системные программы генерации.

Генерация ОС проводится в две стадии (рис. 16.3). На первой стадии анализируется последовательность макрокоманд, используемая для создания потока заданий. На второй стадии происходит выполнение заданий, в результате чего на устройствах прямого доступа создаются библиотеки системных модулей, которые и составляют операционную систему пользователя.

Первая стадия генерации состоит из двух фаз. Во время первой фазы проверяются макрокоманды, описывающие генерируемую ОС. В процессе проверки выдаются диагностические сообщения. Если во время первой фазы ошибок не обнаружено, то во время второй фазы создается поток заданий.

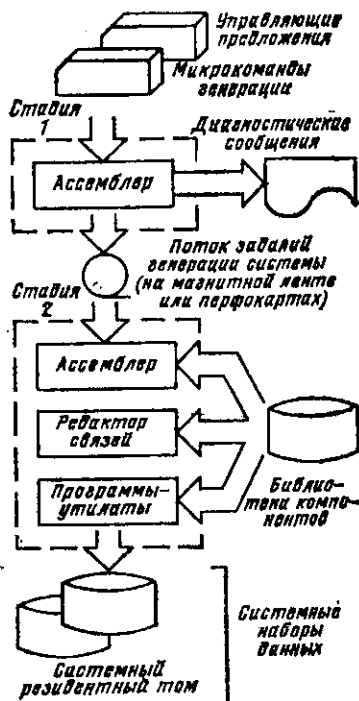


Рис. 16.3. Генерация операционной системы

На второй стадии генерации поток заданий обрабатывается ассемблером, редактором связей и утилитами (служебными программами). При этом происходит ассемблирование заданных пользователем модулей; редактор связей помещает эти модули в резидентную часть управляющей программы (ядро), а затем обрабатывает выбранные модули для создания разделов библиотек новой операционной системы; наконец, программы-утилиты завершают формирование (иницирование) библиотек новой ОС.

Пользователь имеет возможность осуществить генерацию нескольких типов: полную генерацию операционной системы, генерацию трансляторов-библиотек, генерацию программ управления устройств ввода-вывода.

При генерации имеется возможность задания режима управляющей программы: однопрограммного (PCP), мультипрограммирования с фиксационным числом задач (MFT), мультипрограммирования с переменным числом задач (MVT) и др.

Полная генерация системы необходима в том случае, когда изменения касаются возможностей управляющей программы, а также в некоторых случаях при изменении конфигурации установки.

Генерация ядра представляет собой процесс присоединения ядра к ОС. При этом новое ядро данной ОС должно соответствовать режиму, заданному при полной генерации системы, поскольку необходима совместимость резидентной части управляющей программы (ядра) и ее нерезидентной части.

При генерации трансляторов-библиотек происходит присоединение к операционной системе трансляторов и соответствующих им библиотек. При генерации программ управления устройств ввода-вывода пользователь имеет возможность присоединять, отключать или изменять устройства ввода-вывода, устройства управления и каналы. Возможно изменение группового имени устройства, изменение конфигурации консолей, введение возможности разделения устройств прямого доступа и т. д.

Адаптация ОС к конкретным условиям. После генерации в систему включают нестандартное программное обеспечение, к которому относятся дополнительные трансляторы, системы управления базами данных, сервисные программы и процедуры, — как разработанные в организации, проводящей эксплуатацию ОС, так и полученные от других организаций. Кроме того, часто проводится корректировка модулей ОС для увеличения удобства работы с ней (например,

исключение некоторых сообщений на консоль оператора, если они при данных условиях эксплуатации не несут информации).

В новой ОС каталогизируются наборы данных, используемые в ходе эксплуатации системы, системным параметрам присваиваются значения, соответствующие условиям работы на данной установке и вычисленные в ходе эксплуатации предыдущих версий ОС. Для облегчения адаптации в ходе эксплуатации необходимо поддерживать мобильность системы по отношению к смене версий ОС, т. е. по возможности разделить на носителях стандартные и нестандартные компоненты программного обеспечения, не использовать системозависимых программ, использовать каталоги на дополнительных томах магнитных дисков, а не на резидентном томе и т. д.

Для повышения производительности ЭВМ необходимо определение «узких мест» в ее работе и перераспределение ресурсов системы с целью их устранения. Для этого необходимо измерение параметров функционирования ОС и вычислительной установки в целом. Измерения проводятся с помощью специальных программных и программно-аппаратурных средств.

Проверка работоспособности ОС. Сгенерированная система подвергается тестированию. Для этого на ней выполняются пакеты прикладных программ, используемые при эксплуатации системы, а также прогоняются сервисные программы и процедуры, включенные после генерации. В ходе тестирования фиксируются все ненормальные результаты выполнения программ и производится их анализ.

Отказы прикладных программ могут быть вызваны ошибками в самих программах, не проявлявшимися на предыдущей версии (например, иногда некорректное использование буферов ввода-вывода не проявляется при работе в системах с реальной адресацией и приводит к аварийному завершению задачи при работе в системах с виртуальной памятью). В результате испытаний может быть выявлено также ухудшение характеристик системы по сравнению с предыдущей версией (например, снижение производительности, увеличение частоты сбоев и т. п.).

По результатам испытаний выносится решение о целесообразности перехода на новый вариант ОС, либо о необходимости доработки прикладных программ для их использования в новой версии ОС, либо об отказе от использования новой версии.

Нарушения в работе ОС. В ходе эксплуатации ОС необходимо тщательно фиксировать все случаи нарушения ее работы. Они проявляются в виде регистрируемых сбоев и отказов оборудования, а также в виде отказов ОС (аварийное завершение системных задач и зависания).

Отказы ОС могут быть вызваны либо дефектами системных программ, либо сбоями аппаратуры, не выявленными средствами контроля, либо ошибками или недосмотром оператора, неправильно отреагировавшего на запрос ОС.

Дефекты системных программ в большинстве случаев являются ситуационными, т. е. проявляются только в определенных ситуациях. Наиболее типичными примерами нарушений этого типа являются переполнение системных таблиц или очередей, а также клинчи, т. е. взаимные блокировки программ при обращениях к ресурсам. Иногда удается выйти из этих ситуаций с помощью удачных действий оператора, но во многих случаях приходится перезагружать ОС. Возникновение таких ситуаций возможно из-за ошибок генерации системы (например, выделение слишком малого объема ОП для области системных очередей) или из-за ошибок в программе ОС (например, параллельное использование программы двумя задачами может привести к сбою, если программа объявлена реентерабельной, а на самом деле ею не является).

Отказ ОС вследствие сбоя оборудования возникает в том случае, если после успешного запуска операции ввода-вывода аппаратура не сформировала прерывание по концу операций. Операционная система остается работоспособной, но устройство, потерявшее прерывание, становится недоступным, так как ОС считает, что оно занято операцией ввода-вывода.

Последние версии ОС содержат компонент МПС, контролирующий зависание описанного типа и имитирующий прерывание ввода-вывода, если оно не поступило в течение 15 с (для некоторых устройств — через большие промежутки времени).

Однако более сложные случаи нарушений не распознаются ОС и приводят к заикливанию супервизора или к программному прерыванию в супервизоре.

В табл. 16.2 приведены сведения о некоторых типах зависания ОС ЕС [72].

Примером некорректного действия оператора, приводящего к отказу ОС, является попытка перевести в оперативное состояние устройство ввода-вывода, отсутствующее в реальной конфигурации. Операционная система запрашивает

Таблица 16.2. Виды зависаний операционной системы

Тип зависания	Внешнее проявление	Возможная причина
Незавершенная операция процессора	1. Останов синхронизации при выполнении команды 2. Не завершилась команда ввода-вывода	Сбой в процессоре Сбой в процессоре или канале
Переход ЭВМ в состояние «ожидания», маски системы открыты	1. Процессор обрабатывает прерывания, отвечает на команды оператора, задания не выполняются 2. Прерывания не обрабатываются, каналы на запросы устройств выставляют БЛК-К	Нет сигнала от периферийного устройства о завершении ранее начатой операции ввода-вывода Устройство, на котором находится системный пакет, сняло готовность, выключено или не завершило ранее начатую операцию
Переход ЭВМ в состояние «ожидания», маски системы закрыты	Прерывания не обрабатываются, задания не выполняются	Код прерывания в ССП указывает вид ошибки
Защелкивание программы	1. Процессор выполняет цикл из нескольких команд, одна из которых — команда ввода-вывода 2. Процессор выполняет цикл из нескольких команд, среди которых нет команды ввода-вывода 3. Процессор выполняет цикл из большого числа команд, проводящих обработку программных прерываний	Устройство занято, а оно необходимо для продолжения работы Ошибка в программе операционной системы Цепочка программных прерываний, возможна ошибка в операционной системе

ет установку на это устройство тома и ожидает, пока произойдет прерывание, свидетельствующее о готовности устройства к работе.

Для фиксации отказов ОС применяются, как правило, дампы памяти. В их число входит системный дамп, автоматически записываемый при аварийных завершениях в системных программах, или по команде оператора, и незави-

симый дамп, выдаваемый автономной программой и используемый в тех случаях, когда система потеряла работоспособность. Оба вида дампа состоят из образа памяти и могут выводиться непосредственно на печатающее устройство, либо на магнитный носитель с последующей распечаткой специальной утилитой. Наиболее сложные случаи, когда система возобновляет работу, не формируя дампа (например, при программных прерываниях в супервизоре ввода-вывода). Для анализа подобных ситуаций приходится применять трассировщики, так как возникновение таких ситуаций часто невозможно предсказать и воссоздать, то накапливают «трассу» (информацию об особенностях прохождения программы). Иногда, однако, информации, содержащейся в трассе, бывает недостаточно для выяснения причины отказа ОС, и в этом случае приходится создавать специальные программы-ловушки для фиксации конкретного нарушения.

Устранение отказов ОС, как правило сводится к доработке аппаратуры, программ или, в худшем случае, к отказу от режимов работы, в которых нарушение проявляется наиболее часто.

Требования фиксации отказов ОС и быстрого возобновления работы после сбоя являются противоречивыми, поэтому, в зависимости от типа задач, решаемых на конкретной вычислительной установке (пакетные задания или режим реального времени), изменяются методы фиксации отказов ОС и объем собираемой информации.

Усовершенствование ОС. В процесс эксплуатации ОС входят также ее усовершенствование и доработка по мере обнаружения ошибок и неудобств в использовании. Простейшим вариантом доработки является составление процедур языка управления заданиями, включающими в себя наиболее часто используемые в данном ВЦ задания. Удачно составленный пакет процедур существенно облегчает работу оператора и программистов, повышает производительность ЭВМ, снижает вероятность разрушения информации от неумелого использования (например, при использовании стандартной утилиты ошибка всего в одном символе управляющего оператора приводит к уничтожению всей библиотеки вместо одного раздела) и т. п. Использование специальных процедур позволяет избежать таких ситуаций.

В более сложных случаях доработки состоят в написании сервисных программ расширяющих возможности стандартных утилит.

16.10. ОБСЛУЖИВАНИЕ НОСИТЕЛЕЙ ДАННЫХ НА МАГНИТНЫХ ЛЕНТАХ И ДИСКАХ

Носители данных на магнитных лентах должны храниться в специальных шкафах, приспособленных для вертикального хранения магнитных лент. Шкафы устанавливаются в отдельных помещениях, в которых поддерживается температура $20 \pm 5^\circ\text{C}$, влажность $65 \pm 5\%$ и высокий уровень чистоты, достигаемой ежедневной влажной уборкой.

При работе с магнитными лентами необходимо выполнять следующие требования:

- соблюдать осторожность во избежание смятия, загрязнения и образования царапин;

- не оставлять магнитные ленты с информацией на накопителях;

- регулярно проверять магнитную ленту с информацией на качество;

- для сохранности информации следует иметь три экземпляра магнитных лент: подлинник, контрольную копию и дубликат;

- при работе с постоянно корректируемыми массивами необходимо создание поколений массивов «дед — отец — сын». Каждая очередная корректура массивов создает новое поколение. При повреждении массива его можно восстановить по-предыдущему поколению и имеющейся корректуре;

- магнитные ленты с постоянными массивами информации должны выдаваться для работы без кольца защиты записи;

- каждая магнитная лента должна иметь этикетку с названием ленты, датой ее записи, названием программы, с помощью которой выполняется ее копирование, тип накопителя, на котором записана информация.

Носители данных на магнитных дисках (пакеты сменных магнитных дисков) должны храниться в отдельных помещениях на стеллажах или специальных шкафах. Каждый пакет магнитных дисков хранится в своем пластмассовом контейнере. Климатические условия аналогичны условиям хранения магнитных лент.

При работе с магнитными дисками необходимо выполнять следующие требования:

- периодически очищать верхнюю крышку контейнера от пыли изнутри;

- периодически очищать конусообразную ступицу пакета, устанавливаемую на шпиндель накопителя;

- периодически очищать мягкой безволокнистой материей, смоченной в спирте, поверхности пакета;

регулярно проверять фильтр пакета и заменять его при загрязнении и повреждении;

после снятия пакета с накопителя с помощью верхней крышки контейнера его необходимо сразу же закрыть нижней крышкой;

запрещается трогать пальцами лаковое покрытие дисков, так как появление на поверхности жировых пятен может привести к прилипанию пыли и порче поверхности пакета.

СПИСОК ЛИТЕРАТУРЫ

1. Автоматическая система контроля и диагностики электропитания ЭВМ ЕС-1045/А. А. Погосян, С. Е. Казарян, Л. А. Григорян и др. — Вопросы радиоэлектроники, Сер. ЭВТ, 1978, 10 вып., с. 96—99.
2. Андраш Г., Лайош Н. Измерительная техника в системе телеобработки данных. Приборы по передаче данных производства завода ОРИОН. — Сборник докладов ВНР. Выставка средств ЕС ЭВМ и СМ ЭВМ и их применение. — Н.: 1979, с. 23—27.
3. Берберян А. К., Лалаян Р. А. Система сериализации ЭВМ ЕС-1045. — Вопросы радиоэлектроники. Сер. ЭВТ, 1978, вып. 10, с. 100—104.
4. Вентцель Е. С. Теория вероятностей. — М.: Наука, 1969. — 576 с.
5. Вентцель Е. С. Исследование операций. — М.: Советское радио, 1972. — 551 с.
6. Волков А. П., Михайлов И. Б. К вопросу об автоматической генерации диагностических средств. — Вопросы радиоэлектроники, сер. ЭВТ, 1979, вып. 1, с. 39—45.
7. Волков О. М., Замулюкин А. Т. Противопожарная защита вычислительных центров. — М.: Стройиздат, 1982. — 65 с.
8. Георгиев Н. В., Орлов Б. В. Функциональный контроль полупроводниковых запоминающих устройств. — Электронная промышленность, 1980, № 6, с. 3—20.
9. Гнедов Г. М. Контроль аппаратуры передачи данных. — М.: Радио и связь, 1981. — 152 с.
10. Гордон Г., Надиг Х. Локализация неисправностей в микропроцессорных системах при помощи шестнадцатиричных ключевых кодов. — Электроника, 1977, № 5, с. 23—33.
11. Горский Л. К. Статистические алгоритмы исследования надежности. — М.: Наука, 1970. — 400 с.
12. Горский Л. К., Нестеренко Б. Н. К вопросу об оценке экономической эффективности комплексного централизованного обслуживания средств ЭВТ. — Вопросы радиоэлектроники. Сер. ЭВТ, 1978, вып. 14, с. 26—37.
13. Горский Л. К., Бацуро Г. Н., Кобранов М. Е. Проблемы обеспечения надежности средств ЭВТ. — Вопросы радиоэлектроники, Сер. ЭВТ, 1978, вып. 14, с. 43—53.
14. Громыко Г. Л. Статистика. — М.: Изд-во МГУ, 1981. — 407 с.
15. Гурвич И. С. Защита ЭВМ от внешних помех. — М.: Энергия, 1975. — 160 с.
16. Дроздов Е. А., Комарницкий В. А., Пятибратов А. П. Электронные вычислительные машины Единой системы. — М.: Машиностроение, 1981. — 648 с.
17. Дружинин Г. В. Надежность автоматизированных систем. — М.: Энергия, 1977. — 536 с.

18. Дружинин Г. В. Процессы технического обслуживания автоматизированных систем. — М.: Энергия, 1973.

19. Жорницкий В. Б., Поволоцкий Ф. Б. Измеритель прикладных программ (ИПП 1). — В кн.: АСУ технологических процессов. — М.: МДНТП, 1980, с. 41—45.

20. Залан А. Е., Флеров А. Б., Хорошавин В. В. О тестировании периферийных устройств параллельно с выполнением программ пользователей. — Тезисы докладов Первой Всесоюзной конференции пользователей ЕС ЭВМ. — М., 1975, с. 53—55.

21. Заренин Ю. Г. Корректирующие коды для передачи и переработки информации. — Киев: Техника, 1965. — 170 с.

22. Зиновьев В. Д., Клочков В. Д. Контроль и управление обработкой данных на вычислительном центре. — Вопросы радиоэлектроники. Сер. ЭВТ, 1979, вып. 5, с. 9—21.

23. Имитатор канала для наладки внешних устройств ЕС ЭВМ/ В. А. Исаенко, В. А. Калинин, В. М. Тафель, В. И. Цуранич. — Обмен опытом в радиопромышленности, 1980, № 9, с. 55—56.

24. Каган Б. М. Электронные вычислительные машины и системы. — М.: Энергия, 1979. — 528 с.

25. Каган Б. М., Мкртумян И. Б. Об одном подходе к проектированию систем автодиагностики. — Вопросы радиоэлектроники. Сер. ЭВТ, 1971, вып. 8, с. 42—51.

26. Каган Б. М., Мкртумян И. Б., Бузюк М. А. Об одной реализации D-алгоритма в системе автоматизации проектирования диагностических тестов для комбинационных автоматов. — Вопросы теории вычислительных систем, вып. 395, — М.: МИИТ, 1971, с. 195—210.

27. Каган Б. М., Котляровский В. Н. Многофазные вероятностные модели вычислительных систем. — Методические указания. — М.: МИИТ, 1977. — 24 с.

28. Каган Б. М., Карачаров А. Ф. Вероятностные модели информационно-вычислительных систем на основе динамики средних. — Автоматика и телемеханика, 1975, № 3, с. 153—161.

29. Каган Б. М., Сташин В. В. Микропроцессоры в цифровых системах. — М.: Энергия, 1979. — 196 с.

30. Казак А. С., Кондрашев А. Ф. Проблемы организации обработки данных у пользователей. — Вопросы радиоэлектроники. Сер. ЭВТ, вып. 5, 1979, с. 3—8.

31. Казак А. С., Ключев В. Е., Максимов Н. С. Некоторые аспекты эксплуатации операционных систем ОС ЕС ЭВМ. — Вопросы радиоэлектроники. Сер. ЭВТ, 1979, вып. 5, с. 31—36.

32. Казаков А. К. Методы и средства восстановления после машинных ошибок. — Вопросы радиоэлектроники. Сер. ЭВТ, вып. 5, 1980, с. 31—40.

33. Карнаух В. Г., Гольдкабер Е. М., Горячкина З. В. Оценка производительности и контроль за функционированием вычислительных систем (обзор зарубежной литературы), 1975, № 1, с. 71—79.

34. Ключев В. Е., Максимов Н. С. Режим пакетной обработки в ОС ЕС ЭВМ. — Вопросы радиоэлектроники. Сер. ЭВТ, 1979, вып. 5, с. 31—36.

35. Козлов Б. А., Ушаков И. А. Справочник по расчету надежности. — М.: Советское радио, 1975. — 432 с.

36. Комплекс программ моделирования цифровых устройств в процессе диагностирования/В. Т. Веселовский, Л. М. Кавальчук, Л. Н. Нечипоренко и др. — Вопросы радиоэлектроники. Сер. ЭВТ, 1979, вып. 12.

37. Кучукян К. А. Особенности системы аппаратного контроля ЭВМ ЕС-1045. — Вопросы радиоэлектроники. Сер. ЭВТ, 1978, вып. 10, с. 69—74.
38. Липаев В. В. Надежность программного обеспечения АСУ. — М.: Энергониздат, 1981. — 240 с.
39. Логическая организация управляющей вычислительной машины ВНИИЭМ-3/Б. М. Каган, В. М. Долкрат, М. М. Каневский, Г. Х. Новик. — Электричество, 1966, № 3, с. 1—8.
40. Майерс Г. Надежность программного обеспечения. — М.: Мир, 1980. — 260 с.
41. Мазур Е. И., Новик Г. Х., Сташин В. В. Схема построения простого сигнатурного анализатора. — В кн.: Цифровые вычислительные машины и системы и их применение на железнодорожном транспорте. — М.: МИИТ, 1979, вып. 642, с. 60—65.
42. Манукян В. Г., Аюнджания Р. В., Багдасарян А. Б. Особенности профилактики ЭВМ ЕС-1045. — Вопросы радиоэлектроники. Сер. ЭВТ, вып. 5, с. 25—32.
43. Мартин Дж. Системный анализ передачи данных, т. 1. — М.: Мир, 1975. — 256 с.
44. Мартин Дж. Организация баз данных в вычислительных системах. — М.: Мир, 1978. — 616 с.
45. Мкртумян И. Б., Маркарян М. С., Минасян А. Е. Средства восстановления ЭВМ ЕС-1045. — Вопросы радиоэлектроники. Сер. ЭВТ, 1978, вып. 10, с. 59—68.
46. Средства диагностики ЭВМ ЕС-1045/И. Б. Мкртумян, Э. В. Аюпян, М. О. Карахатян и др. — Вопросы радиоэлектроники. Сер. ЭВТ, 1978, вып. 10, с. 52—58.
47. Морозов Е. А. Об одной модели восстановления базы данных. — Вопросы радиоэлектроники. Сер. ЭВТ, 1978, вып. 8.
48. Морозов Е. А. Восстановление баз данных. — Вопросы радиоэлектроники. Сер. ЭВТ, 1978, вып. 8, с. 86—91.
49. Надежность автоматизированных систем управления/И. О. Атовмян, А. С. Вайрадян, Ю. П. Руднев и др. — М.: Высшая школа, 1979. — 287 с.
50. Муса Дж. Д. Измерение и обеспечение надежности программных средств. — ТИИЭР, т. 68, 1980, № 9, с. 113—128.
51. Новик Г. Х., Сташин В. В., Мазур Е. И. Элементы теории сигнатурного анализа. — В кн.: Цифровые вычислительные машины и системы и их применение на железнодорожном транспорте. — М.: МИИТ, 1979, вып. 642, с. 52—59.
52. Об одном примере построения аппаратно-программной диагностики периферийного устройства ЭВМ/И. Я. Доморадов, В. А. Ключев и др. — Вопросы радиоэлектроники. Сер. ЭВТ, 1979, вып. 12, с. 100—106.
53. Основы технической диагностики/В. В. Карибский, П. П. Пархоменко, Е. С. Согомоян, В. Ф. Халчев. — М.: Энергия, 1976. — 460 с.
54. Основы эксплуатации радиоэлектронной аппаратуры/А. К. Быкадоров, Л. К. Кульбак, В. Ю. Лавриненко и др.; Под ред. В. Ю. Лавриненко. — 2-е изд. — М.: Высшая школа, 1978. — 320 с.
55. Панфилов И. В., Половко А. М. Вычислительные системы. — М.: Советское радио, 1980. — 304 с.
56. Пархоменко П. П., Согомоян Е. С. Основы технической диагностики. — М.: Энергониздат, 1981. — 320 с.
57. Перспективы развития автоматизации разработки средств диагностики/Ю. С. Ломов, И. Б. Михайлов, Ю. А. Третьяков и др. — Вопросы радиоэлектроники. Сер. ЭВТ, 1980, вып. 3.

58. Питерсон У. Коды, исправляющие ошибки. — М.: Мир, 1976. — 594 с.
59. Пржилковский В. В., Ломов Ю. С. Технические и программные средства Единой системы ЭВМ (ЕС ЭВМ-2). — М.: Статистика, 1980. — 232 с.
60. Проектирование свойства эксплуатационной надежности для системы IBM 360. — Кибернетический сборник, № 1, — М.: Мир, 1964, с. 137—161.
61. Пятибратов А. П., Бацюрю Г. А. Вопросы построения системы комплексного централизованного обслуживания средств ЭВТ. — Вопросы радиоэлектроники. Сер. ЭВТ, 1978, вып. 14, с. 15—25.
62. Саати Т. Я. Элементы теории массового обслуживания и ее приложения. — М.: Советское радио, 1971. — 520 с.
63. Сантони А. Контрольно-измерительные приборы для испытаний логических схем. — Электроника, 1975, № 19, с. 37—45.
64. Староверова Г. Н., Трофимов Г. Е. Программное обеспечение средств восстановления ввода-вывода в ОС ЕС. — Вопросы радиоэлектроники. Сер. ЭВТ, 1980, вып. 5, с. 41—47.
65. Степанов С. В. Профилактические работы и сроки их проведения. — М.: Советское радио, 1972. — 136 с.
66. Тейер Т., Липов М., Нельсон Э. Надежность программного обеспечения. — М.: Мир, 1981. — 392 с.
67. Титов Ю. А. Состояние и перспективы развития комплексного централизованного обслуживания Единой системы электронно-вычислительных машин. — Вопросы радиоэлектроники. Сер. ЭВТ, 1978, вып. 14, с. 8—14.
68. Терзян О. А., Чахони Л. М. Способы и схемотехнические решения, обеспечивающие работоспособность полупроводниковых ПЗУ с неисправными запоминающими элементами. — Вопросы радиоэлектроники. Сер. ЭВТ, 1977, вып. 8, с. 56—67.
69. ГОСТ 20911—75. Техническая диагностика. Основные термины и определения.
70. Хауз Ч. Особенности проектирования и испытания информационных систем. — Электроника, 1975, № 9, с. 26—34.
71. Хоффман Л. Дж. Совершенные методы защиты информации. — М.: Советское радио, 1980. — 264 с.
72. Чернышев Г. М. Обеспечение ремонтпригодности в ЕС ЭВМ. — Вопросы радиоэлектроники. Сер. ЭВТ, 1979, вып. 5, с. 77—90.
73. Шлимович Е. М. Логические анализаторы для проверки и наладки сложных цифровых устройств и систем. — Вопросы радиоэлектроники. Сер. ЭВТ, 1980, вып. 4, с. 27—42.
74. Шлимович Е. М. Анализатор логических состояний и временных диаграмм. — Вопросы радиоэлектроники. Сер. ЭВТ, 1981, вып. 2, с. 69—83.
75. Шубинский Н. Б., Пивень Е. Н. Расчет надежности ЭВМ. — Киев: Техніка, 1979. — 232 с.
76. Шураков В. В. Надежность программного обеспечения системы обработки данных. — М.: Статистика, 1981. — 216 с.
77. Шрайбер Т. Дж. Моделирование на GPSS. — М.: Машиностроение, 1980. — 592 с.
78. Электронная вычислительная машина ЕС-1030/Под общей ред. А. М. Ларионова. — М.: Статистика, 1977. — 256 с.
79. Электронная вычислительная машина ЕС-1050/Под общей ред. А. М. Ларионова. — М.: Статистика, 1978. — 302 с.
80. Электронная вычислительная машина ЕС-1045/А. Т. Кучукая,

Т. Е. Саркисян, И. Б. Мкртумян и др. — М.: Финансы и статистика, 1981. — 183 с.

81. Akljama F. An Example of Software System Debugging. — Proc. IFIPS, 1971, Conf. p. 350.

82. Boehm B. W., Brown J. R., Kasper M. Software Quality Evaluation. — W.: Ney and Son, 1976, № 4.

83. Carter W. C., McCarthy C. E. Implementation of an experimental fault-tolerant memory system. — Digest of papers of 1978. International Symposium on Fault-Tolerant Computing, p. 113—118.

84. Curtice R. Integrity in Data Base Systems. — Datamation, 1977, № 5, p. 64—68.

85. Camuso J. A. Specifying and Selecting Uninterruptable Power Supply Systems. — Computer Design, 1974, № 7, p. 63—65.

86. Cheung R. C. A User-Oriented Software Reliable Model. — IEEE Transactions on Software Engineering, March 1980, vol. SE-6, № 2, p. 118—125.

87. Dick G. The Communication Channel. It's Broken. Now What? — Datamation, 1977, vol. 10, p. 113—126.

88. Endres A. An Analysis of Errors and Their Sources in System Programs. — IEEE Transactions on Software Engineering, vol. SE-1, № 2, p. 140—149.

89. Gay A. Reliability of partially self-checking circuits. — Digest of 7th International Symposium on Fault-Tolerant Computing, 1977, p. 135—141.

90. Gourdon E. Reliability of real-time systems. Part 1. Different concepts of reliability. — Modern Data, 1972, № 1, p. 36—42.

91. Holscher J. N. Static UPS Systems: Configurations and Inverter Specifications. — Computer Design, 1974, № 7, p. 65—71.

92. Inagaki M. Test and Diagnosis Program Generation Using Microinstruction. — NEC Research and Development № 26, July 1972, p. 35—52.

93. Jellinski Z., Moranda P. B. Software Reliability Research, Statistical Computer Performance Evaluation. — N.Y. Academic Press, 1972, p. 484.

94. Littlewood B., Verrall J. L. A Baejesian Reliability Growth Model for Computer Software, Record of the 1973. — IEEE Symposium on Computer Software Reliability, 1973, p. 70—77.

95. Manna Z., Vnillemin J. Fixpoint approach to the theory of computation. — Communications of the ACM, 1972, vol. 15, № 7, p. 528—536.

96. Nelson E. Estimating software reliability from test data microelectronics and reliability. IEEE Transactions on Computers, 1978, vol. 7, p. 67—74.

97. Ramamoorthy C. V., Yih-Wo Han. Reliability Analysis of Systems with Concurrent Error Detection. — IEEE Transactions on Computers, 1978, vol. C-24, № 9, p. 868—878.

98. Roth J. P. Diagnosis of automata failures: a calculus and methods. — IBM Journal of Research and Development, 1966, № 10.

99. Roth J. P., Bonriclus W. G., Sneider P. R. Programmed algorithms to compute tests to detect and distinguish between failures in logic circuits. — IEEE Transaction on Electronic Computers, 1967, EC-16.

100. Schneidewind N. F. An Approach to Software Reliability Prediction and Quality Control. — Proc. of the 1972 Fall Joint Computer Conference, 1972, p. 837—847.

101. Shoomann M. L. Operating Testing and Software Reliability Estimation During Program Development. — Record of the 1973 IEEE Symposium on Computer Software Reliability, 1973, p. 51—57.

102. Saur E. M. Syndrom—Testable Design of Combinational Circuits. — IEEE Transactions on Computers, 1980, C-29, № 6, p. 442—451.
103. Sechu S. On an improved diagnosis program. — IEEE Transaction on Electric Computers, 1965, EC-14.
104. Sechu S., Freeman D. N. The diagnosis of asynchronous sequential switching systems. — IRE Trans. on Reliability, 1966, R-15.
105. Sakanchi M., Hasnikl K., Inose H. Synthesis and Realization of Diagnosable Processor with Nesessary Hardware Redundance for Locating Faulty Packages. Annual Report of the Engineering.
106. Stewiorek D. P., McClusky E. J. An Iterative Cell Swith Design for Hybrid Redundance. — IEEE Transactions on Computers, 1973, vol., C-22, № 3, p. 290—297.
107. Triveoli A. K., Schooman M. L. A many state Markov Model for the Estimation and Prediction. — Proc. of the International Conference on Reliable Software, 1975, p. 208—215.
108. Williams W., Angell J. Enhancing Testability of Lage-Scale Integrated Circuits via Test Points and Additional Logic. — IEEE Transactions on Computers, 1973, vol. C-22, № 1.
109. Нар. 4 023 142 (США). Common diagnostic bus for computer systems to enable testing concurrently with normal system operation/Woessner R.

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

А

- Автоматизация профилактических испытаний 297
- — — с изменением напряжений источников питания 300
- Автоматическое восстановление вычислительного процесса после отказов 145
- — — — — сбоях 154
- — — — — в периферийных устройствах 153
- — — после сбоях в процессоре 147
- Автотестер 234
- Алгоритм диагностирования 191
- Анализатор временных диаграмм 334
- логических состояний 334
- Аппаратурно-микропрограммные средства накопления информации об ошибках 311

Б

- Безопасность данных 187
- Безотказность 12

В

- Вероятность обнаружения неисправности 195
- правильного диагностирования 195
- события 19
- Возмущения в системе электропитания 315
- Восстанавливаемость программно-го обеспечения 61
- Восстановление баз данных 176, 181
- Вторичный источник питания 302

- Гарантированное электропитание 317
- — большей длительности 322
- — длительность 317
- Генерация операционной системы 354

Д

- Дамп памяти 359
- Диагностирование каналов ввода-вывода 235
- ОЗУ 237
- периферийных устройств 273
- — — с помощью встроенных средств 281
- — — — — диагностических приказов 276
- — — — — имитатора канала 279
- — — — — процессора 274
- — — — — тестеров 278
- по методу раскрутки 192
- функциональное комбинационных схем 270
- Достоверность функционирования 13

Ж

- Журнал изменений базы данных 178

З

- Защита баз данных от ошибок 174
- информации от несанкционированного доступа 186

И

- Идентификация пользователей 188

Избыточность алгоритмическая 102

- аппаратурная 101
- временная 101
- информационная 101

Измерение и контроль напряжений ВИП 314

Измеритель тока 332

Имитатор канала 279

- линии связи 285

Импульсная помехозащищенность ЭВМ 316

К

Кодовое расстояние 104

Код с проверкой четности 105

- Хемминга 108, 120

- циклический 110

Комплекс неавтономных тестов устройств 276

- программ технического обслуживания 276

Комплексное централизованное обслуживание 80

Комплексный коэффициент эксплуатационной надежности (использования) 16

Контроль арифметических операций 113

- каналов 141

- комбинированных схем 117

- оперативной памяти 129

- процессора 137

Конфиденциальность данных 187

Коэффициент эффективности процессоров обслуживания 16

Криптографическое закрытие 189

Л

Логический анализатор 333

- импульсный генератор 331

- компаратор 330

- пробник 327

М

Мажоритарная система 166

- — *N*-ированная 167

- — троированная 166

Метод *D*-алгоритма 265

- диагностирования 194

- — двухэтажный 197

- — по результатам регистрации состояний 222

Метод диагностирования последовательным сканированием неисправностей 197

- — по эталонным состояниям 212

- — со схемами встроенного контроля 220

- — с проверкой сменных блоков 215

- защиты от помех 317

- с самопроверяемым дублированием 221

- чувствительных путей 265

Микродиагностика 208

- встроенная 208

- динамическая 232

- загружаемая 208

- статическая 229

Модель ЗИП 42

- имитационная 58

- надежности программ с дискретно-понижающей частотой проявления ошибок 67

- — — больших программных комплексов 72

- — — с дискретным увеличением времени наработки на отказ 68

- — — экспоненциальная 69

- потоков восстановлений 33

- — отказов 26

- — сбоя 32

- профилактических испытаний 39

Н

Надежность 12

- программного обеспечения 59

О

Обработчик канальных ошибок 306

- машинных ошибок 306

Обслуживаемость ЭВМ 9, 15

Отказ 12

- внезапный 12

- постепенный 12

Ошибка жесткая 143

- мягкая 143

- скрытая 143

- явная 143

Объект восстанавливаемый 9

- невозстанавливаемый 9

- необслуживаемый 9

П

- Период проявления ошибки 143
- Помехи в цепях сигналов 323
- Построение гистограмм 49
- Порог повторения команды 149
- Поток пуассоновский 28
- событий 28
- Проверка канала оперативной частоты 291
- статистических гипотез 51
- Проверкопригодность ЭВМ 15
- Программы накопления информации об ошибках 309
- Профилактическое испытание 15

Р

- Распределение биномиальное 24
- Вейбулла 26
- Гамма 25
- геометрическое 23
- дискретное пуассоновское 23
- — равномерное 24
- логарифмически-нормальное 26
- нормальное 26
- экспоненциальное 25
- Эрланга К-го порядка 25
- Регистратор отказов ПУ 308
- перезагрузок операционной системы 309
- помех 316
- потери прерываний 309
- программных ошибок 309
- реконфигурации 309
- сбоя ПУ 307
- Режим использования баз данных 174
- Резерв нагруженный 163
- ненагруженный 163
- облегченный 163
- Резервирование без восстановления 163
- гибридное 163
- динамическое 162
- с восстановлением 163
- статическое 162
- структурное 162
- Реконфигурация ОП 156
- периферийных устройств 161
- Ремонтопригодность ЭВМ 13

С

- Сигнатурный анализ 243
- анализатор 244

- Система автоматизации профилактических испытаний 93, 95
- автоматического контроля 91, 95, 99, 125
- восстановления вычислительного процесса 92, 95, 145
- диагностирования 190
- — характеристики 190
- защиты информации 188
- контроля и диагностики электропитания 314
- оперативного тестирования 299
- проектирования тестов 250
- регистрации и обработки данных об отказах и сбоях 93, 95
- Состояние неработоспособное 6
- работоспособное 6
- Сохранность ЭВМ 18
- Среднее время восстановления 13
- — — достоверности информации 14
- — — на сбой 14
- — — наработки на отказ 13
- Среднее квадратическое отклонение 23
- Средняя продолжительность однократного диагностирования 195
- — профилактических испытаний 15
- Схема контроля дешифратора 121, 138
- — самопроверяемая 119
- — — по четности 120, 138
- — полностью самопроверяемая 119

Т

- Тест автономный 296
- АП 290
- «все нули/единицы» 237
- галонирующий 241
- долбление 239
- запись/считывание вперед и назад 238
- крест 240
- параллельный 298
- передачи данных 238
- пинг-понг 240
- разрушение считыванием 240
- секция 297
- — строки 0 и 1 337
- чередующиеся столбцы 0 и 1 239
- шахматный 237
- ядра 297

Тестер ПУ 278
Трасса 277

У

Управление обновлением базы
данных 174
Установление подлинности 188
Устойчивость функционирования
программного обеспечения 62

Ф

Формы эксплуатационного обслужи-
вания 79
Функции системы контроля 39

Ш

Шлейф дискретного канала 287
— местный 287
— непрерывного канала 287
— удаленный 287

Э

Эксплуатационная техническая
документация 47
Эксплуатационное обслуживание
6
Эксплуатационные ресурсы 18
Эксплуатация объекта 6
Эффективность систем контроля
121

ОГЛАВЛЕНИЕ

Предисловие	3
Глава первая. Основные понятия эксплуатационного обслуживания ЭВМ	6
1.1. Особенности ЭВМ как объекта эксплуатационного обслуживания	6
1.2. Основные эксплуатационные характеристики ЭВМ	10
Глава вторая. Элементы теории эксплуатации ЭВМ	18
2.1. Некоторые сведения из теории вероятностей	18
2.2. Модели потоков отказов и сбоев	29
2.3. Модели потоков восстановлений	33
2.4. Модель профилактических испытаний. Влияние периодичности профилактических испытаний на коэффициент готовности ЭВМ с частичным контролем	37
2.5. Модели ЗИП	42
2.6. Построение моделей надежности по экспериментальным данным	47
2.7. Имитационные модели процессов эксплуатационного обслуживания ЭВМ	53
Глава третья. Надежность программного обеспечения	58
3.1. Основные понятия	58
3.2. Причины отказов программного обеспечения	62
3.3. Последствия и признаки появления ошибок в программе	64
3.4. Аналитические модели надежности программ	66
Глава четвертая. Системотехнические аспекты эксплуатационного обслуживания ЭВМ	76
4.1. Иерархия процессов обслуживания. Формы организации эксплуатационного обслуживания	76
4.2. Комплексное централизованное обслуживание ЭВМ	80
4.3. Система дистанционного обслуживания ЭВМ	82
4.4. Режимы работы вычислительных машин и систем и проблемы их надежности и обслуживаемости	85
4.5. Системный подход при разработке концепции и аппаратурно-программных средств обслуживания ЭВМ	89
4.6. Требования к эксплуатационной технической документации	97
Глава пятая. Автоматический контроль правильности функционирования ЭВМ в процессе нормальной эксплуатации	99
5.1. Функции и характеристики системы контроля	99
5.2. Контроль передачи информации	103
5.3. Циклические коды	110

5.4. Контроль арифметических операций	113
5.5. Контроль комбинационных схем	117
5.6. Самопроверяемые схемы контроля	119
5.7. Определение эффективности системы автоматического контроля ЭВМ	121
5.8. Система автоматического контроля ЭВМ	125
Глава шестая. Автоматическое восстановление вычислительного процесса после машинных ошибок	142
6.1. Типы машинных ошибок и их влияние на вычислительный процесс	142
6.2. Автоматическое восстановление вычислительного процесса после сбоев	145
6.3. Автоматическое восстановление вычислительного процесса после отказов	154
6.4. Структурное резервирование	162
Глава седьмая. Защита, сохранение и восстановление баз данных	173
7.1. Проблемы эксплуатации баз данных	173
7.2. Защита баз данных от ошибок	174
7.3. Восстановление баз данных при аварийных ситуациях	176
7.4. Методы защиты информации от несанкционированного доступа	187
Глава восьмая. Системы диагностирования ЭВМ	190
8.1. Характеристики систем диагностирования	190
8.2. Метод двухэтапного диагностирования	197
8.3. Метод последовательного сканирования	203
8.4. Метод микродиагностирования	207
8.5. Метод эталонных состояний	212
8.6. Метод диагностирования, ориентированный на проверку сменных блоков	215
8.7. Метод диагностирования с помощью схем встроенного контроля	220
8.8. Метод диагностирования с помощью самопроверяемого дублирования	221
8.9. Метод диагностирования по результатам регистрации состояния	222
8.10. Сериализация данных о состоянии ЭВМ	223
8.11. Система автоматического диагностирования ЭВМ ЕС-1045	227
8.12. Особенности диагностирования каналов ввода-вывода	235
8.13. Диагностирование оперативных запоминающих устройств	237
8.14. Особенности обслуживания и диагностирования микропроцессорных устройств и микро-ЭВМ. Сигнатурный анализ	242
Глава девятая. Проектирование диагностических тестов	249
9.1. Общие принципы организации систем проектирования диагностических тестов (СПДТ) для ЭВМ	249
9.2. Проектирование тестов для систем диагностирования, реализующих метод двухэтапного диагностирования	252
9.3. Проектирование тестов для систем диагностирования, реализующих метод эталонных состояний	254
9.4. Проектирование тестов для систем диагностирования, реализующих метод микродиагностирования	256

9.5. Проектирование тестов для систем диагностирования, реализующих метод последовательного сканирования	263
9.6. Проектирование тестов для проверки ТЭЗ	263
9.7. Построение диагностических тестов для комбинационных схем	264
9.8. Функциональное диагностирование комбинационных схем	270
Глава десятая. Диагностирование периферийных устройств	272
10.1. Способы диагностирования периферийных устройств	272
10.2. Диагностирование УПУ/ПУ с помощью процессора	274
10.3. Принцип проверки УПУ/ПУ с помощью диагностических приказов	276
10.4. Диагностирование УПУ/ПУ с помощью тестеров	278
10.5. Диагностирование УПУ/ПУ с помощью имитаторов канала	279
10.6. Диагностирование УПУ/ПУ с помощью встроенных в УПУ средств	281
Глава одиннадцатая. Диагностирование средств телеобработки данных	283
11.1. Аппаратурные средства телеобработки данных	283
11.2. Диагностирование мультимплексора передачи данных	284
11.3. Диагностирование канала передачи данных	287
11.4. Проверка канала тональной частоты	291
Глава двенадцатая. Принципы автоматизации профилактических испытаний	294
12.1. Типы профилактических испытаний	294
12.2. Программные средства профилактических испытаний	296
12.3. Автоматизация профилактических испытаний с изменением напряжений вторичных источников питания	300
Глава тринадцатая. Автоматическое накопление информации об ошибках, ее обработка и использование	305
13.1. Цель накопления информации об ошибках	305
13.2. Программные средства обработки ошибок	305
13.3. Программные средства накопления, редактирования и печати информации об ошибках	309
13.4. Аппаратурно-микропрограммные средства накопления информации об ошибках	311
13.5. Сбор, обработка, обобщение и накопление информации о надежности ЭВМ	312
Глава четырнадцатая. Методы повышения эксплуатационной надежности систем электропитания ЭВМ	314
14.1. Основные проблемы эксплуатации систем электропитания ЭВМ	314
14.2. Защита ЭВМ от возмущений в системе электропитания	315
14.3. Защита ЭВМ от длительных перерывов электропитания	317
14.4. Помехи в цепях сигналов и способы выполнения межсоединений интегральных микросхем	323
Глава пятнадцатая. Контрольно-измерительная аппаратура для эксплуатационного обслуживания ЭВМ	327
15.1. Типы контрольно-измерительных приборов	327
15.2. Одноконтактный логический пробник	327
15.3. Многоконтактный логический пробник	329

15.4. Логический компаратор	330
15.5. Логический импульсный генератор	331
15.6. Измерители тока	332
15.7. Осциллографы	332
15.8. Логические анализаторы	333
Глава шестнадцатая. Процессы эксплуатационного обслуживания ЭВМ	336
16.1. Оборудование помещений для ЭВМ	336
16.2. Техника безопасности при работе с ЭВМ	340
16.3. Обеспечение пожарной безопасности вычислительных центров	341
16.4. Процессы планово-профилактического обслуживания	344
16.5. Ведение журнала эксплуатации ЭВМ	345
16.6. Эксплуатационная документация	347
16.7. Процедуры автономного тестирования ЭВМ с помощью программы «тест-монитор» (ЕС ЭВМ)	349
16.8. Процедуры неавтономного тестирования периферийных устройств под управлением ОС ЕС	352
16.9. Особенности эксплуатации операционных систем	354
16.10. Обслуживание носителей данных на магнитных лентах и дисках	361
Список литературы	363
Алфавитный указатель	369

БОРИС МОИСЕЕВИЧ КАГАН
ИГОРЬ БОРИСОВИЧ МКРТУМЯН

ОСНОВЫ ЭКСПЛУАТАЦИИ ЭВМ

Редактор Ф. Б. Поволоцкий
Редактор издательства А. Н. Гусьякая
Технический редактор Н. П. Собакина
Корректор М. Г. Гулина

ИБ № 2300

Сдано в набор 16.06.83. Подписано в печать 22.09.83. Т-19210. Формат 84×106¹/₂ мм.
Бумага типографская № 2. Гарнитура литературная. Печать высокая. Усл. печ. л. 19,74. Усл. кр.-отт. 19,74. Уч.-изд. л. 22,11. Тираж 27 000 экз. Заказ 497.
Цена 1 р. 10 к.

Энергоатомиздат, 113114, Москва, М-114, Шлюзовая наб., 10

Владимирская типография «Союзполиграфпрома» при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли
600000, г. Владимир, Октябрьский проспект, д. 7