

# ОСНОВЫ ЭКСПЛУАТАЦИИ И РЕМОНТА ЭВМ

*Допущено Министерством высшего  
и среднего специального образования СССР  
в качестве учебного пособия для учащихся  
средних специальных учебных заведений*

МОСКВА ЭНЕРГОИЗДАТ, 1981

ББК 32.972  
О-75  
УДК 681.3.004 (075.3)

Рецензенты: кафедра «Электронные вычислительные машины» МИИТ; Нешумова К. А.

Авторы: И. А. Орлов, В. Ф. Корнюшко, В. В. Бурляев,  
В. Н. Авдеев

**О-75** Основы эксплуатации и ремонта ЭВМ: Учеб. пособие для техникумов/И. А. Орлов, В. Ф. Корнюшко, В. В. Бурляев, В. Н. Авдеев. — М.: Энергоиздат, 1981. — 280 с., ил.

В пер.: 65 к.

Рассмотрены вопросы контроля, диагностики, а также общие принципы организации профилактических и ремонтных работ в процессе технической эксплуатации ЭВМ. Приведены общая структура системы контроля работы ЭВМ, наиболее распространенные коды, обнаруживающие ошибки, описана организация аппаратурного, тестового и программно-аппаратурного контроля ЭВМ. Проводится технико-экономическая оценка мероприятий по техническому обслуживанию ЭВМ.

Расчитано на учащихся средних специальных учебных заведений. Может быть полезным широкому кругу инженеров и техников, интересующихся вопросами эксплуатации и ремонта ЭВМ.

О  $\frac{30502-030}{051(01)-81}$  149-81(Э). 2405000000

ББК 32.972  
6Ф7

© Энергоиздат, 1981

## ПРЕДИСЛОВИЕ

Настоящее учебное пособие ориентировано на курс «Основы эксплуатации и ремонта ЭВМ» для учащихся средних специальных учебных заведений по специальности «Электронные вычислительные машины, приборы и устройства».

Широкое внедрение вычислительной техники в различные отрасли народного хозяйства требует организации правильной и эффективной эксплуатации ЭВМ.

Под эксплуатацией такой сложной динамической системы, какой является современная ЭВМ, следует понимать процесс использования ЭВМ по назначению при поддержании ЭВМ в технически исправном состоянии. Этот непрерывный процесс состоит из различных мероприятий: плановое техническое обслуживание, восстановление работоспособности после отказа, профилактические регламентные работы и т. д. Осуществление этих мероприятий требует значительных затрат, причем затраты в значительной степени зависят от того, насколько рационально организованы указанные мероприятия.

По мере расширения области применения цифровых вычислительных машин все большее значение приобретает задача обеспечения надежности их работы. В то же время усложнение ЭВМ, увеличение числа режимов их работы, применение новой элементной базы — интегральных схем — затруднили контроль правильности работы и поиск неисправностей машины.

Существует большое количество методов технической диагностики ЭВМ, использующих как дополнительную контрольную аппаратуру, так и чисто программные средства автоматической диагностики неисправностей машины. История развития методов диагностики ЭВМ показывает, что машины первого и второго поколений обслуживались весьма опытными техниками, которые применяли для выявления неисправностей в основном интуитивные процеду-

ры поиска, основанные на их опыте работы с конкретной машиной. При работе с современными машинами накопить такой опыт и передать его довольно трудно. Кроме того, трудность аппаратурной диагностики усугубляется тем, что в интегральных схемах нельзя наблюдать сигналы непосредственно на внутренних точках схемы, а введение многочисленных контрольных точек явно нерационально.

В связи с этим в настоящее время интенсивно развиваются и приобретают все большее значение программные и аппаратурно-программные методы контроля работоспособности машины.

Учитывая сложность поставленного вопроса и общую направленность курса, авторы не ставили перед собой цели написать справочник с детальным изложением всех методов и способов автоматического контроля, технического обслуживания и ремонта ЭВМ. Цель книги — рассмотреть основные методы контроля и эксплуатации ЭВМ в общем и по возможности достаточно полно в виде на примере современных машин третьего поколения, входящих в состав ЕС ЭВМ. При изложении конкретных примеров схем контроля отдельных узлов и устройств ЭВМ, тестов для программного контроля, рекомендаций по эксплуатации и ремонту ЭВМ авторы стремились рассматривать общие для ЭВМ единой серии решения (например, система тестового контроля ТЕСТ-монитор является общей для большинства таких машин). В тех же случаях, когда это оказывается невозможным, примеры приводились для одной из наиболее развитых и распространенных ЕС ЭВМ средней мощности — ЕС-1022. Так как учебное пособие рассчитано на учащихся, которым в дальнейшем придется непосредственно участвовать в решении задач и эксплуатации ЭВМ, руководствуясь эксплуатационной документацией, то конкретные тестовые примеры, технические схемные решения и рекомендации по эксплуатации изложены на основе инструктивных материалов по технической эксплуатации ЕС ЭВМ. Однако это ни в коей мере не снижает общности выводов и рекомендаций, приводимых в книге, так как методы технической диагностики и ремонта различных универсальных ЭВМ третьего поколения весьма схожи.

Работа над книгой распределялась между авторами следующим образом: гл. 1—5 написаны И. А. Орловым, § 5.6 — В. Н. Авдеевым, гл. 6 и 7 — В. Ф. Корнюшко, гл. 8—11 — В. В. Бурляевым.



Авторы выражают искреннюю благодарность рецензентам — коллективу кафедры ЭВМ и математического обеспечения Московского института инженеров железнодорожного транспорта, а также доценту этой кафедры, канд. техн. наук В. Г. Першееву, преподавателю Московского радиотехнического техникума им. А. А. Расплетина К. А. Нешумовой и научному редактору книги, канд. техн. наук Б. Н. Севрюкову, которые своими замечаниями и полезными советами способствовали улучшению рукописи.

*Авторы*

# ГЛАВА ПЕРВАЯ

## ЭКСПЛУАТАЦИЯ ЭВМ

### 1-1. ОБЩИЕ ПОНЯТИЯ

Эксплуатация ЭВМ заключается в использовании машины по назначению, когда машина должна выполнять весь комплекс возложенных на нее задач. Для эффективного использования и поддержания ЭВМ в работоспособном состоянии в процессе эксплуатации проводится техническое обслуживание.

Техническое обслуживание — это комплекс организационных мероприятий, в том числе обеспечение ЭВМ необходимой аппаратурой и оборудованием, предназначенный для эффективной эксплуатации и ремонта ЭВМ. Существуют три вида обслуживания (сервиса): индивидуальное, групповое и централизованное.

При индивидуальном сервисе обеспечивается обслуживание одной ЭВМ силами и средствами персонала, обслуживающего данную машину. В состав комплекта оборудования для этого вида сервиса входят:

- аппаратура контроля элементной базы ЭВМ и электропитания;

- контрольно-наладочная аппаратура для автономной проверки и ремонта технических средств ЭВМ;

- комплект электрорадиоизмерительной аппаратуры, необходимый для эксплуатации ЭВМ и работы стендов проверки отдельных узлов и типовых элементов замены (ТЭЗ) машины;

- комплект программ (тестов) для проверки работы ЭВМ; инструмент и ремонтные принадлежности;

- вспомогательное оборудование и приспособления;

- специальная мебель для хранения имущества вычислительного центра и рабочие места оператора ЭВМ и наладчика элементной базы.

При таком оборудовании предусматривается возможность оперативного поиска и устранения неисправности с помощью стендовой и контрольно-измерительной аппаратуры. Данный комплект в сочетании с необходимым ЗИП (запасным инструментом, приборами, ТЭЗ, запасными частями к механическим устройствам и т. п.) должен обеспечивать заданное время восстановления ЭВМ.

Групповой сервис служит для обслуживания нескольких ЭВМ, сосредоточенных в одном вычислительном центре (ВЦ), силами и средствами персонала вычислительного центра. Структура состава оборудования группового сервиса та же, что и при индивидуальном, но при этом предполагается наличие большего числа аппаратуры, приспособлений и т. д., исключающее, однако, неоправданное дублирование. Комплект группового сервиса включает как минимум комплект оборудования индивидуального сервиса ЭВМ, дополненный аппаратурой и приспособлениями других ЭВМ.

При наличии необходимого ЗИП, развитой сервисной аппаратуры и квалифицированного технического персонала индивидуальный сервис позволяет существенно сократить время восстановления машины, но при этом требуются значительные расходы на содержание технического персонала и сервисной аппаратуры.

Централизованное техническое обслуживание ЭВМ является более прогрессивной формой обслуживания вычислительных машин. Система централизованного технического обслуживания представляет собой сеть региональных центров обслуживания и их филиалов — пунктов технического обслуживания. Эти организации в централизованном порядке осуществляют:

монтажно-наладочные работы и ввод в эксплуатацию ЭВМ;

устранение сложных отказов, возникающих в процессе эксплуатации;

централизованный ремонт ТЭЗ и электромеханических устройств в ремонтных органах;

снабжение вычислительных центров ЗИП;

оказание помощи обслуживающему персоналу по вопросам математического обеспечения и совершенствования эксплуатации;

управление процессом обслуживания ЭВМ на основе данных системы учета и анализа;

подготовку эксплуатационного персонала средств вычислительной техники;

ввод в эксплуатацию существующих и разрабатываемых операционных систем ОС/ЕС и ДОС/ЕС, пакетов прикладных программ и т. п.

При централизованном обслуживании сокращаются расходы на содержание технического персонала ЭВМ, сервисной аппаратуры и ЗИП. Однако время восстановления ЭВМ в этом случае зависит от оперативности пунктов централизованного обслуживания и может составлять от одного до нескольких часов в зависимости от различных факторов (расстояния от ВЦ, условий работы бригады и др.).

Все мероприятия, входящие в техническое обслуживание, можно разделить на три группы: контроль технического состояния; профилактическое обслуживание; текущее техническое обслуживание.

Контроль технического состояния ЭВМ служит для контроля работы машины, локализации места неисправности, исключения влияния случайных сбоев на результаты вычислений. В современных ЭВМ подобный контроль осуществляется главным образом с помощью самой вычислительной машины.

Профилактическое обслуживание представляет собой ряд мероприятий, направленных на поддержание заданного технического состояния машины в течение определенного промежутка времени и продление ее технического ресурса. Профилактические мероприятия, проводимые на ЭВМ, можно в свою очередь разделить на две группы.

К первой группе относятся: внешний осмотр, чистка, смазка и устранение дефектов, обнаруженных при осмотре. Эти работы проводятся на выключенной машине.

Ко второй группе относятся контрольно-настроечные работы, проводимые на включенной машине.

С точки зрения организации профилактического обслуживания наибольшее распространение получило планово-предупредительное обслуживание, основанное на календарном принципе. При этом составляется график проведения регламентных работ, в котором указываются объемы и сроки профилактических мероприятий.

Под текущим техническим обслуживанием ЭВМ понимают комплекс настроечных и ремонтных работ, направленных на восстановление утраченных машиной свойств или работоспособности путем замены или восстановления ее деталей, узлов и блоков.

Эффективность эксплуатации ЭВМ в значительной степени зависит от уровня ее организации. Организация

эксплуатации представляет собой комплекс мероприятий, направленных на подготовку обслуживающего персонала, планирование работ, своевременное и полное обеспечение требуемым ЗИП, расходными материалами, правильное и систематическое ведение документации и т. п.

Рациональная организация системы эксплуатации ЭВМ и постоянное совершенствование этой системы является одним из главных средств обеспечения эффективного использования вычислительных машин.

## 1-2. ОСНОВНЫЕ ЭКСПЛУАТАЦИОННЫЕ ХАРАКТЕРИСТИКИ

Степень пригодности вычислительной машины к использованию по назначению и возможности ее технического обслуживания определяются эксплуатационными характеристиками ЭВМ.

К эксплуатационным характеристикам можно отнести: работоспособность, безотказность, сохранность, ремонтпригодность, долговечность и др.

Под *работоспособностью* ЭВМ понимается способность машины функционировать, обеспечивая выполнение заданных функций с параметрами, установленными требованиями технической документации. Эта характеристика позволяет судить о состоянии машины в определенный момент времени. Однако при эксплуатации ЭВМ важно знать ее состояние не только в данный момент, но и способность выполнять возложенные на машину задачи в течение заданного промежутка времени. Для этих целей вводится понятие *безотказность*.

Под *безотказностью* ЭВМ понимают ее способность сохранять работоспособность в течение заданного интервала времени при определенных условиях эксплуатации.

На этапе хранения ЭВМ пользуются такой характеристикой, как *сохранность*, под которой понимают способность машины сохранять исправное состояние при заданных условиях хранения.

В процессе эксплуатации ЭВМ обслуживающий персонал систематически сталкивается с такими вопросами, как удобство доступа к блокам и монтажу, приспособленность машины к устранению неисправностей и т. п. Для характеристики машины с точки зрения ее приспособленности к ремонту вводится понятие *ремонтпригодность*. Требования к ремонтпригодности машины предъявляются в зависимости от условий ее эксплуатации. Так, бортовые

машины в силу своей специфики использования не рассчитаны на обычное техническое обслуживание, поэтому такие машины относятся к неремонтопригодным. Отсюда и другой подход к структуре компоновки и монтажа подобных машин, поскольку вопросы ремонта здесь полностью возлагаются на саму машину.

Под *долговечностью* понимают свойство ЭВМ сохранять работоспособность до предельного состояния с необходимыми перерывами для технического обслуживания и ремонтов.

Важной характеристикой ЭВМ является *надежность* ее работы. Надежность машины — это свойство устойчиво функционировать при заданных условиях обслуживания и эксплуатации ЭВМ. Для оценки надежности служат ее количественные характеристики, рассчитываемые на базе статистической информации об обслуживании и эксплуатации ЭВМ. Количественные характеристики надежности позволяют изучить закономерности возникновения неисправностей, разработать эффективные меры их предупреждения, что в свою очередь дает возможность активно влиять на качество услуг, предоставляемых системой обслуживания ЭВМ. На практике используется большое количество показателей надежности, характеризующих свойства ЭВМ. Рассмотрим основные из них.

**Коэффициент использования**  $K_{и}$  — отношение времени, в течение которого ЭВМ находится во включенном состоянии  $t_{вкл}$ , к календарному времени за выбранный интервал функционирования (например, квартал)  $t_{кв}$ :

$$K_{и} = \frac{t_{вкл}}{t_{кв}}.$$

Коэффициент использования показывает степень загруженности ЭВМ, т. е. только организационную сторону использования ЭВМ на ВЦ.

**Коэффициент технического использования**  $K_{т.и}$  — отношение времени полезной работы ЭВМ за определенный период  $t_{п.р}$  ко времени включенного состояния машины  $t_{вкл}$ :

$$K_{т.и} = \frac{t_{вкл} - (t_о + t_y + t_{сб} + t_{пот} + t_{проф})}{t_{вкл}} = \frac{t_{п.р}}{t_{вкл}},$$

где  $t_о$ ,  $t_y$  — время обнаружения и устранения неисправностей;  $t_{сб}$  — время, потерянное на сбои (кратковременное нарушение работы ЭВМ) и устранение их последствий;  $t_{пот}$  — время потерь исправной ЭВМ по организационным причинам (ошибки оператора, неправильная программа, некачественные носители информации и т. п.);  $t_{проф}$  — время, затраченное на профилактические работы.

Коэффициент технического использования отражает, с одной стороны, деятельность пользователей ЭВМ, а с другой — качество технического обслуживания машины.

Коэффициент готовности  $K_r$  дает оценку готовности ЭВМ, работающей непрерывно, обеспечивать ее работоспособность в любые промежутки времени между выполнениями планового технического обслуживания. Он также характеризует долю времени правильного функционирования ЭВМ и не включает время, израсходованное на проведение профилактических мероприятий. Коэффициент готовности определяется по формуле

$$K_r = \frac{T_o}{T_o + T_b},$$

где  $T_o$  — время безотказной работы ЭВМ за рассматриваемый период;  $T_b$  — суммарное время восстановления работоспособности ЭВМ за рассматриваемый период.

Коэффициент эффективности профилактики  $K_{\text{проф}}$  определяется формулой

$$K_{\text{проф}} = \frac{n_{\text{проф}}}{n_o + n_{\text{проф}}},$$

где  $n_{\text{проф}}$  — количество неисправностей (отказов), выявленных при профилактике;  $n_o$  — количество отказов за рассматриваемый период, происшедших за полезное время работы ЭВМ.

Данный показатель определяет вероятность выявления отказа при проведении профилактик и характеризует существующую систему профилактического обслуживания ЭВМ в процессе эксплуатации.

Среднее время безотказной работы ЭВМ  $T_o$  — среднее значение наработки в часах между двумя отказами — вычисляется по формуле

$$T_{o, \text{cp}} = \frac{t_{\text{вкл}} - (t_o + t_y + t_{\text{сб}} + t_{\text{птг}})}{n_o}.$$

Данный показатель характеризует общую надежность работы ЭВМ и отдельных устройств при длительной непрерывной эксплуатации и реальную эксплуатационную надежность ЭВМ, достигнутую на данном ВЦ.

Среднее время восстановления ЭВМ  $T_b$  — среднее время вынужденного и нерегламентированного простоя, вызванного обнаружением и устранением отказа:

$$T_{b, \text{cp}} = \frac{\sum_{i=1}^{n_o} t_{vi}}{n_o},$$

где  $t_{vi}$  — время устранения  $i$ -го отказа.

Показатель  $T_b$  характеризует:  
 степень ремонтпригодности ЭВМ;  
 уровень культуры и организации работ в системе технического обслуживания;

оснащенность ВЦ технологической оснасткой для хранения тестового хозяйства, технической документации, ЗИП;

соответствие производственных условий и норм на ВЦ требованиям ТУ (благоприятные условия для производства работ: освещенность, уровень шумов, температура, планировка размещения ЭВМ в машинном зале, достаточное количество контрольно-измерительной аппаратуры);

квалификацию специалистов, обслуживающих машину.

Остановимся еще на одном важном понятии, характеризующем эксплуатационные свойства ЭВМ, — на *производительности* машин.

На протяжении всего развития ЭВМ оценка их производительности была и остается одной из сложных проблем. Для ее решения предлагались различные критерии и методы.

С развитием и совершенствованием больших вычислительных машин различных поколений сравнивать их по одному определенному критерию стало трудно. Если ЭВМ первого и второго поколений сравнивались по быстродействию (количеству команд, выполняемых в секунду), то для ЭВМ третьего поколения стали вводить такие понятия, как общая производительность машины, вычислительная мощность и др.

В зависимости от области применения менее быстродействующая, но с лучшим набором команд машина часто имела большую производительность, чем более быстродействующая.

Для первых моделей машин приемлемой оценкой считалось количество операций сложения в секунду. Например, если время сложения одной операции составляло 1 мкс, то считалось, что машина может производить 1 млн. операций в секунду.

Затем в качестве показателя производительности было избрано среднее быстродействие  $V_{\text{ср}}$ , характеризуемое средним количеством операций в единицу времени  $v_i$  и выражаемое через скорость выполнения каждой операции  $q_i$ :

$$V_{\text{ср}} = \sum_{i=1}^k v_i q_i,$$

где  $k$  — общее количество операций, выполняемых данной ЭВМ.

Для машин второго и третьего поколений производительность оценивалась по методикам Найта и Гибсона.

Вычислительная мощность  $P$ , определяемая по методике Найта, выражается через скорость, с которой ЭВМ обрабатывает информацию, при этом главными факторами являются: 1) скорость работы цен-



трального процессора  $t_c$ ; 2) время простоя центрального процессора  $t_n$ ; 3) емкость памяти, выражаемая через коэффициент  $M$ :

$$P = \frac{10^{12}}{t_c + t_n} M.$$

Благодаря особенностям оборудования две ЭВМ могут решать одну и ту же задачу с помощью различных машинных операций. Следовательно, значение  $P$  должно определяться при решении одной и той же или эквивалентной задачи.

Следует, однако, заметить, что большинство современных универсальных машин комплектуются различными по объему устройствами памяти (более восьми вариантов), к которым может быть придано несколько разных систем ввода-вывода, арифметическо-логических устройств и устройств управления (всего получается более 300 различных комбинаций вычислительных систем). Поэтому даже для небольшого числа вариантов вычисление  $P$  затруднительно, и чаще всего рассматривается только один наиболее типичный вариант структуры каждой ЭВМ.

Для оценки производительности современных ЭВМ чаще всего прибегают к методике Гибсона. Суть этой методики состоит в том, что на основе частоты появления тех или иных команд в процессе выполнения программы устанавливаются веса операций  $\rho_i$  и далее производятся вычисления производительности машины по формуле

$$P = \frac{\sum_{i=1}^n \rho_i}{\sum_{i=1}^n \rho_i t_i},$$

где  $t_i$  — время выполнения  $i$ -й операции. Чтобы вычислить производительность по формуле Гибсона, необходимо привести систему команд к одноадресной структуре.

Набор весов  $\rho_i$  зависит от характера решаемых задач. Существует несколько систем весовых коэффициентов, отражающих статистику круга задач, решаемых на машине. Эти системы называются смесями Гибсона. Например, для оценки быстродействия машин при решении задач научного характера принята такая система весов:

Сложение, вычитание и другие короткие операции 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

Используя эти веса и зная время выполнения операций, для любой машины можно быстро вычислять характеристику ее производительности и сравнивать таким способом ЭВМ с различной структурой. По этой методике рассчитывалась, в частности, производительность машин IBM 360 и IBM 370 и отечественных машин серии ЕС.

Так как результаты оценки производительности ЭВМ по разным методикам значительно отличаются, то для выявления закона изменения параметра пользуются одной методикой оценки.

### 1-3. ПРИНЦИПЫ ОРГАНИЗАЦИИ ЭКСПЛУАТАЦИИ

Эффективность использования ЭВМ во многом зависит от того, насколько рационально организована эксплуатация машин. В целом организация эксплуатации включает в себя:

- выбор системы обслуживания;
- материальное обеспечение обслуживания ЭВМ;
- определение необходимого количества обслуживающего персонала и его квалификации;
- планово-профилактические работы;
- эксплуатационную документацию;
- планирование эксплуатации ЭВМ;
- анализ и учет результатов эксплуатации;
- организацию и систематическое обучение обслуживающего персонала.

**Выбор системы обслуживания.** Общая система эксплуатации машин и количество персонала, обслуживающего и обеспечивающего работу ЭВМ, зависит от количества и класса машин, характера решаемых задач, от режима работы (односменная или круглосуточная), от типа предприятия, в котором эксплуатируются машины.

Наиболее типичными предприятиями, занимающимися эксплуатацией машин, являются ВЦ. По характеру решаемых задач и составу оборудования вычислительные центры укрупненно подразделяются на ВЦ для решения научных и инженерно-технических задач, плановых расчетов и экономических исследований и центры автоматизированного управления объектами или технологическими процессами.

Обычно в состав ВЦ входят подразделения технического обслуживания ЭВМ, математической подготовки задач, программирования и операторов. Вычислительные центры для плановых расчетов и экономических исследований могут, помимо того, иметь подразделения по типам решаемых задач (планирования, материально-технического учета, подготовки и унификации первичных документов, нормативного хозяйства и т. п.).

Центры автоматизированного управления объектами или технологическими процессами не имеют больших подразде-

лений программистов. Основным подразделением таких ВЦ является подразделение технического обслуживания ЭВМ и аппаратуры приема и передачи данных между объектом, диспетчером и ЭВМ.

Независимо от назначения и области применения современная вычислительная машина представляет собой систему, состоящую из технической (аппаратурной) и программной части. Для рациональной организации эксплуатации такой системы необходимо учитывать три стороны ее обслуживания: *оперативную, техническую и математическую*.

*Оперативное* обслуживание заключается в подготовке и вводе в ЭВМ исходной информации, контроле за ходом вычислительного процесса и вмешательстве в него по мере необходимости.

*Техническое* обслуживание служит для поддержания ЭВМ в работоспособном состоянии посредством проведения профилактических работ и текущего ремонта машины.

*Математическое* обслуживание обеспечивает подготовку задач к реализации их на ЭВМ.

В процессе работы ЭВМ ее обслуживание ведется специальными дежурными группами — сменами. В обязанности дежурных групп входит контроль за общим техническим состоянием машины и при необходимости выполнение текущего ремонта ЭВМ. Если дежурная группа не может справиться с ремонтом своими силами, то вызываются специалисты из централизованного органа при условии, что ЭВМ находится на централизованном обслуживании. Отсюда легко понять, что при индивидуальном обслуживании машины эффективность ее эксплуатации прежде всего зависит от квалификации обслуживающего персонала. Вычислительная машина представляет собой весьма сложную систему и поэтому требует от обслуживающего персонала больших знаний и большого практического опыта, что особенно сказывается на времени устранения неисправности в процессе текущего ремонта ЭВМ и, следовательно, на эффективности использования машинного времени.

**Материальное обеспечение обслуживания ЭВМ.** Наряду с этим качество эксплуатации ЭВМ зависит от обеспечения ее запасными элементами, контрольно-измерительными приборами, инструментом, различными приспособлениями и расходными материалами. Большое значение также имеет создание необходимых условий для нормального функционирования вычислительных средств (температурно-влажностный режим, режим электропитания и т. п.) и для обслу-

живающего персонала (климатические условия, уровень шумов, освещенность и т. п.).

Эксплуатация ЭВМ должна тщательно планироваться. Планирование должно охватывать весь круг вопросов, относящихся как к составлению общей программы работы ВЦ, распределению машинного времени и т. п., так и ко всей работе обслуживающего персонала.

Рациональная организация эксплуатации должна предусматривать накопление статистического материала по результатам эксплуатации машины с целью его обобщения, анализа и выработки рекомендаций по совершенствованию структуры обслуживания, повышению эффективности использования ЭВМ, снижению эксплуатационных расходов.

**Состав обслуживающего персонала.** Важным вопросом организации обслуживания ЭВМ является вопрос определения рациональной численности, квалификации и схемы расстановки специалистов для его проведения.

При круглосуточной работе ЭВМ обслуживание ведется сменным персоналом, организованным в дежурные группы (смены). Обычно имеют 3—4 смены, с помощью которых организуется 8-часовая или 12-часовая сменная работа обслуживающего персонала. Эксплуатация машины может быть также односменной или двухсменной.

Состав обслуживающего персонала зависит от вида технического обслуживания и режима эксплуатации ЭВМ.

Например, при индивидуальном обслуживании машины ЕС-1030 в течение одной смены требуется следующий инженерно-технический персонал: начальник смены с правом эксплуатации процессора, основной памяти, каналов, отвечающий за эти устройства — 1; специалист по накопителям на магнитных лентах и дисках (инженер-электрик) — 1; специалист по устройствам ввода-вывода и подготовки данных (инженер-электрик) — 1; специалист по обслуживанию устройств ввода-вывода, подготовки данных и накопителей (инженер-механик) — 1; оператор по операционной системе — 1; специалист по ремонту и проверке ТЭЗов и системы питания (техник) — 1.

При круглосуточной эксплуатации рекомендуется иметь одну усиленную смену, состоящую из специалистов по обслуживанию всех устройств, и остальные смены, состоящие из трех-четырех специалистов.

За качественное функционирование ЭВМ отвечает начальник машины, в обязанности которого входит участие в профилактических работах, а также руководство работами по внесению изменений и улучшений в схемы и конструкцию машины. Характер изменений может сообщаться заводом-изготовителем, другими организациями, эксплуатирующими

аналогичные машины, или выясняться на месте эксплуатации.

Начальник машины занимается также ведением документации, необходимой для статистического анализа работы ЭВМ, ее элементов и узлов, для определения надежности, необходимого режима профилактических работ, анализа условий решения задач на данной ЭВМ. Кроме того, начальник машины анализирует причины различных недоразумений, возникающих при работе машины, и совместно с математиками принимает участие в рассмотрении программы, вызвавшей недоразумение, для нахождения неисправности ЭВМ или ошибки в программе.

**Планово-профилактические работы.** Эти работы проводятся с целью поддержания устройств ЭВМ в исправном состоянии, выявления отказов в оборудовании, предупреждения сбоев и отказов при работе ЭВМ.

Например, для машины ЕС-1030 рекомендуется следующий регламент и периодичность планово-профилактических работ: ежедневная — 1 ч, двухнедельная — 4 ч, ежемесячная — 8 ч, полугодовая — 72 ч.

При проведении профилактических работ необходимо руководствоваться инструкциями по эксплуатации отдельных устройств, входящих в состав ЭВМ. При выполнении ежедневной и двухнедельной профилактики проверяются технические средства машины с помощью специальных тестов, входящих в состав программного обеспечения ЭВМ, а также работы, предусмотренные для внешних устройств. При проведении ежемесячной и полугодовой профилактики рекомендуется дополнительно проверять работоспособность технических средств машины совместно с операционной системой (см. гл. 9).

**Эксплуатационная документация (ЭД).** Состав ЭД зависит от класса ЭВМ, ее назначения, состава оборудования и т. п. В комплект ЭД на машину из основных документов включаются техническое описание, инструкция по эксплуатации и формуляр.

*Формуляр ЭВМ* является эксплуатационным документом, отражающим технические показатели машины и ее устройств. В формуляре дается комплектность аппаратуры и эксплуатационной документации, технические характеристики машины, данные контрольно-приемочных испытаний и гарантийные обязательства завода-изготовителя.

*Техническое описание ЭВМ* содержит общие сведения о назначении машины, ее характеристиках, принципах работы. В описании приводится принцип представления чисел и команд, поясняется взаимодействие устройств при выполнении различных операций. Кроме того, дается описание элементов машины, сопровождающееся конструктивными, структурными, функциональными и принципиальными схемами, а также временными диаграммами взаимодействия отдельных узлов при выполнении различных операций.

В качестве приложений к техническому описанию могут придаваться описания содержательной части математического обеспечения, а также различные справочные данные по размещению, монтажу, маркировке и т. п. В состав приложений входит описание стеновой аппаратуры, альбомы с электрическими схемами блоков и модулей, всевозможные таблицы и т. д.

*Инструкция по эксплуатации ЭВМ* содержит сведения, необходимые для организации правильной эксплуатации машины и поддержания ее в работоспособном состоянии. Инструкция является основополагающим документом, регламентирующим все необходимые работы на ЭВМ, и включает следующие разделы:

- порядок установки, монтажа и демонтажа машины;
- режимы работы и управления ЭВМ;
- инструкция по программированию и правила пользования системой математического обеспечения;
- порядок проведения профилактических работ на машине и ее технического обслуживания.

**Планирование эксплуатации ЭВМ.** Планирование является основной рациональной организации эксплуатации ЭВМ. Оно служит для определения конкретной программы действий обслуживающего персонала на какой-либо календарный срок.

Исходными данными для планирования эксплуатации ЭВМ являются: характер и объем загрузки машин; технические и эксплуатационные характеристики ЭВМ; трудоемкость всех видов работ, выполняемых на машине при ее обслуживании; количество обслуживающего персонала и уровень его квалификации.

Различают следующие виды планирования: оперативно-календарное, организационно-технических мероприятий, обеспечения эксплуатации.

*Оперативно-календарное планирование* заключается в составлении планов загрузки машины и работы обслуживающего персонала, исходя из объема запросов на машинное время, ресурса и норм технического обслуживания ЭВМ.

При оперативно-календарном планировании разрабатываются: план выделения машинного времени, диспетчерский план использования машин и расписание работы дежурных смен.

План выделения машинного времени обычно составляется на 7—10 сут. Непосредственное планирование загрузки ЭВМ осуществляется с помощью диспетчерского плана, при составлении которого учитывается характер использования машин [обработка заказов (программ) на подготовленных носителях, которые оператор пропускает на машине, работа с терминалами, через которые пользователи имеют возможность напрямую обращаться к машине].

*Планирование организационно-технических мероприятий* заключается в составлении программы работы обслуживающего персонала на определенный календарный период эксплуатации ЭВМ.

Обычно планы организационно-технических мероприятий составляются ежемесячно и включают в себя вопросы, связанные:

с порядком выполнения профилактических работ на ЭВМ исходя из годовых планов проведения этих работ, предполагаемой загрузки машины и вида обслуживания (централизованное или нет);

с проведением проверок технического состояния устройств ЭВМ, запасных блоков, контрольно-измерительной аппаратуры и т. д.;

с проведением каких-либо ремонтных работ и др.

В планах указываются сроки выполнения намеченных работ и их ответственные исполнители.

*Планирование обеспечения эксплуатации ЭВМ* сводится к определению потребности в расходных материалах, ЗИП, среднем и капитальном ремонтах устройств, блоков и узлов ЭВМ. На основании расчетов требуемого количества материалов, ЗИП и т. п. составляются заявки и подаются в соответствующие органы.

Определение потребности планового ремонта устройств, блоков и узлов проводится исходя из их технического состояния и степени выработки ими установленных ресурсов.

**Ведение учетной документации.** В процессе эксплуатации машины необходимо вести журнал технической эксплуатации ЭВМ и журнал учета машинного времени.

Первый из них ведется ответственным лицом из состава дежурной смены обслуживающего персонала и служит рабочим документом, в котором указываются сведения о всех проводящихся на машине работах. В него ежемесячно заносятся итоговые данные по продолжительности работы машины и указывается время, затраченное на профилактические работы и текущий ремонт, суммарное время планового отключения ЭВМ, а также данные о всех неисправностях, которые имели место в процессе эксплуатации ЭВМ.

В журнале учета машинного времени по каждой решаемой задаче указываются наименование, номер, дата и время начала и конца ее решения, а также фамилия или шифр пользователя. При обслуживании внешних пользователей эти данные автоматически фиксируются машиной (их, кроме времени решения задачи, указывает пользователь при обращении к машине) и выдаются оператору, который затем вносит все эти данные в журнал.

Накопленная в журналах информация позволяет дать количественную оценку эксплуатационных свойств машины, проанализировать качество ее работы и выработать рекомендации по улучшению обслуживания ЭВМ.

## ГЛАВА ВТОРАЯ

### КОНТРОЛЬ РАБОТЫ ЭВМ

#### 2-1. КЛАССИФИКАЦИЯ ОШИБОК И НЕИСПРАВНОСТЕЙ ЭВМ

Все неисправности машины можно классифицировать по следующим признакам: характеру возникновения, времени существования, внешнему проявлению, степени влия-

ния на работоспособность машины и причинам возникновения.

Ошибки возникают по разным причинам и могут носить систематический характер (отказы) или случайный (сбои).

Отказ — это событие, заключающееся в полной или частичной утрате машиной (системой) работоспособности. Отказ ЭВМ — это такое нарушение ее работоспособности, для восстановления которой требуются определенные действия по ремонту (замене) неисправного элемента, блока или устройства.

По характеру возникновения отказы могут быть внезапными или постепенными.

Внезапные отказы возникают в результате резкого изменения одного или нескольких основных параметров элементов машины, вызывая при этом нарушение ее логической структуры. К таким отказам относятся короткие замыкания, обрывы в электрических цепях, нарушение контактов, пробои диэлектриков и др. Внезапные отказы не поддаются прогнозированию и обнаруживаются лишь после появления.

Постепенные отказы возникают в результате постепенного изменения параметров элементов машины, например, по причине их старения. Причиной старения являются сложные физико-химические процессы, происходящие в элементах (структурные изменения в полупроводниковых материалах). Поскольку эти процессы протекают постепенно, появляется возможность предсказывать их возникновение, т. е. прогнозировать их. Большинство таких отказов выявляется и устраняется при проведении профилактических работ. В зависимости от степени ухода параметра постепенные отказы могут привести к постоянным или случайным ошибкам в вычислениях.

Отказы могут быть зависимыми и независимыми. В первом случае отказ возникает в результате действия другого отказа. В сложных схемах машины между элементами существуют такие взаимосвязи, которые могут существенно затруднить выявление первопричины отказа. Такие взаимосвязи характерны для цепей синхронизации, управления и др., появление ошибки в которых может нарушить работу целого устройства машины.

Под независимым понимают отказ, возникновение которого не связано с действием другого отказа. К таким отказам относятся, например, нарушение контактов, поломка механических узлов и деталей и др. Как правило, такие



отказы не вызывают больших трудностей при их локализации и устранении.

По внешнему проявлению отказы подразделяют на явные и скрытые. Явные отказы — это пропадание напряжения, поломка механических узлов, обрывы магнитных лент и т. п. К скрытым отказам можно отнести нарушение контактов в разъемах, появление микротрещин в печатных платах и др.

По степени влияния на работоспособность машины отказы могут быть частичными и полными. Если в результате возникновения отказа машина может продолжать работать, пусть даже с потерей первоначальной производительности, то такой отказ считают частичным или неполным. Если же такой возможности нет, то отказ является полным.

Наряду с отказами для ЭВМ характерно проявление другого вида ошибки — сбоя.

Под сбоем понимается кратковременное нарушение нормальной работы машины вследствие кратковременного воздействия на один или несколько элементов ЭВМ внешних помех или кратковременного изменения параметров элементов. После сбоя машина длительное время может работать нормально. Сбой сопровождается искажением информации при операциях передачи, хранения или 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 необходимо устранить неисправность в аппаратуре, то при сбое для получения правильного решения задачи требуется восстановить лишь достоверность информации путем ее повторной передачи или повторного прогона искаженной части программы.

Таким образом, возникновение неисправностей в машине вызывается появлением случайных событий — отказов и сбоев, которые в информационном плане приводят к появлению ошибок в работе ЭВМ.

При разработке системы контроля необходимо учитывать, что с точки зрения вероятности возникновения ошибок различные устройства машины не одинаковы. В частности, наиболее слабым местом являются электромеханические устройства (устройства ввода-вывода, накопители на магнитных лентах, дисках, барабанах), на долю которых приходится основная часть неисправностей, возникающих в машине. Неисправности в этих устройствах обнаружи-

ваются сравнительно легко, однако процесс их восстановления довольно трудоемок (замена износившихся деталей, регулировка люфтов и зазоров в механических узлах и т. п.). Среди электронных устройств машины наибольшее число неисправностей приходится на долю оперативных запоминающих устройств, которые являются наиболее критичными к изменениям питающих напряжений и колебаниям температуры окружающей среды.

## **2-2. ОСНОВНЫЕ ЗАДАЧИ КОНТРОЛЯ И ДИАГНОСТИКИ ЭВМ**

По мере расширения области применения ЭВМ все большее значение приобретала задача обеспечения правильности их работы. Машины первых поколений использовались главным образом в неоперативном режиме для последовательной обработки программ, и если ошибки в их работе своевременно не обнаруживались, это не имело сколько-нибудь серьезных последствий. Современные машины все больше используются в оперативном режиме, работая в реальном времени в качестве средств управления полетами космических аппаратов, сложными технологическими процессами, системой противоракетной защиты и т. п. В любом из таких применений неправильное функционирование вычислительной машины может привести к тяжелым последствиям. Поэтому эффективность контроля ЭВМ стала ее важной характеристикой.

Для повышения надежности работы вычислительных машин в настоящее время применяются два принципиально различных способа борьбы с неисправностями компонентов, каждый из которых можно реализовать тем или иным путем. Первый из этих способов — резервирование аппаратуры — позволяет быстро нейтрализовать возникшую неисправность. Второй — обнаружение неисправности диагностической процедурой — позволяет локализовать неисправность и своевременно устранить ее.

Нейтрализация неисправностей фактически служит лишь для того, чтобы отсрочить их проявление (при постоянном резервировании элементы, блоки или узлы машины работают в параллель и выход любого из них из строя не нарушает работы системы, пока не исчерпывается весь заложенный резерв). Однако с течением времени неисправностей может накопиться столько, что они уже не смогут быть нейтрализованы и в расчетах появятся ошибки. Поэтому такой способ применяется в первую очередь в систе-

мах, где правильную работу машины необходимо гарантировать на определенный период времени, а ремонт затруднен или просто невозможен (например, управление космическими кораблями). Способ, основанный на обнаружении неисправности в сочетании с ремонтом, целесообразен тогда, когда остановки машины допустимы, а неисправности нельзя оставлять необнаруженными. Это соответствует случаю неоперативной работы машины в режиме последовательной обработки программ. Выполненные работы в этом случае после обнаружения и устранения неисправности можно выполнить повторно, хотя это и неудобно для пользователя.

В условиях управления процессами в реальном времени и для обработки информации в оперативном режиме (например, в системах с разделением времени) очень важно предотвратить полный выход системы из строя в течение весьма длительного времени. Это вызывает необходимость использования избыточности для обеспечения правильной работы системы при наличии неисправностей, а также обуславливает необходимость использования средств обнаружения и диагностики неисправности для обеспечения ее быстрого удаления во время профилактики.

Первые вычислительные машины обслуживались опытными техниками, которые нередко пользовались интуитивными методами, основанными на их опыте работы с машиной. В качестве вспомогательных средств они использовали так называемые проверочные программы-тесты, которые указывают лишь на наличие ошибки в том или ином устройстве, нисколько не конкретизируя ее. Ремонт машины при таких условиях существенно затруднен.

Трудность поиска неисправностей в современных системах усугубляется тем, что строятся эти системы на базе интегральных схем, в которых нельзя наблюдать сигналы на внутренних точках схемы.

Следовательно, необходимы такие тест-процедуры, при использовании которых для получения сведений о наличии неисправностей и местах их нахождения требовались бы только естественные входы и выходы схемы, содержащей десятки или даже сотни логических элементов. Необходимо также, чтобы эти процедуры были в значительной степени автоматизированными, поскольку работающая система не должна иметь заметного перерыва в работе, а это требует обнаружения неисправностей за короткие промежутки времени.

Система контроля вычислительной машины представляет собой совокупность программных и аппаратурных средств, предназначенных для определения ее технического состояния и поддержания необходимого уровня эффективности работы ЭВМ.

Процессы определения технического состояния (процесс диагноза) и поддержания заданного уровня эффективности работы ЭВМ реализуются средствами проверок, поиска неисправности и рационального обслуживания. В процессе диагноза с помощью системы контроля машины решается ряд последовательно связанных задач:

- ① контроля — обнаружение ошибки в работе ЭВМ;
- ② классификации — определение характера ошибки (сбой или отказ);
- ③ диагностики — поиск места отказавшего элемента;
- ④ коррекции — устранение ошибки, замена отказавшего элемента.

Перечисленные задачи могут решаться в машине в различных режимах: оперативном (одновременно с решением основной задачи машиной); автоматически либо с привлечением оператора; профилактических проверок.

В разных задачах контроля возможно использование различных технических средств. Ими, как уже указывалось, являются:

программные средства, реализующиеся с помощью своих определенных форм представления;

аппаратурные средства, основанные на использовании схемных решений с учетом общей структуры машины; комбинации программных и аппаратурных средств.

Введение этих средств в состав вычислительной машины не является необходимым при ее работе непосредственно для вычислений, поэтому они являются дополнительными, избыточными, что и вызывает затруднения при решении вопросов практической реализации различных методов контроля.

Современная ЭВМ представляет собой сложную систему, производящую ряд специфических операций по обработке информации, которые необходимо контролировать. К ним относятся:

операции, связанные с вводом и выводом информации;

хранение информации в запоминающих устройствах и передача ее между устройствами;

арифметическая и логическая обработка информации;

выполнение управляющих операций, определяющих ход вычислительного процесса.

Но современные вычислительные машины — это не только оборудование, но и хорошо развитые системы математического обеспечения, которые составляют существенный процент от общей стоимости ЭВМ. Это предопределило появление новых объектов контроля — системных программ (супервизоров, диспетчеров, трансляторов и др.). Для их проверки широко используются методы, разработанные для основного оборудования ЭВМ.

Разнообразие функциональных устройств вычислительных машин, их сложность создали предпосылки для использования большого числа разновидностей методов контроля, как программных, так и аппаратурных.

### 2-3. СТРУКТУРА СИСТЕМЫ КОНТРОЛЯ И ДИАГНОСТИКИ ЭВМ

В зависимости от метода, положенного в основу контроля ЭВМ, различают два основных вида контроля: программный и аппаратурный. Каждый из них может использоваться как в оперативном режиме, т. е. в процессе работы машины, так и в режиме профилактических проверок, причем контроль может быть автоматическим или с привлечением оператора.

Программный контроль ЭВМ основан на использовании специальных программ, контролирующих работу машины. Он подразделяется на программно-логический и тестовый.

Программно-логический контроль основан на том, что в основную рабочую программу вводятся дополнительные операции, при выполнении которых получается избыточная информация, необходимая для обнаружения и исправления ошибок. Наличие избыточности в информации позволяет, например, находить те или иные контрольные соотношения, которые связывают получаемые в процессе расчета значения и которые можно проверять по программе в конце каждого этапа вычислений. Так, если вычисляются значения синусов и косинусов, то правильность их вычислений можно проверить по известному соотношению: сумма квадратов синуса и косинуса равна 1. Часто прибегают к двойному просчету, при котором избыточность информации создается путем повторения вычислений, а контрольные соотношения — это совпадение результатов первого и второго просчетов.

Программно-логический контроль не требует применения специальной аппаратуры и позволяет обнаруживать ошибки, обусловленные случайными сбоями, в процессе проведения вычислений. Однако этот вид контроля приводит к значительному увеличению времени решения задачи.

Тестовый контроль предназначен для проверки правильности работы ЭВМ или ее отдельных устройств с помощью специальных программ — тестов. Контроль с помощью тестов сводится к выполнению вычислительной машиной определенных действий над исходными числами и сравнению полученных результатов с известными. В случае несовпадения ответов фиксируется ошибка.

Все тесты подразделяются на наладочные, проверочные и диагностические.

Наладочные тесты служат для проверки правильности функционирования устройств и блоков во время наладки машины. Эти тесты предназначаются для обнаружения грубых ошибок (ошибки в монтаже, логике работы отдельных узлов и т. д.).

Проверочные тесты предназначены для периодической проверки работоспособности машины и для обнаружения неисправностей в процессе эксплуатации. Эти тесты обеспечивают более полный контроль и создают более разнообразные режимы работы узлов машины. Однако как наладочные, так и проверочные тесты свидетельствуют лишь о факте появления ошибки в том или ином устройстве, не указывая места ее возникновения.

Диагностические тесты служат не только для обнаружения ошибки, но и для локализации места неисправности.

Наладочные тесты используются для проверки центральных процессоров, устройств ввода-вывода, оперативной и внешней памяти. Они являются самостоятельными программами и выполняются без помощи операционной системы.

Проверочные и диагностические тесты работают под управлением специальной тестовой программы проверки — монитора (часть управляющей программы), которая осуществляет вызов, выполнение каждого отдельного теста и управление им. Проверка устройств может производиться как в профилактическом, так и в оперативном (мультипрограммном) режиме, т. е. наряду с выполнением других программ (последний вид проверки устройств машины называется неавтономной проверкой).

Программа проверки устройств позволяет:  
периодически осуществлять профилактическую проверку работы устройства;

при появлении ошибок в работе устройства локализовать места возникновения этих ошибок;

убедиться в правильности работы устройства после устранения ошибки или внесения в устройство технических изменений.

Для различных типов устройств существуют свои тестовые программы.

В современных вычислительных системах запуск тестов может производиться и автоматически по сигналу ошибки с контрольных схем машины. При этом после локализации ошибки система саморемонтируется (реконфигурация системы). В менее мощных системах процесс локализации ошибки сопровождается подачей оператору соответствующего сигнала.

Аппаратурные средства контроля создаются введением в состав машины специального дополнительного контрольного оборудования, работающего независимо от программы. Аппаратурный контроль обеспечивает проверку правильности функционирования машины практически без снижения ее быстродействия. Однако использование только аппаратурного контроля приводит к значительному усложнению и удорожанию ЭВМ. Кроме того, введение в состав машины большого количества избыточного сложного оборудования может привести к снижению ее общей надежности. Поэтому в современных ЭВМ применяется комбинированный метод контроля, представляющий собой сочетание программных и аппаратурных средств.

С целью предупреждения существенного искажения обрабатываемой информации (т. е. до того предела, когда она не сможет быть восстановлена) обнаружение ошибок в машине должно производиться непрерывно. Поэтому эта функция возлагается на быстродействующие аппаратурные средства контроля, которые позволяют практически полностью совместить во времени выполнение основных операций машины и необходимых контрольных операций. Локализация места возникновения неисправности и ликвидация последствий сбоев при этом возлагаются на программный контроль.

Комбинированный метод контроля позволяет при незначительном снижении эффективного быстродействия машины существенно сократить время поиска и устранения ошибок

и общий объем дополнительного оборудования ЭВМ, требуемого для этих целей.

В целом эффективность системы контроля ЭВМ характеризуется следующими показателями:

отношением количества оборудования, охваченного системой контроля, к общему количеству оборудования ЭВМ; вероятностью обнаружения системой контроля ошибок в работе ЭВМ;

степенью детализации, с которой система контроля указывает место возникновения ошибки (точность диагноза);

отношением количества оборудования системы контроля к общему количеству оборудования ЭВМ.

Следует отметить, что эффективные системы контроля и диагностики могут быть созданы при условии, если их разработка и проектирование машины проводятся одновременно и взаимосвязанно. Только такой подход позволяет создавать наиболее рациональный контроль с минимальными затратами на его реализацию.

## ГЛАВА ТРЕТЬЯ

### ОБЩИЕ ПРИНЦИПЫ ОБНАРУЖЕНИЯ ОШИБОК И РЕМОНТА ЭВМ

#### 3-1. СИСТЕМЫ АВТОМАТИЧЕСКОГО КОНТРОЛЯ И ДИАГНОСТИКИ И ИХ ВЗАИМОДЕЙСТВИЕ

В процессе выполнения программы необходимо предотвратить распространение возможной ошибки, так как в противном случае сильно усложняется и удлиняется процедура проверки правильности ее выполнения, а также процедура определения и устранения искажений данных и промежуточных результатов расчета. Для этого необходимо обеспечить возможность обнаружения ошибки в работе машины как можно ближе к моменту ее возникновения, т. е. в момент наличия неискаженной информации, необходимой для повторного запуска данной команды, части программы из нескольких команд или сегмента программы. С этой целью ЭВМ оснащается системой автоматического контроля достоверности функционирования машины, останавливающей процесс вычисления при появлении ошибки. Наличие такой системы позволяет в значительной мере освободить пользователя от необходимости жесткого кон-



тролирования работы машины программными методами и снизить связанные с этим потери времени.

Поскольку последствия, вызываемые ошибкой, различны в зависимости от того, вызвана ошибка сбоем или отказом, необходимо диагностирование (определение) характера и места возникновения ошибки с тем, чтобы в случае сбоя восстанавливались достоверность информации и выполнение программы, а в случае отказа — оператор извещался о необходимости ремонта машины.

Вообще говоря, между контролем и диагностикой трудно провести достаточно четкую границу, поскольку обе эти процедуры тесно связаны между собой.

Контроль — это проверка правильности работы объекта (элемента, узла, устройства). Правильно работает устройство — схема контроля не вырабатывает никаких сигналов (в некоторых системах вырабатывается сигнал нормальной работы), неверно работает устройство — схема контроля выдает сигнал ошибки. На этом, собственно, заканчиваются функции контроля. Другими словами, контроль — это проверка: правильно—неправильно.

Процесс диагноза можно разделить на отдельные части, называемые элементарными проверками. Элементарная проверка состоит в подаче на объект тестового воздействия и в измерении (оценке) ответа объекта на это воздействие. Алгоритм диагноза определяется как совокупность и последовательность элементарных проверок вместе с определенными правилами анализа результатов последних с целью отыскания места в объекте, параметры которого не отвечают заданным значениям. Таким образом, диагностика — это тоже контроль, но контроль последовательный, направленный на отыскание неисправного места (элемента) в диагностируемом объекте. Обычно диагностика начинается по сигналу ошибки, вырабатанному схемами контроля машины.

Систему автоматического контроля и диагностики часто называют системой обнаружения ошибок. Мы также будем пользоваться этим понятием.

Рассмотрим теперь общий принцип организации такой системы.

Возникновение ошибки в каком-либо устройстве машины вызывает сигнал ошибки, по которому выполнение программы приостанавливается. При этом на сигнальном световом табло на пульте оператора высвечивается соответствующий разряд регистра ошибок, указывающий место обнаружения ошибки (устройство, узел и т. п.).

По сигналу ошибки сразу же начинает работать система диагностики, которая во взаимодействии с системой контроля машины выполняет следующие функции:

1. Распознавание (диагностирование) характера ошибки (сбой, отказ).

2. Повторный пуск программы (части программы, операции), если ошибка вызвана сбоем.

3. Локализация места неисправности, если ошибка вызвана отказом, с последующим ее устранением путем автоматической замены (или отключения) вышедшего из строя элемента или замены с помощью оператора.

4. Запись в память машины информации о всех происшедших сбоях и отказах для дальнейшего анализа.

На рис. 3-1 представлена общая логика работы систем автоматического контроля и диагностики. Работа начинается с записи в специальные регистры или память ситуации в машине (состояние запоминающих элементов и шин передачи данных неисправного устройства) в момент появления ошибки. Затем производится обработка сигнала прерывания от схем контроля, которая начинается с записи состояний регистров процессора и старого слова состояния программы в соответствующие места памяти и выборки нового слова состояния программы диагностической системы.

Диагностика характера ошибки осуществляется повторением операции, в которой обнаружилась ошибка. Операция может быть повторена, если не пройден «порог повторения», т. е. если используемая в операции информация (данные, адреса) не исказилась. Если порог пройден, то программа повторяется с ближайшей контрольной точки (предусматривается программистом), сохраняющей необходимые для повторения промежуточные данные. Порог повторения определяет специальная программа обработки сбоев по записанной ситуации, соответствующей появлению ошибки.

Если при повторном запуске операции или части программы ошибка не повторяется, то ее появление классифицируется как сбой и управление передается в точку прерывания программы для продолжения работы. При этом в память записывается дополнительная информация о состоянии в момент сбоя машины (выполняемые операции, название программы, адрес команды, операнды, время). Подобная информация, систематически накапливаемая в запоминающем устройстве, обрабатывается затем специальной программой, вырабатывающей определенные рекоменда-

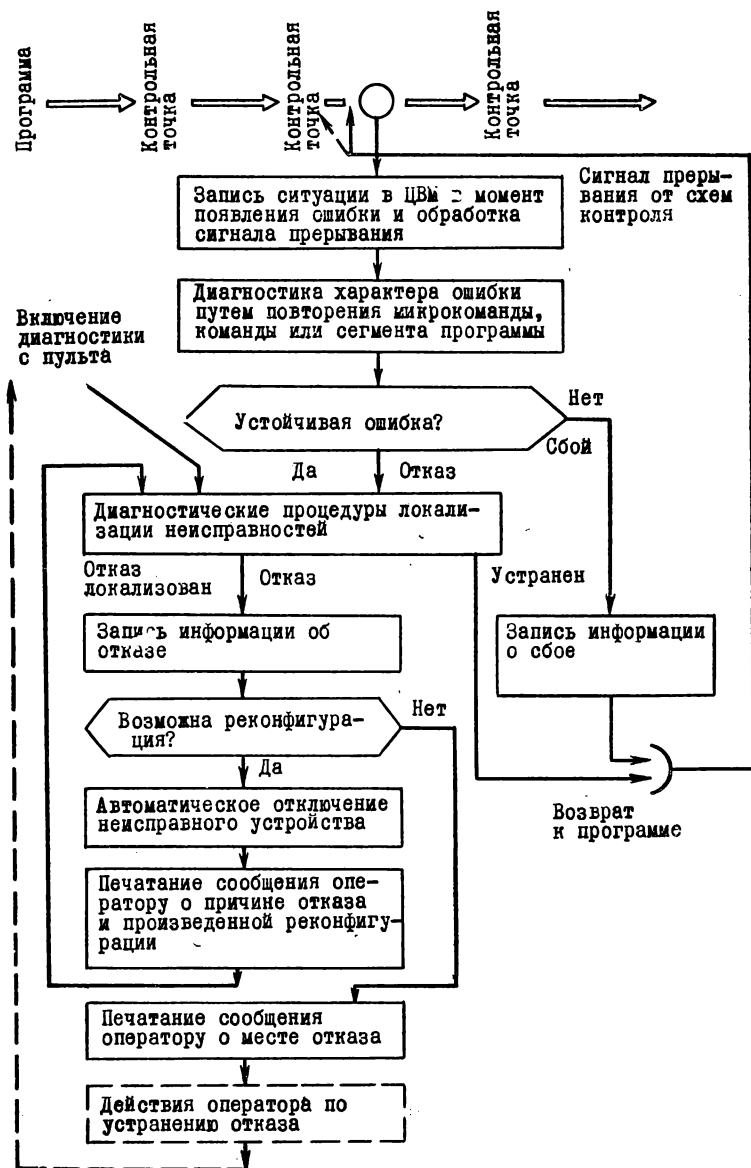


Рис. 3-1. Блок-схема алгоритма работы систем контроля и диагностики.

дацин, учитываемые при проведении профилактических работ.

Если при повторении ошибка сохраняется (проверка осуществляется несколько раз), то в работу вступает диагностическая система для отыскания места неисправности. (Диагностические процедуры могут производиться и путем запуска их с пульта оператора).

В комплекс диагностических программ входят программы по диагностике всех основных устройств машины — процессора, оперативной памяти, каналов и периферийных устройств. Работа диагностических программ и схем диагностики производится под управлением диагностического монитора, который получает команду от супервизора на диагностирование данного устройства. В распоряжении монитора имеется также специальная команда ДИАГНОСТИКА, которая управляет работой схем диагностики.

После локализации места неисправности система диагностики посылает на пульт оператора код неисправности, по которому оператор, пользуясь таблицами неисправности, устраняет неисправный блок. Затем оператор с пульта запускает диагностические программы, удостоверяющие исправное состояние машины. Если неисправность устранена, возврат к контрольной точке программы происходит автоматически.

**Система обнаружения ошибок и проектирование машины.** Все необходимые функции, которые выполняются системой контроля машины при обнаружении ошибки, должны быть четко определены еще до начала проектирования. Не предусмотренная заранее реакция системы на ту или иную ошибку не всегда может быть заложена в законченную систему.

Однако мало определить эти функции. В самой конструкции системы необходимо предусмотреть и возможность их выполнения. Если, например, на систему контроля возлагается функция проведения повторного вычисления, прерванного вследствие обнаружения ошибки, то система повторного пуска должна получить необходимую информацию о контрольной точке в программе, с которой можно продолжить расчет. Если такая информация не предусматривается конструкцией системы контроля, то поставленная задача не сможет быть выполнена.

Проектирование системы обнаружения ошибок должно предусматривать и возможность проверки работоспособности схем контроля, поскольку эти схемы сами могут выходить из строя, а кроме того, могут иметь конструктивные недостатки, устранение которых требует определенного статистического анализа. По этой причине в машине предусматривается проверка работоспособности схем контроля с помощью имитации ошибок. Хотя такая проверка является довольно грубой, она необходима (особенно для отладки схем), поскольку ошибки на машине возникают редко. Окончательная же проверка системы

контроля производится на реальных ошибках, причем в течение длительного времени работы машины.

Определение функций, которые должна выполнять система контроля машины, является непростым вопросом. Безусловно, на нее нельзя возлагать задачу выявления всех без исключения ошибок, поскольку такая система контроля окажется очень сложной и дорогой. Кроме того, она неизбежно снизит, и в значительной степени, общую эффективность работы машины. Поэтому при проектировании системы контроля проектировщики исходят из предположений относительно возможных типов неисправностей для данных схем машины. Например, типичными неисправностями являются: выходной сигнал не меняет своего значения — короткое замыкание или обрыв цепи в схеме; логический блок одинаково реагирует на отдельные комбинации входных сигналов — неисправность элементов входной цепи; неисправен целый модуль — нарушена цепь питания или поврежден модуль и т. п. Это отказы.

Однако более трудным видом ошибок являются сбои, причинами которых могут быть электромагнитные помехи в логических цепях, изменение параметров отдельных элементов, плохие контакты в разъемных соединениях и т. п. Сбои предусмотреть сложнее. А ведь именно предусмотренные типы неисправностей будут в дальнейшем фиксироваться системой контроля как ошибки, которые должны быть обнаружены и исправлены.

Во-первых, многое здесь зависит от предусмотренных самими разработчиками различных типов неисправностей. Во-вторых, предположения о неисправностях делаются в соответствии с двумя критериями: может ли произойти предполагаемая неисправность и каковы затраты на обнаружение данного типа неисправностей. Некоторые неисправности, например, приходится устранять дублированием ввиду сложности их выявления. Таким образом, процент обнаруживаемых ошибок и стоимость соответствующей системы контроля являются важными параметрами, которые необходимо принимать во внимание при проектировании системы.

### **3-2. ОСНОВНЫЕ ТРЕБОВАНИЯ К СИСТЕМЕ АВТОМАТИЧЕСКОГО КОНТРОЛЯ И ДИАГНОСТИКИ**

#### **А. Общие требования к системе**

Как правило, основной процент ошибок, возникающих в вычислительных машинах в процессе выполнения операций, составляют случайные одиночные ошибки, т. е. сбои.

В машинах первых поколений отсутствовали надлежащие схемы контроля хода вычислительного процесса, в результате чего ошибки обнаруживались по итогам выполнения всей программы или ее отдельных частей, если программа была большая. Естественно, это приводило к большим потерям времени и снижению общей производительности машины. Чтобы избежать таких потерь, программисты широко использовали программные методы контроля, которые хотя и исключали влияние случайных ошибок на

правильность выполнения программы, но существенно увеличивали общее время процесса вычисления, даже если машина работала без сбоев. Кроме того, программные методы контроля не всегда способны обнаружить систематическую ошибку (при выходе из строя какого-либо элемента двойной просчет, например, даст одинаковые, но неверные результаты), что опять-таки приведет к неправильному выполнению программы.

Современные вычислительные машины оснащены развитым аппаратным контролем, который позволяет вовремя обнаружить возникшую ошибку и тем самым предотвратить искажение обрабатываемой информации. Многие (особенно большие) машины имеют систему контроля, которая восстанавливает прерванную случайной ошибкой работу автоматически. Это достаточно сложная система, которая реализует свои функции при следующих условиях:

определенное время информация о состоянии выполняемой программы должна храниться в некоторых ее контрольных точках (адреса текущих контрольных точек могут находиться, например, в регистрах). В случае возникновения ошибки эта информация будет использована машиной для осуществления повторного пуска программы с одной из таких контрольных точек;

программа должна иметь возможность выполнять такие действия, как запоминание или вывод информации;

после обнаружения ошибки должен осуществляться повторный пуск программы;

операционная система должна возвращать машину в состояние, соответствующее данной контрольной точке, производить необходимую корректировку искажений рабочей информации и давать повторный пуск программы.

Подобная система контроля позволяет исключить остановы в работе из-за случайных ошибок и избавляет от необходимости вручную выполнять процедуру восстановления программы в таких случаях, освобождая тем самым обслуживающий персонал от рутинной работы и потери дорогого машинного времени.

Для реализации возложенных на систему контроля функций она должна удовлетворять следующим основным требованиям, закладываемым в нее при проектировании:

1. Обнаруживать определенный процент неисправностей, возникающих в схемах машины.

2. Производить диагноз, локализуя место возникновения ошибки.

3. Давать информацию, необходимую для замены блока, имеющего неисправность, если ошибка повторяется.

4. Обнаруживать ошибку достаточно быстро, чтобы не дать ей распространиться в вычислительном процессе, и позволить машине повторить операцию (в крайнем случае часть программы), прерванную ошибкой.

5. Классифицировать ошибки для обеспечения правильной реакции на них со стороны машины: а) отказ, б) сбой, в) перемежающаяся неисправность (многократные сбои), г) программные ошибки (запрещенный код и др.), д) ошибки в памяти (оперативная, диски, ленты и др.), е) ошибки оператора (не заправлена лента, в печати кончилась бумага и др.), ж) ошибки в схемах контроля.

Рассмотрим суть этих требований.

1. Как уже говорилось, на систему контроля трудно, да и нецелесообразно возлагать обязанности по обнаружению всех без исключения типов ошибок. В принципе возможность обнаружения неисправностей всегда может быть увеличена, но это связано с удорожанием и усложнением системы контроля, что должно быть оправданным.

Обычно система обнаружения ошибок проектируется с таким расчетом, чтобы обнаруживались все одиночные ошибки. Связано это с тем, что, во-первых, надежность интегральных схем, на которых строятся современные машины, достаточно высокая и наиболее вероятным типом ошибок являются одиночные, возникающие в машине вследствие влияния помех. Для примера укажем, что если гарантированный срок службы ламп, которые использовались при построении машин первого поколения, составлял  $10^3$  ч, то срок службы даже первых серийных микромодулей равнялся  $10^6$ — $10^7$  ч. Во-вторых, редко бывает, чтобы элемент отказывал сразу. Обычно все начинается с изменения параметров его работы, которое сопровождается появлением прежде всего одиночных ошибок. Следовательно, система, которая способна обнаружить одиночную ошибку, отреагирует и на появление многократных ошибок, т. е. такая система контроля практически сможет обнаруживать большинство возникающих неисправностей.

2. Желательность определения места возникновения ошибок с точностью до сменного блока (ТЭЗ) очевидна. Прежде всего такая возможность существенно облегчает работу обслуживающего персонала по устранению неисправности, позволяя тем самым сводить к минимуму простой машины. Если же машина имеет резервные элементы того же типа и в случае неисправности может автоматически перейти на резервный, то появляется возможность исключать неисправные элементы, не прерывая работы машины.

3. Третье требование дополняет предыдущее и имеет в виду форму выдачи требуемой информации. Дело в том, что в разных машинах система обнаружения может обладать различными возможностями и, следовательно, использоваться по-разному:

для простейшей сигнализации на пульт оператора о наличии ошибки в устройстве (процессоре, канале и т.п.);

для сигнализации на пульт оператора в сочетании с другими индикаторами, уточняющими место возникшей неисправности;

для автоматического отключения сменного блока с выдачей соответствующей информации оператору (при этом в одних машинах отключенный блок может автоматически заменяться на резервный, в других — после отключения блока или функционального узла производится автоматическое перераспределение имеющихся в машине ресурсов — реконфигурация).

Во всех этих случаях требования к системе обнаружения различны и должны определяться в соответствии с общей структурой проектируемой машины и задачами, стоящими перед ней.

4. Возможность повторения операции, прерванной возникшей ошибкой, или определенной части программы является одним из основных требований, предъявляемых к системе обнаружения ошибок. Повторение возможно в случае, если сохранено достаточное количество данных, к которым можно вернуться для продолжения нормальной работы. Отсюда вытекает и требование быстрого обнаружения ошибки, чтобы она не успела исказить всей или большей части

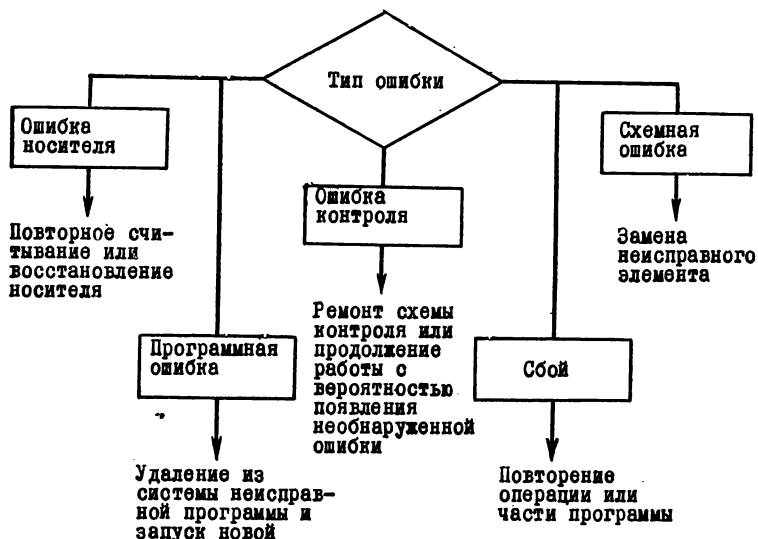


Рис. 3-2. Типы ошибок и операции, связанные с ними.

информации, необходимой для повторного пуска данного участка программы. Вопрос, на каком уровне возможно повторение (микрооперации, операции или сегмента программы), зависит от чувствительности системы контроля машины. Во всяком случае на уровне микрооперации повторение способно осуществлять лишь небольшое количество машин.

5. Возможность различения ошибок имеет большое значение для системы обнаружения, так как машина должна понять, каким образом ей реагировать на принятые от схем контроля сигналы и какую следует произвести операцию в зависимости от типа ошибки. В качестве примера на рис. 3-2 приведены некоторые операции, которые могут иметь место при возникновении того или иного типа ошибки.



Как и все другие требования, вид классификации ошибок и способ их определения должны планироваться на стадии проектирования системы.

## **Б. Средства проверки схем контроля**

Как логические схемы машины, так и схемы, осуществляющие контроль правильности ее функционирования, могут явиться источником ошибок. Неправильная работа схем контроля может привести к необнаруживанию возникшей ошибки и, следовательно, к искажению информации либо к появлению ложного сигнала ошибки, на который машина отреагирует как на ошибку реальную, нарушив тем самым нормальный ход вычислительного процесса. Поэтому схемы контроля также нуждаются в проверке их работоспособности.

Сложность вопроса заключается в том, что схемы контроля не проверяются автоматически в процессе их работы, поскольку над ними, как правило, нет других контролируемых схем (а если в каких-то случаях и есть, то эти последние сами оказываются в таком же положении). Следовательно, выход здесь один: периодический контроль этих схем с помощью специальных приемов.

Во многих случаях схемы контроля проверяются с помощью ошибок, создаваемых искусственно.

Проверка схем контроля информационных каналов производится путем подачи в эти каналы данных с неправильными контрольными разрядами. При этом комбинации контрольных разрядов подбираются таким образом, чтобы схемы контроля были проверены во всех возможных режимах. Обычно это выполняется посредством самой машины или программного устройства с помощью специальных диагностических программ.

Такие устройства, как дешифраторы и счетчики, используются для проверки их контрольных схем путем анализа условий возникновения сигналов ошибки при подаче на них неправильных контрольных разрядов.

Если нет возможности проверять те или иные схемы контроля на различных режимах, то необходимо создавать хотя бы одну искусственную ошибку, чтобы убедиться в правильности выработки схемой контроля сигнала ошибки и его надлежащего использования. Однако следует иметь в виду, что части схем контроля, не проверяемые программой, могут оказаться источником необнаруживаемых ошибок вследствие появления в них необнаруживаемых несправ-

ностей. Правда, подобных схем бывает немного, и для большей надежности их обычно дублируют.

Специальные контрольные тестовые программы входят в состав математического обеспечения машины, что позволяет оператору проводить тестирование схем контроля периодически.

### **В. Оценка эффективности контроля**

Эффективность работы системы контроля обычно определяется на вновь спроектированной машине с целью проверки соответствия реальной возможности системы запроектированной, а также дальнейшего совершенствования общей идеологии проектирования машин.

Процедура измерения эффективности контроля заключается в следующем. Прежде всего проводится анализ характера возможных ошибок и задается процент неисправностей каждого типа. После этого составляется таблица вероятных неисправностей.

Ошибки вводятся в машину путем включения в нее элементов с преднамеренно внесенными неисправностями. Машина должна реагировать на эти неисправности так, как это было предусмотрено при проектировании. Количество случаев правильной реакции машины подсчитывается и сравнивается с количеством введенных в машину неисправностей. Полученный процент обнаруженных ошибок позволяет сделать вывод, насколько машина отвечает заданным требованиям. В зависимости от этого конструкция системы сохраняется или дорабатывается.

### **3-3. ОБЩИЙ ПОДХОД К РЕМОНТУ МАШИНЫ**

Если поставить вопрос, что же такое ремонт машины, то ответ будет однозначный: выполнение комплекса организационных и технических мероприятий с целью приведения машины в работоспособное состояние. Однако методы ремонта могут существенно различаться в зависимости от видов машин.

Системы контроля машин первых поколений (тесты) фиксировали лишь факт наличия неисправности, локализуя место ее возникновения в лучшем случае с точностью до устройства. Далее все зависело от сноровки и умения технического персонала отыскать в сложных схемах устройства с помощью осциллографа конкретное место неисправности и устранить ее. В зависимости от вида ошибки на это ухо-

дили часы, иногда дни. С усложнением машин стало очевидно, что такой способ ремонта не может обеспечить возросшие требования к производительности машины и эффективности ее использования даже при самой высокой квалификации обслуживающего персонала. Поэтому основная часть обязанностей по контролю работы ЭВМ и устранению ошибок были возложены человеком на саму машину.

Вычислительные машины сегодняшнего дня сами контролируют правильность протекания вычислительного процесса, информируя оператора о том, что прошел сбой, но его последствия устранены и работа продолжается нормально. Лишь при появлении более сложных ошибок машина сигнализирует оператору о необходимости проведения ремонта, указывая при этом конкретное место возникшей неисправности. В наиболее развитых системах при этом работа не прекращается, а неисправный блок автоматически отключается до того момента, пока не будет отремонтирован и вновь введен в строй.

Таким образом, понятие и методика ремонта меняется в зависимости от совершенства ремонтируемой машины. Вместе с тем не следует думать, что теперь стало все просто. Во-первых, не все современные машины обладают такими возможностями саморемонта, о которых только что говорилось. Во-вторых, и в современных машинах возможны неисправности, реакция на которые не была предусмотрена в спроектированной системе контроля. В этом случае отыскание неисправности приближается к методам, которыми пользовались при ремонте машин прежних выпусков.

В целом можно сказать, что на продолжительность поиска неисправностей оказывают влияние такие факторы, как общая структура системы контроля машины, оснащенность ЭВМ проверочными и диагностическими тестами, наличие и состояние контрольно-измерительных приборов, качество таблиц неисправностей для данного типа машины, условия эксплуатации. Кроме того, сюда же можно отнести квалификацию и опыт обслуживающего персонала, его умение пользоваться тестами, таблицами неисправностей, а также знание методов поиска неисправностей в машине.

## **А. Виды ремонтных работ**

Ремонт на машине подразделяют на текущий (внеплановый) и профилактический (плановый). Необходимость в текущем ремонте появляется тогда, когда на машине возни-

кает неисправность. Поскольку возникновение неисправностей носит случайный характер, текущий ремонт заранее спланировать нельзя. Профилактический ремонт производится в целях предупреждения появления в машине неисправностей, обусловленных постепенным изменением параметров элементов; этот вид ремонта планируется на каждой машине заранее и проводится периодически.

Как при текущем, так и при профилактическом ремонте в число основных видов работ входят следующие:

обнаружение и устранение возникших в процессе эксплуатации неисправностей в устройствах машины;

замена неисправных блоков и модулей исправными;

ремонт и замена кабелей в шкафах, блоках и кабельных каналах;

настройка и регулировка отдельных блоков и устройств машины.

Часть из этих работ, особенно если они связаны с текущим ремонтом, выполняется силами обслуживающего персонала. Более сложные из них (ремонт кабелей, ТЭЗ, отладка устройств машины и т. п.) выполняются силами специализированных ремонтных организаций в порядке централизованного обслуживания ЭВМ.

К чему же сводится суть ремонтных работ на машине? Как уже отмечалось, основной причиной сбоев на машине являются помехи, вызываемые электромагнитными наводками или постепенным изменением параметров элементов схем. Ухудшение работы элементов приводит к возникновению неисправности в машине, которую необходимо устранить. Ремонт машины в этом случае может производиться методом замены сменных элементов, методом замены на резерв (при наличии резервирования) и путем профилактического обслуживания, предусматривающего более сложный вид регулировочных работ.

Метод замены сменного элемента (ТЭЗ) сводится к отысканию неисправности с точностью до такого сменного блока и замене его на исправный. Поиск неисправного ТЭЗ ведется с помощью самой машины и при наличии хорошо организованного контроля занимает немного времени. Данный метод ремонта ввиду блочной структуры построения ЭВМ и наличия запасных элементов широко распространен на практике и является основным при восстановлении машины.

Ремонт методом замены на резерв заключается в том, что вышедший из строя элемент (узел, устройство) заменяется

резервным с последующим устранением возникшей неисправности. В зависимости от машины резервирование может производиться вручную или автоматически. Помимо схемного резервирования (более подробно этот вопрос рассматривается в § 4-2), в машинах резервируются устройства ввода-вывода, накопители на магнитных дисках, лентах и др.

Если в процессе эксплуатации машины неисправности начинают возникать часто, то текущий ремонт заменяют более полным профилактическим обслуживанием, в результате проведения которого устраняются не только возникшие, но и потенциально возможные неисправности. Однако такая замена увеличивает затраты времени на восстановление машины и потому является нежелательной. При правильно организованной эксплуатации ЭВМ необходимости в подобной замене, как правило, не возникает.

**Система обнаружения ошибок и ремонт машины.** Следует подчеркнуть, что сокращение времени ремонта является одной из основных задач эксплуатации машины, поскольку продолжительное время ремонта является потерей производительности ЭВМ.

Большие затраты на обслуживание и длительный ремонт являются следствием трудностей диагностики ошибок. Поэтому наличие системы обнаружения и умелое ее использование может существенно облегчить задачу восстановления машины.

В развитых системах контроля предусматривается сигнализация, какой сменный блок следует заменить при обнаружении ошибки. В более простых системах процесс диагностирования может быть упрощен путем предоставления набора готовых рекомендаций, с помощью которых обслуживающий персонал производит диагностирование, выполняя ряд стандартизованных процедур. Хотя в этом случае система обнаружения имеет более простую структуру и не столь широкие возможности, без нее было бы очень трудно диагностировать причины случайных ошибок, так как случайные ошибки никак не фиксируются в машине, не имеющей системы обнаружения. Фиксирование же сбоя позволяет произвести ремонт до появления другой, более серьезной ошибки, поскольку некоторые неисправности, проявляясь сначала как случайные, превращаются затем в устойчивые. Следовательно, чем скорее удастся ликвидировать причину, вызвавшую ошибку, тем надежнее будет работать машина.

Однако сама по себе возможность фиксирования ошибки еще не означает возможности быстрого ликвидации неисправности. Многое здесь зависит от организации системы контроля. Например, если контроль передаваемой информации с одного регистра на другой производится после очистки передающего регистра, то оказывается трудным определить, когда произошла неправильная передача информации. Совсем другое дело, если контроль передачи производится до того, как передающий регистр будет очищен. В этом случае сигнал ошибки может быть использован для блокировки сброса передающего и приемного регистров. При этом обслуживающий персонал будет иметь информацию о состояниях обоих регистров, что позволит доста-

точно просто определить неправильную передачу тех или иных рядов, после чего тестирование и детальный анализ позволяет выявить неисправный сменный элемент.

Таким образом, очень важно иметь возможность определять место неисправности с точностью до сменного элемента (блока), так как в противном случае диагностика существенно затрудняется. В машинах, в которых все сигналы ошибок объединялись в один, ремонт усложнялся ввиду отсутствия информации о месте возникновения неисправности и, следовательно, о том, в каком направлении следовало предпринимать следующие шаги диагностики.

С этим обстоятельством связано и требование к системе контроля быстро обнаруживать ошибку с тем, чтобы она не смогла распространиться в машине и ее можно было локализовать в пределах малого количества логических схем. Лучше всего располагать схему обнаружения ошибки на сменном блоке, чтобы она сигнализировала о появлении ошибки в пределах этого блока. При этом на каждом из таких блоков может располагаться индикация для информации обслуживающего персонала о том, какой блок должен быть заменен.

## **Б. Основные виды ошибок и принцип подхода к ним**

Все ошибки, которые по тем или иным причинам возникают на машине или сказываются на ее работе, можно классифицировать по следующим основным видам:

1. Ошибки в программах.
2. Ошибочные действия оператора.
3. Ошибки в устройствах хранения и передачи информации.
4. Ошибки в оборудовании (ошибки в логическом оборудовании, ошибки в системе контроля, неисправности в системах питания и охлаждения).

Выявление ошибок в программах заключается в выявлении системой обнаружения нарушения формализованных действий со стороны программы, которые влекут за собой появление ошибок в вычислениях. К числу таких нарушений относятся, например, обращение к недействительным или запрещенным адресам, появление недействительных кодов операций и т. п., т. е. все то, что можно как-то формализовать и предусмотреть в системе обнаружения возможность проверки этих формальных требований: Очевидно такая защита способна выявить только элементарные ошибки в программе, ибо трудно создать достаточно простую систему обнаружения ошибок в логике решения задачи. Ошибки подобного вида легче обнаруживаются самими программистами или операторами, прогоняющими программы в соответствии с инструкциями, чем машиной.

Ошибочные действия оператора пока трудно поддаются прогнозированию. Оператор может поставить не ту ленту,

запустить не ту программу, перепутать колоду карт, не говоря уже о других более «мелких» ошибках — нажать не на ту кнопку, забыть включить тумблер, передать не туда управление, потерять в считывающем устройстве одну карту, которая затем попадает в чужую колоду, и т. п. Вся сложность вопроса заключается в том, что причиной ошибок оператора является не только и не столько невнимательность, сколько повышение утомляемости в работе и его внутреннее состояние.

Исследования последних лет наглядно показывают необходимость особого внимания к проблеме повышения надежности человеческого фактора в системах управления различной сложности и назначения. Эффективность человеко-машинных систем резко падает при снижении способности оператора справляться с возложенными на него обязанностями. На способность человека-оператора своевременно и точно выполнять свои функции на протяжении заданного времени влияет множество факторов, из которых едва ли не самым существенным является психофизиологические характеристики, определяющие его состояние. Поэтому возможность исключения ошибок со стороны оператора связана как с созданием оптимальных условий его работы, так и с формализацией действий оператора, позволяющей ввести критерии оценки этих действий. Однако определение того, какая часть деятельности оператора может быть формализована для выявления ошибок, остается пока нерешенной до конца проблемой.

Ошибки в данных, подлежащих записи в память и хранению, устраняются схемой исправления ошибок перед записью или путем восстановления информации в памяти после получения сигналов ошибки. С этой целью исходная информация определенное время сохраняется для возможной корректировки полученных данных, искаженных в результате появления ошибки. В некоторых машинах информация хранится с избыточными разрядами, облегчающими задачу ее корректировки. Существуют различные коды, которые используются в запоминающих устройствах вычислительных машин.

Ошибки при передачах информации по каналам связи аналогичны ошибкам в запоминающих устройствах. Эти ошибки исправляются в процессе передачи (с помощью специальных корректирующих кодов) или информация восстанавливается в памяти (обычно методом повторной передачи данных, принятых с ошибкой).

При появлении ошибок в логических схемах машины производится повторный пуск, если ошибка одиночная. Если ошибка многократная или устойчивая, производится ремонт или реконфигурация системы (исключение неисправного блока с сохранением дальнейшей работоспособности системы).

Если ошибки возникают в самих схемах контроля, то оператор сам должен выбрать дальнейший режим работы. Однако, если есть необходимость в продолжении вычислений, он должен помнить, что на этот период времени машина окажется без защиты.

Неисправности схем контроля могут быть двух типов: появление ложного сигнала ошибки и отсутствие сигнала при появлении ошибки в контролируемой схеме.

Неисправности первого типа обнаруживаются двумя способами. Первый из них заключается в использовании останова устройства при ошибке, после которого производится анализ состояния контролируемой схемы и делается вывод. Второй способ состоит в запуске специальной тестовой программы, диагностирующей сигналы схемы контроля при ее работе. (Для этого разрабатывается определенная последовательность сигналов, подаваемых на схему контроля, которая позволяет характеризовать ее работоспособность.)

Неисправности схем контроля второго типа представляют собой большую опасность, чем появление ложных сигналов ошибки. Поэтому схемы контроля или периодически проверяют с помощью тестовой программы или, если та или иная схема контроля не может быть охвачена такой проверкой, производят их дублирование — использование двух схем вместо одной. При выходе из строя одной из схем контроля сигналы ошибки будут выдаваться другой.

Неисправности в системах питания, охлаждения или механических устройствах машины могут вызвать появление ошибочных результатов подобно неисправностям в логических схемах. При этом машина должна быть остановлена, а возникшая неисправность устранена. Неисправности систем питания и охлаждения выявляются посредством датчиков и контрольных приборов. Неисправности в механических устройствах машины устанавливаются сложнее, поэтому основной гарантией их работоспособности является своевременное проведение профилактического ремонта и поддержание этих устройств в технически исправном состоянии.



## В. Контроль специальных схем

Кроме обычных логических схем, построенных на элементах типов НЕ, И, ИЛИ, в машине существуют специальные схемы, которые по характеру контроля можно разделить на три группы: 1) не требующие контроля, 2) поддающиеся контролю и 3) не поддающиеся контролю (контроль возможен только при дублировании).

К первой группе относятся неответственные цепи и элементы (усилители индикаторов и др.), неисправности которых не приводят к ошибкам в работе машины. Проверка их работоспособности обычно осуществляется в процессе проведения профилактик на машине.

Вторую группу составляют схемы различных преобразователей, например преобразования сигналов, воспроизводимых с ленты, дисков или памяти, преобразования аналоговых сигналов в дискретные и т. п.

Контроль подобных схем осуществляется проверкой на четность (или другим кодом), избыточный разряд которого поступает вместе с контролируемым словом. При возникновении однократной ошибки четность единиц, входящих в слово, нарушается, что фиксируется схемой контроля при приеме числа на регистр. (Подробнее вопросы построения кодов и их применения рассматриваются в следующих главах).

Схемы питания обычно контролируются с помощью специальных схем, чувствительных к уровню напряжения. При отклонении напряжения от заданных пределов схемы контроля выдают соответствующий сигнал, по которому производится автоматическое регулирование напряжения.

К схемам третьей группы относятся схемы одноразового действия, например датчики температуры. Если нет возможности проконтролировать работу таких схем косвенным образом (например, путем контроля правильности работы схемы, в которую они входят), то применяется их дублирование.

Дублирование схем является вынужденным средством, применяемым в случае, если их контроль затруднен. Следовательно, существует вероятность того, что в них может возникнуть неисправность, относящаяся к категории не обнаруживаемых. Однако если эта вероятность мала по сравнению с общим числом ошибок, ожидаемых в машине, то этим можно пренебречь, поскольку создание для таких схем специальных схем контроля явится неоправданным

удорожанием системы. Поэтому проверка работы подобных схем обычно осуществляется во время проведения профилактики машины.

### **Г. Останов из-за ошибки**

Если в машине возникает ошибка, свидетельствующая о неисправности, машина останавливается. Останов в этом случае предусматривается в двух целях: сохранить информацию для возможности повторения программы, обрабатываемой в данный момент времени, и помочь обслуживающему персоналу выявить и устранить причину неисправности.

Схема контроля, вызывающая останов, выдает сигнал ошибки, который фиксируется на специальном регистре. При этом машину нельзя снова пустить до тех пор, пока ошибка не будет устранена или заблокирована с помощью тумблера, расположенного на пульте машины. Блокировка ошибки предусматривается с целью возможности использования машины при наличии ошибки (например, при ремонте).

Если останов происходит в середине передачи данных или в момент выполнения операции, на регистрах машины останется искаженная информация. Чтобы останов позволил произвести повторение, первоначальный анализ или выполнить другие действия, предусмотренные в случае появления ошибки, на регистрах машины необходимо иметь информацию, достаточно определенным образом характеризующую состояние машины в момент останова. С этой целью останов должен конструироваться таким образом, чтобы новый цикл не начинался при появлении ошибки, но при необходимости мог быть продолжен и закончен, если сигнал ошибки возникает после его начала. Это особенно важно для асинхронных машин (машин с переменным тактом), поскольку многие асинхронные схемы нельзя остановить в любой момент, в результате чего они приводят машину к состоянию, которое трудно расшифровать. Поэтому механизм останова в таких машинах является одной из важнейших частей конструкции асинхронных схем.

### **Д. Работа на неисправной машине**

Иногда возникают ситуации, когда по тем или иным соображениям желательно продолжить использование машины, на которой возникла неисправность. Безусловно,

многое здесь зависит от характера неисправности, но в целом ряде случаев ее появление еще не означает, что машина не может выполнять процесс вычисления. Она может быть использована для выполнения, скажем, каких-то конкретных программ, на работу которых данная неисправность не оказывает воздействия. Для этого следует отключить сигналы некоторых типов ошибок и начать обработку программ. Наиболее просто это можно осуществить в системах, где сигналы ошибок поступают в центральный узел управления, который реагирует на них и отвечает соответствующими действиями.

Следует, однако, заметить, что работу на неисправной машине нельзя считать правилом, поскольку неисправности могут усугубляться и тем самым усложнить последующий ремонт машины. Возможны случаи, когда одна неисправность может способствовать возникновению другой неисправности, что также потребует дополнительного ремонта. Кроме того, в некоторых машинах схемы контроля контролируют работу не одного, а большего числа сменных блоков, поэтому не исключена возможность появления непредвиденных неисправностей, которые останутся незамеченными машиной вследствие того, что некоторые сигналы ошибок отключены.

### **Е. Реконфигурация как средство достижения высокой надежности**

Существуют машины, которые в целях обеспечения высокой надежности наделены способностью исключать неисправный элемент на время его ремонта и после проведения соответствующей реконфигурации, т. е. перераспределения имеющихся ресурсов, продолжать работу почти с прежней производительностью. Если машина может работать с несколькими исключенными элементами, то, следовательно, ее надежность оказывается намного выше надежности отдельных ее элементов.

В вычислительных системах, обладающих большими ресурсами, механизм реконфигурации еще более развит, так что в таких системах могут исключаться из работы целые устройства без заметного снижения общей производительности.

Требования высокой надежности и наибольшей готовности при минимальной конфигурации вычислительной системы (минимальном наборе функциональных блоков,

т. е. устройств) и максимальной рабочей нагрузке побудили к созданию мультипроцессорных систем (многопроцессорных машин) с соответствующей организацией и средствами управления прохождением машинных программ (программ пользователей и программ математического обеспечения). Такие системы получили названия систем, устойчивых к ошибкам.

Существуют два основных направления в создании таких систем:

- системы с реконfigurацией;
- дуплексные системы.

В системах с реконfigurацией, получивших наибольшее распространение, предусматривается применение нескольких идентичных устройств каждого типа, организованных и взаимосвязанных таким образом, что каждый из них может выполнять различные программы. В случае выхода из строя одного функционального блока исходная система продолжает работать, несколько снижая свою эффективность и производительность вследствие реконfigurации (перестройки) и перераспределения нагрузки (так называемый принцип постепенной деградации системы). Благодаря возможности реконfigurации система может выполнять наиболее важные задачи, несмотря на уменьшение ее суммарной производительности. В тех случаях, когда эту производительность необходимо сохранить, в системе предусматриваются запасные функциональные блоки, которые используются при выходе из строя основных блоков. Разумеется, в схемах каждого устройства и блоке программного обеспечения предусматривается полный набор контрольных средств обнаружения и исправления ошибки и ликвидации ее последствий.

Другой подход (дуплексные системы) предполагает параллельную работу по обработке данных для одной и той же задачи двух или большего числа функциональных блоков, а в некоторых случаях даже нескольких целых систем с постоянным сравнением получаемых результатов. Любые расхождения в ответах, полученных отдельными устройствами, являются первым признаком сбоя, после чего производится полная проверка устройств путем автоматического прогона тестов и неисправное устройство выводится из работы для его последующего восстановления. (При этом на пульт оператора, следящего за работой системы, выдаются соответствующие сведения). Общая производительность в этом случае не уменьшается, так как дублирующее устройство берет на себя всю нагрузку. Правда, возможности обнаружения ошибки при этом уменьшаются, но в таком режиме система продолжает свою работу лишь до тех пор, пока неисправное устройство не будет вновь введено в строй.

На первый взгляд может показаться, что подобная организация мультипроцессорной системы, обладающей высокой устойчивостью к аппаратным сбоям в отдельных устройствах и способностью к реконfigurации с целью сохранения непрерывности в работе, не требует возложения функций по исправлению ошибок в обрабатываемой инфор-

мации на операционную систему мультипроцессора. Однако это не так. Все потенциальные возможности конфигурации системы оказались бы бесполезными, не будь в распоряжении программного обеспечения специально предусмотренных возможностей для исправления возникающих ошибок.

Так, все системные таблицы и таблицы управления задачами формируются в двойном количестве, причем дублированные экземпляры размещаются в разных банках памяти. Операционная система организуется таким образом, чтобы ее можно было эффективно использовать как в системе с максимальной конфигурацией, так и в любой возможной подсистеме. Условия работы определяются автоматически по индикаторному разряду в управляющем слове памяти, который периодически проверяется операционной системой для определения текущего состояния системы в целом. Для обнаружения и исправления ошибок контролируются все важнейшие точки системы, что осуществляется применением контроля по четности при обращении к памяти и при передаче всех данных во все устройства системы, включая контроль устройств ввода-вывода путем чтения после записи или перфорации и т. д. Кроме того, производится периодическая проверка работоспособности процессоров с помощью тестов, причем такой профилактики подвергаются процессоры, не занятые в данный момент рабочими программами. Это позволяет обнаружить ошибки до того, как они проявятся в процессе обработки информации, к тому же такая проверка не снижает общей эффективности работы системы.

## ГЛАВА ЧЕТВЕРТАЯ

### ОСНОВЫ ПОСТРОЕНИЯ СХЕМНОГО КОНТРОЛЯ

#### 4-1. КОДЫ ДЛЯ ОБНАРУЖЕНИЯ И ИСПРАВЛЕНИЯ ОШИБОК

##### А. Общие сведения о кодах

Коды обнаружения или обнаружения и исправления ошибок применяются в вычислительных машинах для контроля правильности передач информации между устройствами и внутри устройств машины, а также для контроля всего процесса обработки информации на машине.

Под кодом понимают совокупность символов, с помощью которых отображается информация. Поскольку в машинах используются в основном двоичные коды, в их составе содержатся лишь два символа: 0 и 1. Каждому слову или знаку соответствует своя строго определенная комбинация нулей и единиц. Если все слова содержат одинаковое количество разрядов, код называется *равномерным*. В неравномерных кодах количество разрядов в словах может быть

различным. В вычислительных машинах обычно применяются равномерные коды.

С помощью  $n$  двоичных знаков можно представить  $2^n$  различных информационных комбинаций (различных слов). Код, в котором все разряды слова используются для представления информации, называется *простым*. При операции с простым кодом всякая возникшая ошибка, выражающаяся в замене 0 на 1 или 1 на 0, превращает данную информационную комбинацию в другую. Обнаружить ошибку, используя лишь одну вновь возникшую комбинацию, нельзя, так как машина должна для этого произвести какое-то сравнение, т. е. иметь дополнительную (избыточную) информацию. Поэтому при работе с простыми кодами в качестве дополнительной информации используется повторная передача слова, что обеспечивает возможность сравнения и, следовательно, обнаружения ошибки.

Вместо таких систематических повторных передач, на которые тратится дополнительное время и которые не всегда позволяют обнаружить устойчивую ошибку, в машинах применяются не простые, а *избыточные* коды. Это коды, в которых для представления информации используется лишь часть всех возможных знаковых комбинаций. Другая часть комбинаций в избыточных кодах является запрещенной. Появление запрещенных комбинаций расценивается как ошибка, что и фиксируется схемами контроля машины. Например, в простом четырехзначном коде все 16 комбинаций нулей и единиц используются для изображения чисел от 0 до 15. Любая ошибка даст новую, но опять-таки разрешенную комбинацию, т. е. одно из чисел от 0 до 15. В результате ошибка останется необнаруженной.

Если наложить запрет на часть комбинаций, например восемь, а остальные восемь использовать для изображения чисел от 0 до 7, то любая ошибка в знаке приведет к появлению запрещенной комбинации, которая и будет обнаружена машиной (табл. 4-1).

Т а б л и ц а 4-1

Десятичное число	Двоичный код	Десятичное число	Двоичный код
0	0000	4	1001
1	0011	5	1010
2	0101	6	1100
3	0110	7	1111

Поскольку для изображения чисел от 0 до 7 простым кодом достаточно иметь три знака, то представленный в таблице код явится избыточным, так как для изображения тех же восьми чисел ему требуется четыре знака. Однако проигрыш в избыточности дает выигрыш в возможности обнаружения ошибки.

В машинах применяются коды, которые называются *систематическими*. К систематическим относятся коды, состоящие из  $n$  двоичных символов,  $m$  из которых используются для представления информации (информационные символы), а  $k = n - m$  — для обнаружения и исправления ошибок (проверочные символы). Избыточность  $R_n$  в таком коде равна отношению полного числа двоичных символов к минимальному числу символов, необходимых для передачи той же информации, т. е.

$$R_n = \frac{n}{m}.$$

Величина  $R_n$  определяет эффективность кода или степень уменьшения информационной емкости канала.

Для характеристики кодов часто пользуются понятием относительной избыточности  $R$ , которая выражается как

$$R = \frac{k}{m}.$$

Все избыточные коды подразделяются на коды, которые обнаруживают ошибки, и коды, которые не только обнаруживают, но и исправляют (корректируют) ошибки. Последние называются *корректирующими* кодами.

Использование кодов с исправлением ошибок очень усложняет аппаратуру и снижает пропускную способность канала, поэтому применение таких кодов может быть оправдано лишь в особо ответственных случаях, например:

работа системы без надзора в течение длительного времени;

в сложных системах, в которых ошибка может вывести из строя всю установку (например, машины, управляющие работой крупного технологического объекта);

для передачи сигналов при наличии сильных помех, которые не могут быть подавлены.

Следует отметить, что вообще не существует кодов, которые обнаруживали бы все ошибки. Однако в зависимости от вероятности появления тех или иных ошибок, можно соответствующим выбором кода сделать эту вероят-

ность сколь угодно малой. В целом, учитывая эти соображения, можно сформулировать основные требования, предъявляемые к коду:

1. Код должен обнаруживать наиболее часто встречающиеся виды ошибок.

2. Заданная степень обнаружения ошибки должна достигаться присоединением к сообщению минимального количества избыточной информации (минимального числа контрольных разрядов).

3. Процедуры кодирования и декодирования должны быть по возможности простыми и быстрыми.

*Корректирующая способность кода.* Способность кода обнаруживать или исправлять ошибки характеризуется минимальным значением кодового расстояния  $d$ . Кодовым расстоянием между двумя словами (комбинациями) называется число разрядов, в которых соответствующие символы не совпадают. Если длина слова  $n$ , то кодовое расстояние может принимать значение от 1 до  $n$ . Если код имеет хотя бы две комбинации, отличающиеся друг от друга в одном разряде, то минимальное расстояние данного кода равно единице. Другими словами, минимальное кодовое расстояние — это минимальное количество двоичных разрядов, которое достаточно изменить для того, чтобы одну информационную комбинацию превратить в другую.

Простой код имеет минимальное расстояние  $d = 1$ . Для избыточных кодов  $d_{min} > 1$ . Если, например,  $d = 2$ , то любые две комбинации данного кода отличаются не менее чем в двух разрядах. Любая одиночная ошибка не сможет превратить данную комбинацию в какую-либо разрешенную, а приведет к появлению запрещенной. Следовательно, такая ошибка будет обнаружена. Таким образом, для обнаружения одиночной ошибки (искажения в одном из разрядов числа) достаточно иметь код с  $d = 2$ .

Чтобы можно было не только обнаружить, но и исправить одиночную ошибку, необходимо иметь код с  $d \geq 3$ . В этом случае любая одиночная ошибка создает запрещенную комбинацию, отличающуюся от правильной в одном разряде, а от любой другой разрешенной — в двух разрядах. Этого достаточно, чтобы определить искаженный разряд и исправить его. Исправление заключается в поочередном изменении значений каждого разряда слова (0 на 1 и 1 на 0) с проверкой на запрещенность. Если после замены символа комбинация остается запрещенной, символ в данном разряде вновь восстанавливается. Очевидно, что



комбинация окажется разрешенной в единственном случае — когда будет инвертирован ошибочный разряд, поскольку одиночная ошибка изменила символ только в одной позиции.

Рассуждая аналогичным образом, можно показать, что для обнаружения двойных ошибок необходим код с  $d = 3$ , а для их исправления — код с  $d = 5$ .

Следовательно, чтобы избыточный код позволял обнаруживать групповые ошибки кратностью  $t$ , он должен иметь минимальное кодовое расстояние  $d \geq t + 1$ . В самом деле, одновременная ошибка в  $t$  разрядах слова создает новую комбинацию, отстоящую от истинной на расстояние  $t$ . Чтобы новая комбинация не совпала с какой-либо другой разрешенной комбинацией, расстояние между этими комбинациями (новой и любой соседней разрешенной) должно быть хотя бы на единицу больше, чем  $t$ .

Для исправления  $t$ -кратной ошибки необходимо, чтобы новая (искаженная) комбинация не только не совпадала с какой-либо разрешенной, но и оставалась ближе по кодовому расстоянию к истинной, чем к любой соседней разрешенной комбинации. От истинной комбинации новая комбинация должна отстоять не менее чем на  $t + 1$ . Следовательно, кодовое расстояние между истинной комбинацией и соседней разрешенной должно быть не меньше суммы этих значений:

$$d \geq 2t + 1.$$

## **Б. Контроль передачи информации**

### *Код с проверкой четности*

Код с проверкой четности является простейшим избыточным кодом. Он образуется добавлением к группе информационных разрядов одного избыточного (контрольного) разряда.

При формировании кода слова в его контрольный разряд записывается 0 или 1 таким образом, чтобы сумма единиц в слове, включая избыточный разряд, была четной или нечетной (в случае контроля 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, это воспринимается как признак ошибки.

Минимальное расстояние кода  $d = 2$ , поэтому код с проверкой четности обнаруживает все одиночные ошибки и все случаи нечетного числа ошибок (3, 5, 7 и т. д.). При одновременном возникновении двух или другого четного числа ошибок код с проверкой четности их не обнаруживает (четность при этом не нарушается).

Код с проверкой четности имеет небольшую избыточность и поэтому не требует больших затрат оборудования на реализацию. Он широко применяется в современных машинах для контроля передач информации как между устройствами, так и внутри самих устройств.

При последовательной передаче данных контроль по четности легко реализуется путем использования триггера со счетным входом, на который поступает передаваемая информация. После прохождения всех информационных символов он должен оказаться в состоянии 0, если число единиц было четным, или в состоянии 1, если число единиц нечетно. Состояние триггера после прохождения слова представляет собой значение контрольного разряда.

В некоторых случаях осуществляют кодирование и проверку слов на нечетность, что позволяет контролировать полное пропадание информации, поскольку кодовое слово, состоящее из нулей, будет относиться к запрещенным.

Следует отметить преимущества использования контроля нечетности при последовательной передаче данных по сравнению с контролем четности. Во-первых, в случае пропадания информации на вход триггера будут поступать одни нули. Однако при контроле четности триггер ошибки не обнаружит, поскольку останется в нулевом состоянии, как и при пересчете четного числа единиц. Во-вторых, поскольку при контроле по нечетности триггер должен после прохождения слова установиться в противоположное состояние, появляется возможность контролировать его исправность (путем проверки состояния триггера до и после приема слова).

При параллельной передаче информации обычно применяются логические схемы определения четности суммы единиц.

При небольшой избыточности код с проверкой четности обладает значительной контролирующей способностью, так как обеспечивает практически полную надежность обнаружения всех одиночных и нечетных групповых ошибок.

При учете надежности работы элементов вычислительных машин, а также малой вероятности возникновения групповых (и именно четных) ошибок этот код в настоящее время принят в машинах в качестве основного контролирующего кода.

### *Код Хэмминга*

К одним из распространенных кодов, позволяющих обнаруживать и исправлять одиночные ошибки в передаваемых числах, относится систематический код с проверкой на четность, известный под названием кода Хэмминга. В этом коде из  $n$  позиций передаваемого слова  $m$  позиций используются для передачи информации, а  $k$  — в качестве контрольных (проверочных). Все  $m$  информационных разрядов разбиваются на контрольные группы. Каждый контрольный разряд закрепляют за определенной группой. Перед передачей в контрольные разряды записываются символы 0 или 1, являющиеся знаками четности соответствующих групп.

После передачи числа в приемном устройстве производится  $k$  проверок на четность всех  $k$  контрольных групп. После каждой проверки в специальный регистр ошибок записывается 0, если результат проверки свидетельствует об отсутствии ошибки на проверяемых позициях данной группы, и 1, если результат свидетельствует о наличии ошибки. Таким образом, каждая проверка заканчивается записью в соответствующий разряд регистра 0 или 1. Полученная последовательность нулей и единиц образует двоичное число, называемое проверочным, которое должно указывать номер позиции (разряда слова) с искаженным символом.

Учитывая необходимость записи в двоичном коде номера любого из  $m + k$  разрядов принятой комбинации, а также то, что отсутствию ошибки в этой комбинации будет соответствовать число, составленное только из нулей, проверочное число должно описывать  $m + k + 1$  событий. Следовательно, для  $k$  (количества необходимых для этих целей контрольных разрядов) будет справедливо неравенство

$$2^k \geq m + n + 1.$$

Поскольку  $n = m + k$ , можно записать:

$$2^m \leq \frac{2^n}{n+1}.$$

В соответствии с этим неравенством составлена табл. 4.2, которая дает возможность определять минимальное значение  $n$  для данного  $m$ , т. е. соответствующее количество контрольных разрядов  $k$ .

Таблица 4-2

$m$	1	2	3	4	5	6	7	8	9	10	11	12
$n$	3	5	6	7	9	10	11	12	13	14	15	17
$k$	2	3	3	3	4	4	4	4	4	4	4	5

Способ разбивки информационных разрядов на контрольные группы и определение номеров позиций контрольных разрядов следует непосредственно из структуры натурального ряда чисел, представленного в двоичной форме.

Рассмотрим это на примере 13-разрядного двоичного числа ( $n = 13$ ).

Из табл. 4-2 видно, что для формирования корректирующего кода 13-разрядное число должно содержать девять информационных и четыре контрольных разряда. Следовательно, по числу контрольных разрядов вся комбинация должна состоять из четырех контрольных групп.

Разбивка числа на контрольные группы производится следующим образом (табл. 4-3).

Таблица 4-3

Номер разряда числа $n$		Номер разряда числа $n$	
в десятичной системе счисления	в двоичной системе счисления	в десятичной системе счисления	в двоичной системе счисления
0	0000	7	0111
1	0001	8	1000
2	0010	9	1001
3	0011	10	1010
4	0100	11	1011
5	0101	12	1100
6	0110	13	1101

Первая группа (первая проверка) охватывает все позиции кода, номер которых в двоичной системе счисления имеет единицу в первом разряде (1, 3, 5, 7, 9, 11, 13). Вторая группа (вторая проверка) включает позиции, номер которых имеет единицу во втором разряде (2, 3, 6, 7, 10) и т. д.

Разряды 1, 2, 4, 8, каждый из которых принадлежит только одной контрольной группе, используются в качестве контрольных, а остальные разряды, одновременно принадлежащие разным группам, — в качестве информационных.

Таким образом, при контроле правильности приема 13-разрядного числа производятся четыре проверки на четность. Первая проверка охватывает позиции первой группы (с записью результата проверки в первый разряд), вторая — позиции второй группы (результат записывается во второй разряд) и т. д.

В качестве иллюстрации рассмотрим принцип определения ошибки при передаче числа в коде Хэмминга на простом примере.

Пусть передается число 1011, преобразованное в 7-разрядный корректирующий код (см. табл. 4-3).

В табл. 4-4 в первой строке показан код переданной комбинации. Контрольные разряды (выделены шрифтом) автоматически записываются при формировании передаваемого кода в соответствии с рассмотренным правилом проверки на четность: знак первого разряда (1) записан как результат суммирования (по модулю 2) знаков 3, 5 и 7-й позиций кода; знак второго разряда (0) — как результат суммирования знаков второй группы (3, 6 и 7-й позиции); знак четвертого разряда (0) — как результат суммирования знаков третьей группы (5, 6 и 7-й позиции).

Т а б л и ц а 4-4

Число	Номер разряда числа						
	7	6	5	4	3	2	1
Передаваемое	1	0	1	0	1	0	1
Принятое	0	0	1	0	1	0	1

Предположим теперь, что при передаче числа был искажен один из символов. Определим номер разряда искаженного символа.

Произведем три проверки на четность в соответствии с установленным правилом. (При проверках учитывается и контрольный разряд.)

Первая проверка (1, 3, 5, 7) дает 1; вторая проверка (2, 3, 6, 7) дает 1; третья проверка (4, 5, 6, 7) дает 1. Получаем проверочное число 111 (семь), которое указывает на ошибку в седьмом разряде принятой комбинации. Следовательно, для исправления ошибки символ в седьмом разряде необходимо инвертировать. Процесс исправления ошибки производится на машине автоматически.

Код Хэмминга легко может быть расширен до  $d = 4$  (исправление одиночной и обнаружение двойной ошибки). С этой целью к  $k$  контрольным разрядам добавляется еще

один контрольный разряд, который используется для контроля по четности всего кодового слова, включая и контрольные разряды, что позволяет дополнительно осуществлять общую проверку всего слова на четность.

### *Циклические коды*

Циклические коды получили достаточно широкое распространение благодаря способности с высокой эффективностью обнаруживать не только одиночные, но и групповые ошибки при незначительном увеличении избыточности кодирования. Применяются циклические коды в системах с последовательной передачей двоичных разрядов, составляющих слово, в частности при записи и чтении информации в некоторых типах внешних запоминающих устройств на магнитных лентах и дисках.

Циклические коды обязаны своему названию благодаря особенности, которая заключается в том, что если  $n$ -значная информационная комбинация  $a_0a_1a_2 \dots a_n$  принадлежит к данному коду, то и комбинация  $a_na_0a_1 \dots a_{n-1}$ , в которой произведена циклическая перестановка символов, также принадлежит к этому коду.

Такая особенность циклического кода позволяет сравнительно просто организовать кодирование и декодирование информации, что достигается использованием сдвигающего регистра с обратной связью, обладающего циклическими свойствами.

Принцип построения циклического кода основан на следующем. Любая двоичная кодовая комбинация может быть представлена в виде полинома ( $n - 1$ )-й степени некоторой переменной  $x$ , коэффициентами которого являются двоичные символы соответствующих разрядов числа. Например, число 1011001 (младшие разряды справа) представляется полиномом шестой степени:

$$\begin{aligned} 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0 &= \\ &= x^6 + x^4 + x^3 + 1. \end{aligned}$$

Над такими полиномами можно производить все математические действия в соответствии с известными законами алгебры за исключением сложения, которое, в отличие от обычной алгебры, осуществляется по модулю 2 (поразрядное сложение без учета переносов).

Циклический код  $n$ -значного числа строится путем добавления к  $m$  двоичным информационным разрядам  $k$  конт-

рольных разрядов. Последние являются младшими разрядами числа и поэтому при передаче идут последними, вслед за информационными разрядами.

Кодирование заключается в составлении на основе информационной двоичной комбинации  $G(x)$ , подлежащей кодированию, такого многочлена  $F(x)$ , который должен делиться без остатка на заранее выбранный порождающий полином  $P(x)$  степени  $k$ , с помощью которого и происходит образование циклического кода. Следовательно, если (например, при приеме числа) деление многочлена  $F(x)$  на многочлен  $P(x)$  даст остаток  $R(x)$ , это будет означать появление ошибки в переданной комбинации.

Наиболее простым способом получения  $F(x)$  было бы перемножение многочленов  $G(x)$  и  $P(x)$ . Однако этот способ неприемлем, так как в этом случае полученный код окажется несистематическим — в нем не будут выделены контрольные разряды. Поэтому кодовая комбинация  $F(x)$  строится следующим образом. Информационный многочлен  $G(x)$ , подлежащий кодированию, умножается на  $x^k$ . Полученный многочлен  $G(x)x^k$  делится на порождающий полином  $P(x)$  для определения остатка  $R(x)$ , после чего остаток складывается (по модулю 2) с  $G(x)x^k$ . Таким образом, получаем:

$$F(x) = G(x)x^k \oplus R(x).$$

Принимая во внимание, что при делении многочлена  $G(x)x^k$  на порождающий полином  $P(x)$  помимо остатка  $R(x)$  получается частное, можно записать:

$$G(x)x^k = Q(x)P(x) \oplus R(x),$$

где  $Q(x)$  — частное от деления (в работе не используется).

Поскольку суммирование производится по модулю 2 (сложение и вычитание производятся одинаково), то, перенеся  $R(x)$  в левую часть равенства, получим:

$$F(x) = G(x)x^k \oplus R(x) = Q(x)P(x),$$

откуда следует, что кодовый многочлен  $F(x)$  кратен значению  $P(x)$  и при отсутствии ошибки должен делиться на  $P(x)$  без остатка. Кроме того,  $G(x)x^k$  имеет нули в  $k$  разрядах (умножение производилось путем сдвига слова), следовательно, старшие разряды  $m$  кодовой комбинации  $F(x)$  представляют собой разряды двоичной комбинации  $G(x)$ , т. е. являются информационными. Младшие раз-

ряды  $k$  кодовой комбинации  $F(x)$  отводятся под остаток  $R(x)$  и представляют собой контрольные разряды.

В процессе передачи комбинации  $F(x)$  могут возникнуть помехи, которые исказят передаваемую информацию. Неправильно принятую комбинацию  $H(x)$  можно в этом случае представить следующим образом:

$$H(x) = F(x) \oplus E(x),$$

где  $E(x)$  — многочлен ошибок, имеющий ненулевой член в каждом искаженном разряде [при правильно принятой информации  $E(x) = 0$ ].

Основным признаком появления ошибки, как уже говорилось, служит появление остатка при делении  $H(x)$  на  $P(x)$ . Однако возможен случай, когда остаток не появится даже при наличии ошибки — это если  $E(x)$  без остатка разделится на  $P(x)$ . Следовательно, для обнаружения ошибки необходимо выполнить еще одно условие:  $E(x)$  не должен делиться на  $P(x)$  без остатка. Этому добиваются за счет определенного выбора числа членов и степени старшего разряда порождающего полинома  $P(x)$ .

В качестве примера в табл. 4-5 приведен вид многочлена  $P(x)$  и количество контрольных разрядов для данной кодовой комбинации, при которых циклический код позволяет обнаруживать все одиночные и двойные ошибки.

Проследим теперь процесс преобразования числа в циклический код и декодирования на конкретном примере.

Т а б л и ц а 4-5

Порождающий полином $P(x)$	Число разрядов		
	$n \leq$	$m$	$k$
$x^3 + x + 1$	7	4	3
$x^4 + x + 1$	15	11	4
$x^5 + x^2 + 1$	31	26	5

Пусть дано число 11010. Искомый код образуется с помощью порождающего полинома  $P(x)$  степени  $k$ . Для нашего случая  $P(x) = x^3 + x + 1$  ( $n \leq 7$ ), что соответствует коду 1011.

Следующим этапом является умножение информационной комбинации, подлежащей кодированию,  $G(x)$  на  $x^k$  (в нашем случае  $k = 3$ ), что соответствует сдвигу кодируемого числа на  $k$  разрядов влево (в сторону старших разрядов). Полученный после сдвига многочлен  $G(x)x^k$  делится на  $P(x)$  для определения остатка, разряды которого и представляют собой контрольные знаки. Так как необходимо найти только остаток (частное не используется), на практике вместо обычного деления производится вычитание по модулю 2 (или сложение, что то же



самое) делителя из делимого и получающихся разностей до тех пор, пока разность не будет иметь более низкую степень, чем делитель. Эта последняя разность и есть искомый остаток, который необходимо дописать в информационную комбинацию. Таким образом, после сдвига числа приступаем к последовательному вычитанию из него кода порождающего полинома 1011:

$$\begin{array}{r}
 \oplus 11010000 \\
 \underline{1011} \\
 1100 \\
 \oplus 1011 \\
 \underline{1110} \\
 \oplus 1011 \\
 \underline{1010} \\
 \oplus 1011 \\
 \underline{010}
 \end{array}$$

Полученный остаток 010 дописывается к кодируемому числу 11010, что и дает искомый код: 11010010.

Проверка правильности кодирования (декодирование при приеме числа) состоит в выполнении деления закодированного числа на порождающий полином (1011). Как и при кодировании, здесь производится последовательное вычитание по модулю 2 делителя из делимого. При отсутствии ошибки деление будет выполнено без остатка:

$$\begin{array}{r}
 \oplus 11010010 \\
 \underline{1011} \\
 1100 \\
 \oplus 1011 \\
 \underline{1110} \\
 \oplus 1011 \\
 \underline{1011} \\
 \oplus 1011 \\
 \underline{0000}
 \end{array}$$

При применении циклического кода в качестве корректирующего для исправления одиночных ошибок определение номера искаженного разряда производится по виду полученного остатка (между остатком и многочленом ошибок существует вполне определенная связь: ошибке в любой из  $n$  позиций соответствует один из  $2^k - 1$  ненулевых остатков). Анализ остатка и исправление искаженного разряда производятся автоматически.

Следует отметить, что циклический код с  $d = 3$  считается циклическим вариантом кода Хэмминга, в котором в отличие от рассмотренной структуры последнего все контрольные разряды размещаются в конце информационной комбинации.

## Равновесные коды

Равновесный код применяется при передаче данных внутри устройств машины и по каналам связи. Особенностью равновесного кода является то, что в нем каждая информационная комбинация содержит фиксированное число единиц и нулей. Отличаются комбинации только позициями единиц.

Существует много видов равновесных кодов, но наибольшее практическое применение получил код «2 из 5», т. е. две единицы из пяти знаков (табл. 4-6). Этот код получается путем добавления пятого разряда к тетраде, изображающей десятичную цифру. Код имеет высокую корректирующую способность при обнаружении асимметричных ошибок,

Т а б л и ц а 4-6

Кодируемое число	Код «2 из 5»	Кодируемое число	Код «2 из 5»
0	01100	5	00110
1	11000	6	10001
2	10100	7	01001
3	10010	8	00101
4	01010	9	00011

т. е. когда происходит инверсия только единиц или только нулей. Ошибка не сможет быть обнаружена, если количество переходов нулей в единицу будет равно количеству переходов единиц в нули. Другими словами, признаком ошибки здесь является изменение количества единиц в комбинации.

Избыточность равновесного кода «2 из 5» выше, чем у обычного двоично-десятичного кода. Добавление пятого разряда увеличивает число возможных комбинаций ( $2^5 = 32$ ), однако для изображения цифр используется только 10, которые выбираются так, чтобы минимальное кодовое расстояние было равно 2.

Поскольку для отображения информации используется меньше половины всех возможных комбинаций, корректирующая способность рассматриваемого кода выше, чем у обычного кода с проверкой на четность.

Код «2 из 5» обнаруживает одиночные и групповые асимметричные ошибки. Схемная реализация кода рассматривается в следующей главе.

В некоторых машинах были попытки применить равновесный код «2 из 7». Однако резкое увеличение избыточности не повышает корректирующую способность этого кода, так как его минимальное кодовое расстояние остается тем же, т. е. равным 2. В то же время увеличение количества знаков в коде приводит к увеличению вероятности появления необнаруживаемых симметричных ошибок.

## **В. Контроль арифметических и логических операций**

Все рассмотренные выше коды использовались для контроля передачи информации. Особенностью подобного вида контроля является то, что с его помощью решается сравнительно несложная задача — убедиться в неизменности передаваемой информационной комбинации или восстановить эту информацию, если в ней произошли искажения. Совсем другие требования возникают при контроле обрабатываемой информации, которая не остается постоянной, а все время изменяется в процессе выполнения тех или иных операций. Следовательно, в этом случае необходимо обеспечить контроль правильности ее преобразования, т. е. правильности выполнения этих операций. И если возникшая ошибка при передаче информации искажает одно число или отдельные числа, не связанные друг с другом, то та же ошибка при расчетах начинает распространяться в вычислительном процессе, поскольку исходные данные одной операции являются результатом предшествующих операций.

Из множества разработанных методов контроля арифметических операций наибольшее распространение получил *контроль по модулю*, который называют также контролем по остаткам или наименьшим вычетам. Суть организации такого контроля заключается в том, что каждому числу, участвующему в операции, ставится в соответствие контрольный код, который представляет собой остаток от деления контролируемого числа на некоторое заранее заданное целое число  $q$ , называемое модулем. Использование остатка в качестве контрольного кода возможно по той причине, что любое число  $A$  сравнимо с остатком, полученным в результате деления этого контролируемого числа на модуль  $q$ .

При выполнении операции над числами та же операция выполняется над их контрольными кодами, после чего контрольный код результата основной операции сравнивается с результатом аналогичной операции над контрольными

кодами исходных чисел. Это вытекает из теории чисел, которая говорит, что для произвольных целых чисел  $A$  и  $B$  справедливо равенство (в случае операции сложения):

$$R(A + B) = R[R(A) + R(B)], \quad (4-1)$$

где  $R(X)$  означает остаток числа  $X$  по модулю  $q$ .

Если сравнение указанных результатов дает совпадение, операция считается выполненной правильно, при несовпадении — фиксируется ошибка.

Представим каждое из чисел  $A$  и  $B$  в виде

$$A = q\alpha + \beta; \quad B = q\gamma + \delta,$$

где  $\alpha$  и  $\gamma$  — частные от деления соответственно  $A$  на  $q$  и  $B$  на  $q$  (целые числа), а  $\beta$  и  $\delta$  — остатки чисел  $A$  и  $B$  по модулю  $q$ , т. е.  $\beta = R(A)$ ,  $\delta = R(B)$ . Тогда сумма чисел

$$A + B = (q\alpha + \beta) + (q\gamma + \delta) = q(\alpha + \gamma) + (\beta + \delta).$$

Разделим левую и правую части равенства на  $q$ :

$$\frac{A + B}{q} = (\alpha + \gamma) + \frac{\beta + \delta}{q}.$$

Значение  $\alpha + \gamma$  представляет собой целую часть от деления  $A + B$  на  $q$ , т. е. частное. Следовательно, остаток от суммы  $A + B$

$$R(A + B) = R(\beta + \delta),$$

что совпадает с выражением (4-1).

Рассуждая подобным же образом, можно убедиться в справедливости рассмотренного положения и для других операций (умножение, деление).

Значение  $q$  выбирается с таким условием, чтобы: 1) любая одиночная ошибка приводила к нарушению условия сравнимости результатов по модулю  $q$ , 2) операция деления для определения остатка была заменена более простыми методами и осуществлялась по сравнительно несложным признакам делимости, 3) аппаратура, реализующая контроль по модулю, была возможно проще, а это возможно при меньших значениях  $q$ , когда контрольные коды имеют малое число разрядов.

Из первого условия следует, что в качестве основания нельзя выбирать числа 2, 4 и т. п., т. е. типа  $2^n$  ( $n$  — целое число), поскольку при этом одиночные ошибки в старших разрядах не нарушают сравнимости по модулю  $q$  и, следовательно, не могут быть обнаружены. Этому условию лучше всего удовлетворяют основания типа  $2^n + 1$ . Если принять во внимание, что вероятность появления двойных

ошибок крайне мала, эффективность контроля при разных основаниях типа  $2^n \pm 1$  будет отличаться незначительно. Следовательно, целесообразно остановиться на основании  $2^2 - 1 = 2^1 + 1 = 3$ , т. е. на контроле по модулю 3, позволяющем наиболее просто и экономично реализовать схему контроля.

Для нахождения остатка от деления двоичного числа на модуль 3 достаточно просуммировать цифры разрядов контролируемого числа по модулю 3 с учетом знаков четных и нечетных разрядов, что удовлетворяет второму условию.

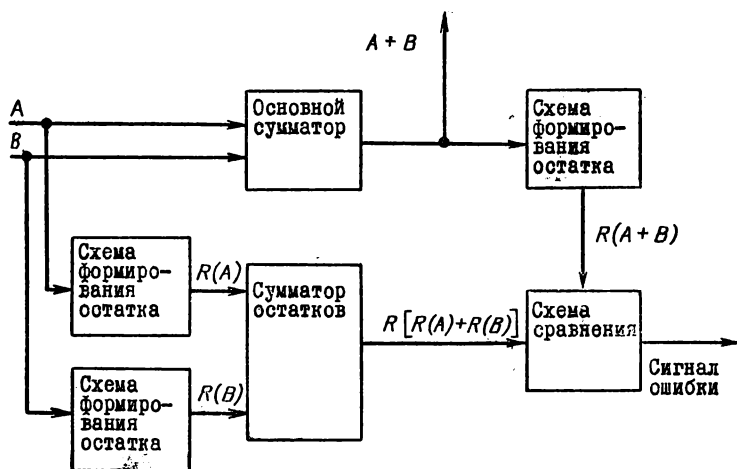


Рис. 4-1. Структурная схема сумматора с контролем по модулю 3.

Одиночная ошибка в одном из разрядов двоичного числа соответствует его изменению на  $\pm 2^i$ . Для возможности обнаружения ошибки необходимо, чтобы контрольные коды чисел  $A$  и  $A \pm 2^i$  не совпадали, т. е. нужно выполнить условие

$$R(A) \neq R(A \pm 2^i).$$

Поскольку  $2^i$  не делится на три без остатка, последнее условие всегда выполняется. Кроме одиночных ошибок, контроль по модулю 3 выявляет все двойные ошибки, для которых справедливо указанное условие.

На рис. 4-1 приведена структурная схема сумматора с контролем по модулю 3, общая логика работы которого ясна из рисунка (более подробно вопрос аппаратурной

реализации данного вида контроля рассматривается в следующей главе).

*Контроль логических операций*, в частности таких поразрядных операций, как логическое сложение (ИЛИ), логическое умножение (И) и исключающее ИЛИ (сложение по модулю 2 или операция неравнозначности), не имеет такой структуры, как контроль арифметических операций. Объясняется это тем, что, в отличие от арифметических, логические операции выполняются поразрядно и результат операции в каждом конкретном разряде определяется только состоянием соответствующих разрядов операндов, не связанных с другими разрядами числа. Следовательно, для большинства логических операций невозможно найти общие контрольные разряды, которые оказались бы совместимыми с данной операцией. Реализация же поразрядного схемного контроля в принципе возможна, но неэкономична, так как это потребует резкого увеличения контрольной аппаратуры.

По этой причине наиболее целесообразным является осуществление контроля логических операций путем их повторения, т. е. путем использования временной избыточности. При этом увеличение времени исполнения таких операций не является столь критичным, поскольку в целом выполнение логических операций занимает сравнительно небольшую часть общего времени выполнения программы.

Другим методом контроля является одновременное выполнение двух разных логических операций с последующим сравнением результатов по модулю 3. Для этих целей используются некоторые соотношения между результатами двух различных логических операций над операндами  $A$  и  $B$  и их алгебраической суммой  $A + B$ .

Примером таких соотношений могут служить выражения:

$$(A \wedge B) + (A \vee B) = A + B; \quad (4-2)$$

$$2(A \wedge B) + (A \oplus B) = A + B. \quad (4-3)$$

Подставив любые значения отдельных разрядов  $A_i$  и  $B_i$ , нетрудно убедиться, что оба слагаемых в левой части равенства (4-2) и (4-3) будут содержать вместе столько же единиц, сколько и в правой части обоих выражений, что подтверждает справедливость обоих равенств.

Для контроля выбирают то равенство, в которое входит контролируемая операция (например, для контроля операции неравнозначности выбирается второе соотношение). После этого одновременно выполняются обе логические операции, входящие в левую часть равенства. Затем определяются сумма контрольных кодов операндов  $A$  и  $B$  (по модулю 3) и сумма контрольных кодов результатов обеих

логических операций (по тому же модулю), после чего обе суммы контрольных кодов (операндов и результатов логических операций) сравниваются. Несравнение сумм вырабатывает сигнал ошибки.

### Г. Сравнительные характеристики избыточных кодов

Сравнение кодов между собой производится по двум основным характеристикам: относительной избыточности  $R$  и корректирующей способности (потенциальной эффективности)  $K$ .

Относительная избыточность кодов позволяет оценивать их с точки зрения объема контрольной аппаратуры, которая усложняется с ростом  $R$ . Правда, здесь не существует линейной зависимости и многое определяется алгоритмом реализации работы данного вида кода. Поэтому указанная оценка является приближенной.

Потенциальная эффективность кода характеризует способность данного кода обнаруживать (или обнаруживать и исправлять) ошибки, возникающие в информации при ее передаче или обработке, т. е. показывает степень уменьшения вероятности появления необнаруженных ошибок при использовании данного кода для контроля информации в машине.

Значение  $K$  также является ориентировочным, так как рассчитано из условия, что возникновение ошибок в отдельных разрядах информационной комбинации можно считать независимым случайным событием, имеющим равную вероятность (биномиальный закон распределения ошибок, справедливый при параллельной передаче и обработке информации), что не всегда справедливо.

Кроме того, сам по себе выбор системы контроля для машины зависит от многих факторов, о которых говорилось ранее. Поэтому не существует единых рецептов по применению методов и средств контроля, следовательно, и оценку применимости того или иного кода в тех или иных случаях следует воспринимать как ориентировочную.

Расчетные формулы для определения  $R$  и  $K$  в зависимости от длины слова  $n$  и вероятности возникновения ошибки в одном двоичном знаке  $p$  приведены в табл. 4-7.

Код с проверкой четности и код Хэмминга, очевидно, целесообразно применять в случае, если предполагается контролировать передачу информации параллельным кодом (внутренние передачи информации в машине), когда справедлив биномиальный закон распределения ошибок. При увеличении вероятности появления групповых ошибок эффективность, например, кода четности снижается несуществ-

венно, если контрольным знаком снабжается каждый блок информационной комбинации (в современных машинах это байты, в каждом из которых девятый разряд является контрольным).

Т а б л и ц а 4-7

Код и его минимальное расстояние	Относительная избыточность $R$	Потенциальная эффективность $K$	$R$	$K$
			при $n=36$	
Код с проверкой на четность ( $d=2$ )	$\frac{1}{n-1}$	$\frac{2}{(n-1)p}$	0,03	$\frac{0,057}{p}$
Код Хэмминга, в том числе его циклический вариант ( $d=3$ )	$\frac{\log_2(n+1)}{n-\log_2(n+1)}$	$\frac{6}{(n-1)(n-2)p^2}$	0,2	$\frac{0,0046}{p^2}$
Остаточный код по модулю 3 ( $d=2$ )	$\frac{2}{n-2}$	$\frac{4}{(n-1)p}$	0,06	$\frac{0,11}{p}$
Остаточный код по модулю 7 ( $d=2$ )	$\frac{3}{n-3}$	$\frac{12}{(n-1)p}$	0,09	$\frac{0,34}{p}$
Остаточный код по модулю 15 ( $d=2$ )	$\frac{4}{n-4}$	$\frac{16}{(n-1)p}$	0,125	$\frac{0,46}{p}$
Код «2 из 5» ( $d=2$ )	0,52	$\frac{0,83}{p}$	0,52	$\frac{0,83}{p}$

Однако код с проверкой четности только фиксирует ошибку, которая часто устраняется путем повторной передачи данных. Если повторная передача затруднена, то обычно используются коды, исправляющие ошибки.

При работе с такими запоминающими устройствами (ЗУ), как магнитная лента, диски, барабаны в некоторых системах применяется дублирование записей. При отсутствии такого дублирования и повышенных требованиях к достоверности в этом случае целесообразнее применять код Хэмминга, а там, где это возможно, — его циклический вариант, требующий более простой контрольной аппаратуры. Следует также отметить применение кода Хэмминга в оперативном запоминающем устройстве (ОЗУ) ЭВМ.

При последовательной передаче информации, когда вероятность появления групповых ошибок достаточно велика, наиболее подходящим кодом является циклический, рассчитанный на подобный режим работы.

Как видно из таблицы, относительная избыточность остаточных кодов при значениях модуля от 3 до 15 отличается почти в 2 раза. Соответственно будет изменяться и объем контрольной аппаратуры.



Вместе с тем, учитывая, что все приведенные остаточные коды имеют  $d = 2$  (только обнаруживают ошибки), а также то, что современные машины имеют достаточно надежные элементы и ошибки в них возникают довольно редко, в основном одиночные, обычно ограничиваются контролем операций по модулю 3.

В некоторых машинах на основе остаточных кодов организуется контроль не только арифметических операций, но также контроль передач и хранения информации в памяти машины. В этом случае отдается предпочтение контролю по модулю 7, эффективность которого при таких условиях оправдывает усложнение контрольной аппаратуры.

Код «2 из 5» обладает высокой эффективностью, но его избыточность очень большая. По этой причине этот код используется только для контроля передачи информации, хотя он может в принципе быть использован и для контроля работы арифметического устройства. Однако арифметическое устройство в этом случае получается очень сложным, поэтому равновесный код практически не находит в них применения.

#### **4.2. РЕЗЕРВИРОВАНИЕ КАК МЕТОД УСТРАНЕНИЯ ОШИБОК**

Резервирование является основным методом, широко применяемым при построении высоконадежных устройств и систем. Существует большое количество различных способов резервирования, и все они основаны на использовании большой избыточности. Высокая избыточность позволяет добиться того, что при резервировании практически полностью игнорируется специфика устройств, защищаемых от ошибок, и корректируются ошибки любых типов, в то время как корректирующие коды позволяют эффективно исправлять лишь одиночные ошибки, возникающие при передаче, обработке и хранении информации.

По способу включения резерва различают постоянное резервирование и резервирование замещением. В том и другом случае резервирование может быть общим (резервируется вся система в целом) и отдельным (система резервируется по отдельным узлам или элементам).

При постоянном резервировании резервные элементы (узлы, устройства) подсоединены к основным рабочим элементам постоянно и находятся в одинаковом с ними режиме. Например, параллельная работа двух машин, решающих одну и ту же задачу, параллельная работа однотипных устройств, узлов, элементов, выполняющих одну и ту же функцию, — все это относится к постоянному резервированию.

При постоянном резервировании система строится таким образом, что неисправность одного или нескольких элементов не нарушает их работу. Отказ происходит тогда,

когда кроме основных элементов выйдут из строя и резервные.

Это является большим достоинством постоянного резервирования, так как, можно сказать, резерв принимает на себя работу при выходе из строя основного элемента без всякого переключения, мгновенно. (Понятия основной и резервные элементы в данном случае являются условными, поскольку все эти элементы работают в одном режиме.)

Недостатком постоянного резервирования является большая аппаратная избыточность (в 2—4 раза).

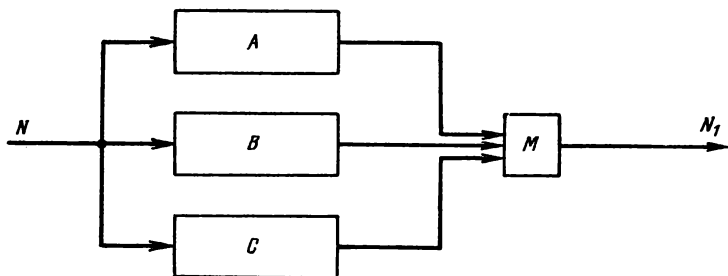


Рис. 4-2. Схема резервирования функциональных узлов с одним выходом.

Резервирование замещением заключается в том, что включение резервных элементов производится только после выхода из строя основных элементов, узлов или устройств. Такое резервирование (если оно автоматическое) связано с применением сложной системы контроля и переключающих схем, которые исключают неисправный элемент, узел или устройство и включают резерв.

Включение резерва может производиться и вручную. Однако в обоих случаях возникают перерывы в работе системы, что является недостатком этого способа резервирования.

К достоинствам резервирования замещением относится меньшая избыточность аппаратуры, так как здесь появляется возможность использовать одно резервное устройство для резервирования нескольких однотипных.

Постоянное резервирование осуществляется путем параллельного включения двух, трех или более узлов вместо одного, достаточного для выполнения заданной функции. При этом входы всех узлов объединяются, а выходы подаются на логический элемент, в качестве которого широко используется так называемый мажоритарный элемент (рис. 4-2). Мажоритарный элемент (строится на основе

элемента ИЛИ) осуществляет выбор входных сигналов по большинству, т. е. является элементом типа «2 из 3». На выходе  $M$  будет такой сигнал  $N_1$  (0 или 1), который подводится ко всем трем его входам одновременно либо к двум из них. При таком подключении узлов выход из строя любого из них не приведет к отказу системы всей схемы. Более того, схема защищает также от сбоев, поскольку появление случайной ошибки в одном из узлов не будет воспринято мажоритарным элементом и, следовательно, не появится на его выходе.

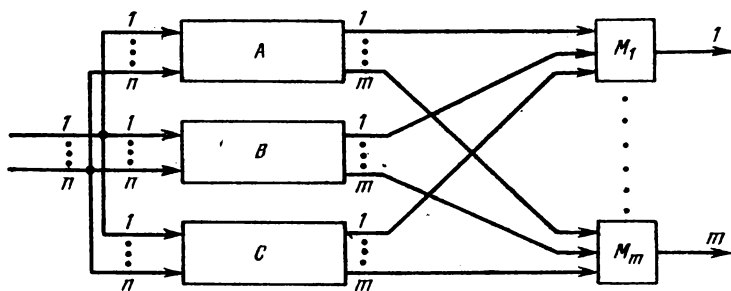


Рис. 4-3. Схема резервирования функциональных узлов с  $t$  выходами.

Подобным методом могут резервироваться и более сложные узлы, имеющие не по одному, а по нескольку входов и выходов. На рис. 4-3 показан пример такой схемы с  $n$  входами и  $t$  выходами, в которую, кроме трех функциональных узлов  $A$ ,  $B$  и  $C$ , входят  $t$  мажоритарных элементов «2 из 3». Принцип работы схемы не отличается от предыдущей.

## ГЛАВА ПЯТАЯ

### МЕТОДЫ АППАРАТУРНОГО КОНТРОЛЯ

#### И ИХ РЕАЛИЗАЦИЯ

##### 5-1. КОНТРОЛЬ ЧЕТНОСТИ

Идея организации контроля по четности (см. § 4-1) заключается в том, что к  $n$  информационным разрядам добавляется  $(n + 1)$ -й контрольный разряд, значение которого выбирается равным сумме всех информационных разрядов

по модулю 2, т. е.

$$A_{n+1} \equiv \sum_{i=1}^n A_i \pmod{2}, \quad (5-1)$$

где  $A_{n+1}$  — контрольный разряд, а  $A_i$  — информационные разряды кода ( $i = 1 - n$ ). В самом деле, если, допустим, имеется четырехразрядный код, во всех разрядах которого записаны единицы, то сумма этих единиц по модулю 2 даст нуль (четно). Следовательно, чтобы сохранить четность этого кода, в его 5-й контрольный разряд должен быть записан 0, поскольку четность всей информационной комбинации определяется по сумме всех разрядов, включая и контрольный. Аналогично, если во взятом нами четырехразрядном коде единицы записаны только в трех его разрядах, то в контрольный разряд следует записать 1, так как сумма четырех разрядов кода будет нечетной. При этом сумма всей информационной комбинации окажется нулевой, т. е. четной. Другими словами, условие выполнения четности всей информационной комбинации можно выразить как

$$\sum_{i=1}^{n+1} A_i \pmod{2} \equiv 0. \quad (5-2)$$

Таким образом, для осуществления контроля передаваемой информации по четности каждое слово перед передачей должно быть закодировано.

Наиболее просто кодирование информации осуществляется при ее передаче в последовательном коде. Один из вариантов схемы такого кодирующего устройства приведен на рис. 5-1.

Кодируемое  $n$ -разрядное слово поразрядно подается в буферный регистр, проходя одновременно разряд за разрядом через триггер  $T$ , работающий по счетному входу. Через  $n$  сдвигов все слово будет записано в буферный регистр. Если слово содержало четное число единиц, то триггер  $T$  окажется в нулевом состоянии, а если нечетное, то в единичном. Поданный вслед за этим очередной  $(n + 1)$ -й импульс сдвига на схему  $I$  переписет значение знака четности в последний (контрольный) разряд регистра. Закодированное таким образом слово будет передано по назначению в соответствующее устройство (не обязательно в последовательном коде).

Если при передаче последовательного кода используется контроль по нечетности (как отмечалось ранее, при пере-

даче информации в последовательном коде контроль по нечетности имеет преимущество перед контролем по четности), то схема от этого не меняется за исключением того,

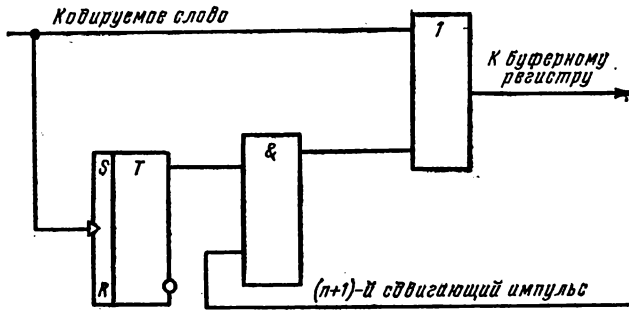


Рис. 5-1. Схема формирования разряда четности в последовательном коде.

что на схему И сигнал снимается не с единичного, а с нулевого выхода триггера  $T$ . Таким образом, если в слове окажется четное число единиц, то триггер  $T$  после прохождения слова вернется в 0, однако в регистр будет переписан обратный код этого значения, т. е. 1, что и создаст нечетность кодовой комбинации в буферном регистре.

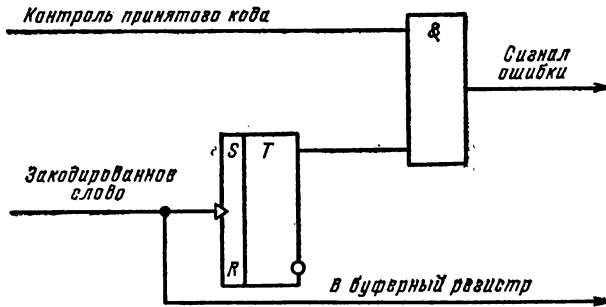


Рис. 5-2. Схема контроля четности последовательного кода.

Проверка на четность (декодирование) информации, переданной в последовательном коде, также не представляет большой сложности и может быть осуществлена с помощью аналогичного триггера. На рис. 5-2 показана одна из возможных схем такого устройства. Закодированное

слово в последовательном коде поразрядно поступает на приемный буферный регистр устройства, проходя одновременно на счетный вход триггера  $T$ . После прохождения

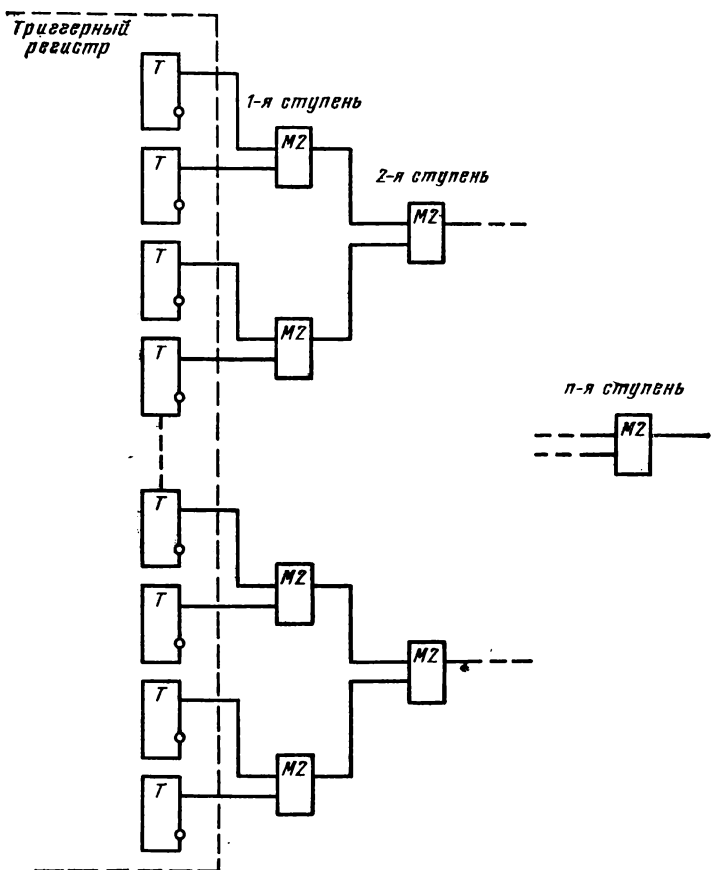


Рис. 5-3. Функциональная схема проверки на четность параллельного кода.

всего слова (вместе с контрольным разрядом) триггер устанавливается в исходное нулевое состояние, если применен контроль на четность и в переданной комбинации нет ошибок. Следующий за этим сигнал стробирования ошибки на выход схемы при этом не пройдет.

При контроле на нечетность в схеме используется не единичный, а нулевой выход триггера. Нетрудно понять,

что после прохождения комбинации с нечетным числом единиц триггер окажется в единичном состоянии (если в принятой комбинации нет ошибки). Низкий уровень с его нулевого выхода заблокирует схему И и сигнала ошибки на выходе схемы не появится. (Как в предыдущей, так и данной схеме триггер  $T$  устанавливается в исходное состояние каждый раз перед началом работы с очередным словом.)

Проверка на четность параллельного кода осуществляется сложнее и требует для своей реализации построения специальной комбинационной схемы с  $n$  входами. Основным ее элементом является схема на два входа, суммирующая два аргумента по модулю 2 (схема  $M2$ ) (рис. 5-3).

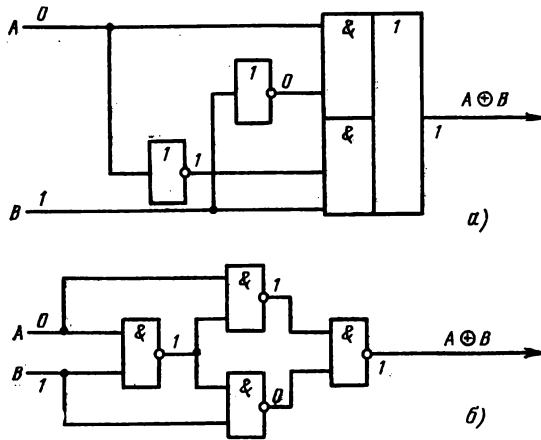


Рис. 5-4. Двухвходовая схема НЕЧЕТ, построенная из элементов НЕ, И — ИЛИ (а) и из элементов И — НЕ (б).

Единичный сигнал на выходе такого элемента будет при условии появления единицы только на одном из его входов. Наличие двух единиц на обоих входах элемента дает на его выходе нуль, т. е. исключается элементом. Следовательно, отсутствие единицы (сигнала ошибки) на выходе комбинационной схемы возможно лишь при условии четного числа единиц в контролируемом коде. Нарушение четности приводит к появлению сигнала ошибки.

Приведенная на рисунке комбинационная схема называется пирамидальной. Количество элементов, входящих в схему, при  $n$ -разрядном регистре равно  $n - 1$ , а количество ее ступеней определяется из выражения  $\log_2 n$  с округ-

лением в большую сторону. Например, для 32-разрядного регистра потребуется пятиступенчатая схема из 31 элемента.

В качестве элементов могут использоваться различные по построению схемы. На рис. 5-4 приведены два варианта схем, суммирующих аргументы по модулю 2. Первая из них построена на логических элементах НЕ, И — ИЛИ, вторая — на элементах И — НЕ. Обе схемы работают

Т а б л и ц а 5-1

A	B	Нечет
0	0	0
0	1	1
1	0	1
1	1	0

в соответствии с табл. 5-1. Схемы подобного типа называются схемами НЕЧЕТ, так как на их выходе образуется 1, если на вход подается нечетное число единиц. В отличие от этих схем схема ЧЕТ образует на своем выходе 1, если на ее вход подается четное число.

Рассмотренная схема контроля на четность требует подачи на ее вход только прямого (или только инверсного) кода каждого разряда. Однако большее применение находят схемы, в которых используются оба выхода каждого триггера, что создает более равномерную нагрузку на триггеры.

Прежде чем перейти к рассмотрению такого рода схем, напомним, что условие выполнения четности информационной комбинации можно представить выражением (5-2). Сумму нескольких двоичных разрядов информационной комбинации по модулю 2 будем называть четностью данной группы разрядов. Тогда общую четность комбинации (общую сумму разрядов комбинации, включая контрольный разряд) можно выразить формулой

$$S_{n+1} = \sum_{i=1}^{n+1} A_i \pmod{2}. \quad (5-3)$$

Схему, вырабатывающую общую четность согласно последнему выражению, можно построить в виде цепочки или в виде пирамиды [1]. Основной элемент обеих этих схем и его условное обозначение изображены на рис. 5-5.

Элемент вырабатывает сумму  $S$  по аргументам  $B$  и  $C$  в соответствии с формулами:

$$S = (B \wedge \bar{C}) \vee (\bar{B} \wedge C); \quad \bar{S} = (B \wedge C) \vee (\bar{B} \wedge \bar{C}).$$

При этом четность каждого аргумента (слагаемого) и суммы кодируется (с учетом их инверсий) двухразрядным



кодом: 10 — четно (0), а 01 — нечетно (1). На рисунке показан пример образования четной суммы по обоим четным аргументам.

Схема четности последовательного типа (рис. 5-6) содержит  $n + 1$  триггеров и  $n$  элементов. Каждый элемент имеет четыре входа, на два из которых подается четность группы из  $i$  всех предыдущих (младших) разрядов, а на два других входа — четность  $(i + 1)$ -го разряда. Таким образом, с выхода  $i$ -го элемента снимается четность группы, состоящей из  $i + 1$  младших разрядов, а с выхода последнего  $n$ -го элемента снимается общая четность всей информационной комбинации  $S_{n+1}$  (5-3).

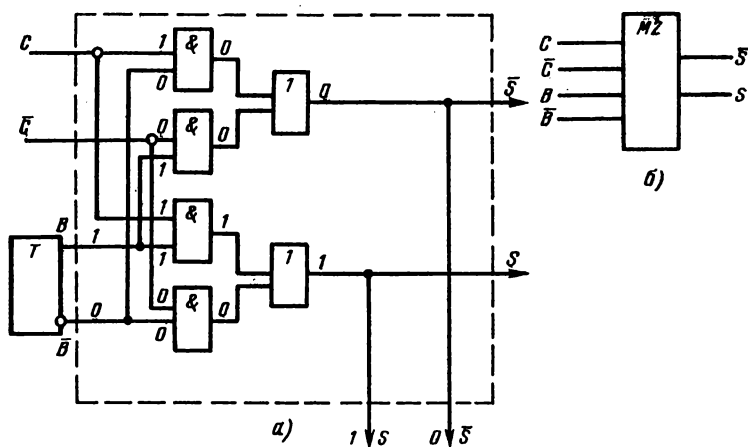


Рис. 5-5. Элемент схемы четности (а) и его условное обозначение (б).

Схема может работать на передачу слова и на его прием. При передаче слова предварительно осуществляется его кодирование, которое заключается в формировании контрольного разряда четности  $A_{n+1}$ . При формировании этого разряда импульс четности  $S_{n+1}$  поступает на счетный вход контрольного разряда и изменяет его содержимое на противоположное, если  $S_{n+1} = 1$ . В результате этой операции возникает такое значение разряда  $A_{n+1}$ , при котором удовлетворяется условие (5-2), т. е. общая сумма  $S_{n+1}$  становится четной. После этого слово передается по назначению (линии передачи и приема слова на схеме не показаны).

При приеме слова осуществляется проверка его на четность, т. е. проверяется выполнение условия  $S_{n+1} = 0$  путем подачи на второй вход схемы  $I_2$ , импульса «контроль принятого кода». В случае невыполнения данного условия

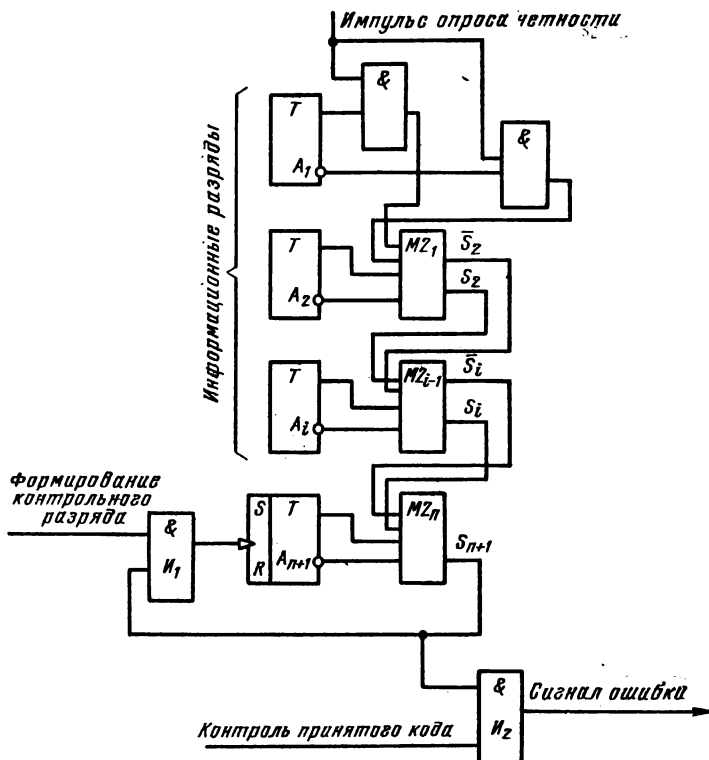


Рис. 5-6. Схема кодирования и декодирования последовательного типа для кода с проверкой четности,

с выхода схемы выдается сигнал ошибки. Выработка  $S_{n+1}$  в обоих случаях производится путем подачи на схему импульса опроса четности.

Вторая схема (рис. 5-7) представляет собой многоступенчатую пирамиду, составленную из тех же элементов. Первая ступень схемы объединяет по два все разряды кода. При нечетном числе разрядов один из них связывается непосредственно со второй ступенью (как это и сделано на рисунке). Каждая последующая ступень комбинационной

Схемы объединяет выходы (по два) предыдущей. Последняя ступень вырабатывает общую четность. В случае нечетного количества выходов в  $(i - 1)$ -й ступени один из них поступает в  $(i + 1)$ -ю ступень. Работа схемы аналогична предыдущей.

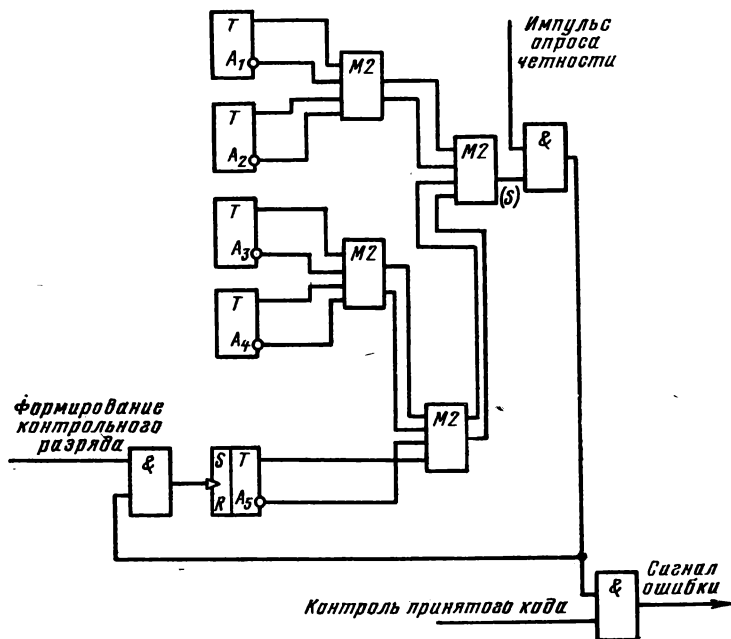


Рис. 5-7. Схема кодирования и декодирования пирамидального типа для кода с проверкой четности.

Подобная многоступенчатая структура комбинационной схемы позволяет существенно сократить объем необходимого оборудования.

## 5-2. КОНТРОЛЬ С ПОМОЩЬЮ КОДА ХЭММИНГА

Как уже отмечалось (гл. 4), кодирование числа в коде Хэмминга заключается в формировании контрольных разрядов числа, значение которых определяется путем проверки на четность (суммирования по модулю 2) отдельных его разрядов по контрольным группам, устанавливающим порядок такой проверки. При этом первая проверка (формирование первого контрольного разряда) охватывает двоичные знаки,

записанные в первой контрольной группе, т. е. 3, 5, 7-й и т. д. разряды. Результат проверки записывается в первый разряд. Вторая проверка охватывает знаки, записанные во второй контрольной группе (3, 6, 7-й и т. д. разряды), и результат записывается во второй (контрольный) разряд и т. д. На рис. 5-8 приведена схема, которая реализует этот алгоритм.

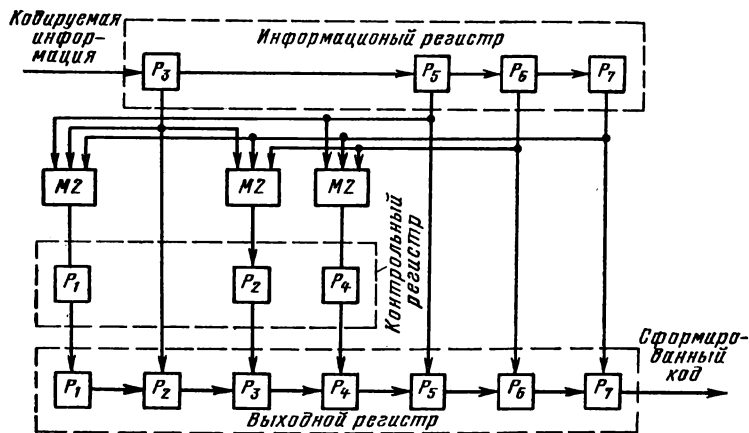


Рис. 5-8. Схема кодирующего устройства для кода Хемминга.

Схема рассчитана на реализацию 7-значного кода Хемминга, имеющего четыре информационных и три контрольных разряда. В соответствии с табл. 4-3 образуются три контрольные группы разрядов, от каждой из которых информация подается на свою суммирующую схему по модулю 2. Результаты суммирования в каждой такой схеме будут записаны в соответствующие разряды контрольного регистра, а затем переписаны в выходной, где и зафиксирован сформированный код, готовый к передаче.

На рис. 5-9 приведена схема декодирующего устройства, предназначенного для исправления одиночных ошибок, возникших при передаче закодированного числа. Суммирующие схемы по модулю 2 здесь служат для определения номера разряда, в котором возникла ошибка.

Схема работает следующим образом. Закодированное число (7-значный код Хемминга) записывается на приемный регистр, после чего открываются входы сумматоров по модулю 2. Если код числа принят без искажения, то на вы-

ходах всех трех сумматоров появятся нули. При этом с выхода дешифратора будет подан сигнал, по которому содержимое 3, 5, 6 и 7-го разрядов приемного регистра переписывается в выходной регистр, где, таким образом, зафиксировано декодированное число. При наличии ошибки в каком-либо из разрядов принятого кода на регистр коррекции в двоичном коде запишется номер искаженного разряда.

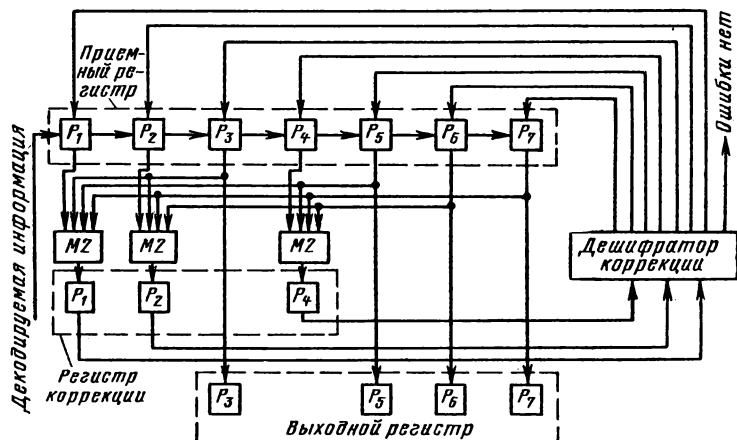


Рис. 5-9. Схема декодирующего устройства для кода Хемминга.

Затем этот номер поступит на дешифратор коррекции, с которого после расшифровки номера на соответствующий разряд приемного регистра будет подан сигнал, исправляющий ошибку.

Аналогичные схемы могут применяться и для параллельного ввода и вывода информации.

### 5-3. КОНТРОЛЬ С ПОМОЩЬЮ РАВНОВЕСНЫХ КОДОВ

В гл. 4 рассматривался вопрос о принципе построения равновесных кодов, которые получили такое название за заданное (равновесное) количество нулей и единиц в каждом слове кода. Так, все слова в коде «2 из 5» содержат две единицы и три нуля, а например в коде «2 из 7» — две единицы и пять нулей. Равновесные коды в основном применяются в передаче данных и в ЗУ машины. В арифметических устройствах (АУ) и устройствах управления (УУ) равновесные коды встречаются сравнительно редко, так как их использование в этих устройствах связано с существ-

венным усложнением последних благодаря большей избыточности аппаратуры. Равновесные коды эффективны при малой вероятности появления симметричных ошибок, т. е. когда происходит одновременное пропадание единицы в одном разряде и появление в другом. (На практике такие ошибки появляются крайне редко.)

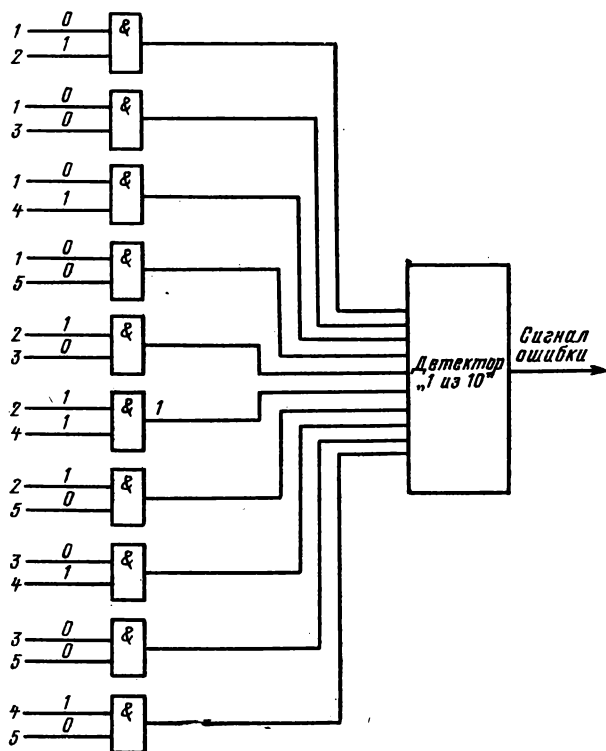


Рис. 5-10. Схема контроля «2 из 5».

Схемы могут строиться как специально для равновесных кодов, так и для равновесных кодов с использованием пороговых схем. Первые оказываются более громоздкими, поэтому предпочтение обычно отдается вторым. Рассмотрим типичную схему контроля для кода «2 из 5», предназначенную для обнаружения всех возможных запрещенных комбинаций единиц, которые могут появиться в слове (рис. 5-10). Схема построена на логических элементах типа

И, выходы которых объединяются специальной схемой — детектором. Детектор строится таким образом, что фиксирует ошибку при поступлении на его вход количества единиц больше или меньше одной.

Код слова подается на входы схемы в соответствии с разрядностью, показанной на рисунке. Возникновение в слове ошибки приводит к появлению на выходах схем *И* сразу нескольких единиц (вместо одной) или к полному их пропаданию. В обоих случаях ошибка фиксируется детектором. В качестве примера на рисунке показано поступление на схему числа 4, закодированного в соответствии с табл. 4-6: 01010. При этом на вход детектора поступает одна единица и девять нулей. Нетрудно проследить, сколько появится единиц на выходах схем *И*, если в слове возникает ошибка (допустим, единица в первом разряде). Вместо десяти схем *И* можно использовать дешифратор для кода «2 из 5» на десять выходов. В этом случае детектор будет фиксировать и ошибки в работе самого дешифратора.

В качестве детектора «1 из 10» удобно использовать пороговые схемы, к рассмотрению которых мы и перейдем.

#### 5-4. ПОРОГОВЫЕ СХЕМЫ

Пороговые схемы используются для контроля работы схем, на выходах которых в каждый момент времени может появиться лишь строго определенное количество сигналов. Схема, контролирующая выполнение такого условия, называется детектором.

Функционально пороговая схема представлена на рис. 5-11. В скобках указывается порог схемы, который означает: если число единиц на

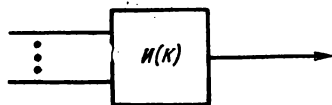


Рис. 5-11. Пороговая схема с порогом  $K$ .

входе схемы равно или больше числа  $K$ , то на выходе появится 1, в противном случае на ее выходе будет 0.

На двух таких пороговых схемах можно построить детектор, который в общем случае можно назвать детектором « $K$  из  $N$ » (рис. 5-12). Такой детектор рассчитан на поступление кодового слова, содержащего  $K$  единиц и  $N - K$  нулей.

Схема с порогом  $K$  вырабатывает на выходе 1, если на ее входе  $K$  или более единиц, а схема с порогом  $K + 1$  — если на ее входе  $K + 1$  или более единиц. При поступле-

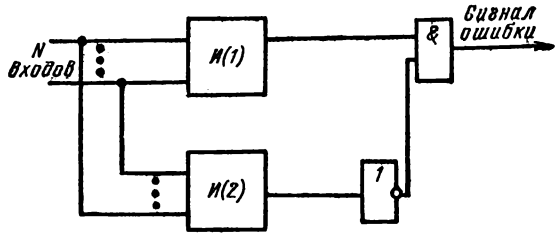


Рис. 5-12. Детектор «К из N» на пороговых схемах.

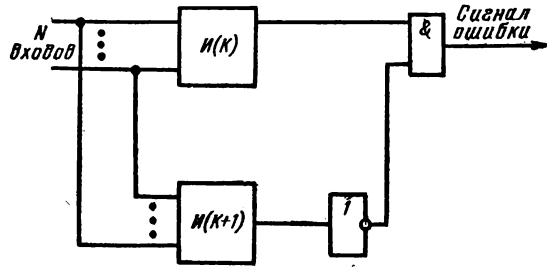


Рис. 5-13. Детектор «1 из N» на пороговых схемах.

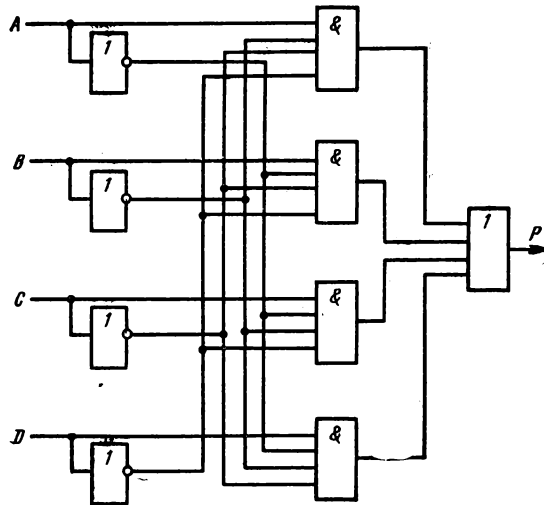


Рис. 5-14. Детектор «1 из 4».



нии на вход детектора количества единиц, меньшего  $K$ , на его выходе будет 0. При поступлении  $K + 1$  или больше единиц нижняя схема вырабатывает 1, которая инвертируется на элементе  $HE$ , и на выходе детектора опять будет 0. Только при одном условии, когда число входных единиц равно  $K$ , верхняя схема выработает 1, а нижняя — 0 (ее порог выше). Следовательно, на выходе детектора появится 1, характеризующая исправность в работе.

Если теперь в качестве порога выберем  $K = 1$ , то получим детектор, используемый в схемах контроля равновесных кодов. Принципиальная схема такого детектора приведена на рис. 5-13 и в дополнительном разъяснении не нуждается.

В качестве варианта на рис. 5-14 показана схема детектора «1 из 4», построенная в виде полного дешифратора функции. На выходе схемы образуется 1, если на входы подается одна единица и три нуля. Схема работает в соответствии с логическим выражением

$$P = (A \wedge \bar{B} \wedge \bar{C} \wedge \bar{D}) \vee (\bar{A} \wedge B \wedge C \wedge D) \vee \\ \vee (\bar{A} \wedge \bar{B} \wedge C \wedge \bar{D}) \vee (\bar{A} \wedge \bar{B} \wedge \bar{C} \wedge D).$$

На практике подобный тип схем применяется только при малом числе входов ввиду их громоздкости.

### 5-5. СХЕМЫ КОНТРОЛЯ ПО МОДУЛЮ

В предыдущей главе рассматривался вопрос организации контроля арифметических операций по модулю, в частности по модулю 3. Этот код позволяет обнаруживать одиночные ошибки и часть двойных, при которых правильный и ошибочный результаты имеют несовпадающие остатки от деления на 3. К достоинствам кода относится и то, что он позволяет проводить отдельный контроль, который заключается в следующем. Каждое число, участвующее в операции, имеет контрольные разряды, в которые записывается значение остатка (при контроле по модулю 3 таких разрядов два, куда записывается одно из трех значений остатка: 0, 1 или 2). Одновременно с выполнением операции над числами в основном сумматоре (см. рис. 4-1) выполняется та же операция над остатками этих чисел в сумматоре остатков. После этого определяется остаток результата операции в основном сумматоре и сравнивается с результатом операции над остатками исходных чисел. В случае равен-

ства остатков операция в основном сумматоре считается выполненной правильно.

При реализации такого контроля особенно важным является построение наиболее экономичных схем формирования (вычисления) остатков, которые определяют основной объем затрачиваемого оборудования. Кодирование по модулю 3 обладает свойством, позволяющим находить остатки, не прибегая к прямому делению чисел, и при этом создавать достаточно экономичные схемы формирования остатков, содержащие минимум аппаратуры. Рассмотрим общий принцип построения таких схем.

Двоичное число

$$A = a_0 2^n + a_1 2^{n-1} + \dots + a_{n-1} 2 + a_n$$

можно представить в четверичной системе счисления:

$$A = k_0 4^m + k_1 4^{m-1} + \dots + k_{m-1} 4 + k_m,$$

где

$$\begin{aligned} m &= \frac{n-1}{2}; \\ k_0 &= a_0 2 + a_1; \\ k_1 &= a_2 2 + a_3; \\ &\dots \dots \dots \\ k_m &= a_{n-1} 2 + a_n. \end{aligned}$$

Тогда остаток числа  $A$  запишется:

$$R(A) = R(k_0 4^m) + R(k_1 4^{m-1}) + \dots + R(k_m).$$

Число  $4^i$  имеет остаток по модулю 3, равный +1, так как

$$4^i = (3 + 1)^i = 3^i + C_1 3^{i-1} + C_2 3^{i-2} + \dots + C_{i-1} 3 + 1,$$

где  $C_1, C_2, \dots, C_{i-1}$  — биномиальные коэффициенты.

Тогда выражение для остатка числа  $A$  можно переписать в виде

$$R(A) = R(k_0) + R(k_1) + \dots + R(k_m) = R(k_0 + k_1 + \dots + k_m).$$

Из последнего выражения следует, что остаток числа  $A$  может быть найден путем определения остатка суммы четверичных цифр этого числа, что легко выполняется с помощью соответствующей комбинационной схемы. На рис. 5-15 представлена схема формирования остатка по модулю 3 восьмиразрядного двоичного числа. Первая ступень схемы определяет попарно остаток по модулю 3 чет-

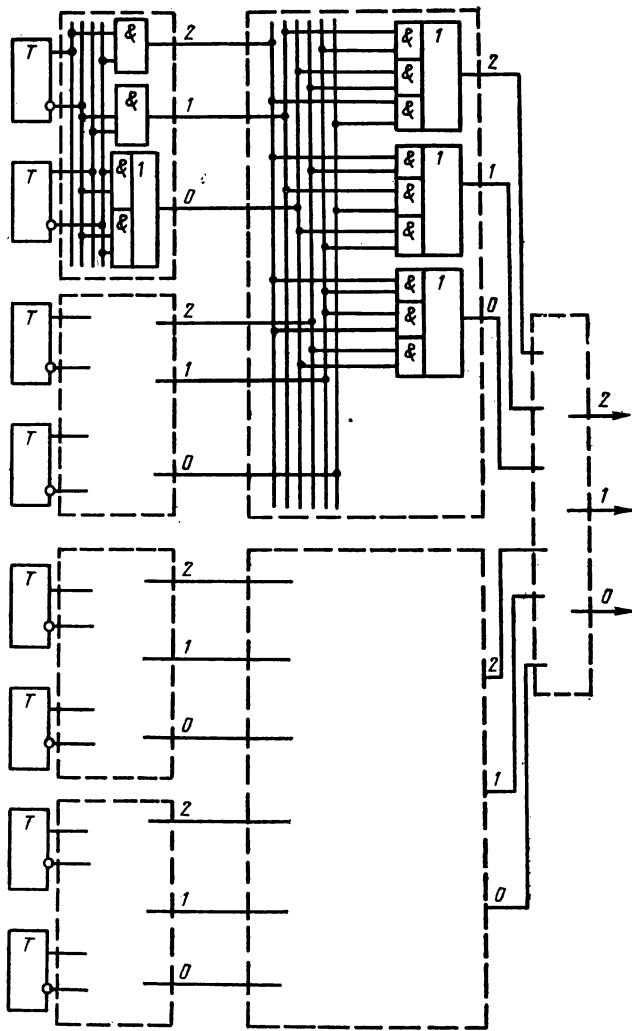


Рис. 5-15. Схема формирования остатка по модулю 3.

веричных цифр, расположенных в соседних разрядах, а ее вторая ступень — остаток по модулю 3 от суммы цифр, выработываемых первой ступенью, и т. д. Сигнал, который появляется на выходе, соответствует остатку контролируемого числа (0, 1 или 2). Схема на большее число разрядов строится по тому же принципу.

#### 5-6. ОБНАРУЖЕНИЕ ОШИБОК МЕТОДОМ АППАРАТУРНОГО КОНТРОЛЯ

Аппаратурный контроль основан на введении в состав ЭВМ дополнительного контрольного оборудования. По характеру использования контрольного оборудования различают внешний аппаратурный контроль и оперативный аппаратурный контроль (см. гл. 2).

Внешний аппаратурный контроль проводится с использованием специальной сервисной аппаратуры при автономной работе ЭВМ при проведении профилактических и ремонтных работ.

При оперативном аппаратурном контроле контрольное оборудование функционирует одновременно с основной аппаратурой.

Оперативный аппаратурный контроль может быть осуществлен различными методами и в принципе позволяет контролировать все этапы работы ЭВМ при выполнении каждой отдельной операции, а также последовательности операций.

Аппаратурные методы оперативного контроля позволяют непрерывно контролировать достоверность функционирования ЭВМ практически без снижения быстродействия, обнаруживая как систематические ошибки, так и случайные сбои в работе непосредственно в момент их возникновения, чем предотвращается бесцельная трата машинного времени на получение заведомо неправильного результата.

Простейшим методом аппаратурного контроля является *дублирование* контролируемых узлов ЭВМ с последующим сравнением выходных сигналов двух одинаковых схем и выдачей сигнала ошибки при несовпадающих сигналах. Таблица 5-2 определяет логику работы схемы сравнения.

Таким образом, сигнал ошибки является результатом операции сложения по модулю 2 выходных сигналов обеих схем

$$c = a \oplus b.$$

Таблица 5-2

Выходной сигнал основной схемы $a$	Выходной сигнал контрольной схемы $b$	Сигнал ошибки $c$
0	0	0
0	1	1
1	0	1
1	1	0

Существуют и другие названия данной операции — схема неравнозначности, операция «исключающее ИЛИ». С использованием системы функций И — ИЛИ — НЕ операцию сравнения можно представить в виде

$$c = a\bar{b} \vee \bar{a}b.$$

На рис. 5-16 приведена функциональная схема формирования сигнала ошибки при использовании метода дублирования (схема M2). Недостатком дублирования является

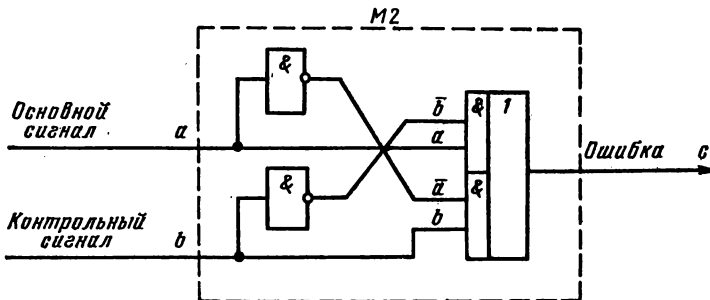


Рис. 5-16. Схема формирования сигнала ошибки.

увеличение более чем в 2 раза необходимого оборудования, поэтому дублирование как способ аппаратного контроля применяется в тех случаях, когда использование других методов контроля затруднено или невозможно.

Наибольшую практическую ценность имеют такие способы аппаратного контроля, которые не требуют большого количества дополнительного оборудования.

Из-за большого разнообразия функций, выполняемых различными узлами ЭВМ, не существует единого способа, который подходил бы для организации эффективного контроля любых узлов. Учет особенностей построения отдельных узлов (или устройств) ЭВМ позволяет строить эффек-

тивные схемы контроля при допустимых затратах контрольного оборудования.

*Контроль ввода-вывода информации.* Ввод информации в ЭВМ осуществляют как дистанционно, используя каналы связи, так и через местные устройства ввода. Ошибки, возникающие при дистанционном вводе, во многом подобны ошибкам в запоминающих устройствах, поэтому они будут рассмотрены в соответствующем параграфе.

В данном разделе рассматриваются методы схемного обнаружения ошибок при работе с местными устройствами ввода.

Почти все современные вычислительные машины имеют устройства ввода с перфокарт или с перфолент или одновременно оба вида этих устройств. Информация на перфоносители (перфокарты или перфоленты) наносится с помощью пробивок отверстий (перфораций) на специальных устройствах подготовки данных (УПД).

При работе с местными устройствами ввода возможны ошибки трех видов: субъективные ошибки оператора при подготовке перфоносителя, ошибки устройств подготовки данных, вызывающие неправильную перфорацию кодов, и ошибки за счет неправильного преобразования перфораций в электрические сигналы в устройствах ввода.

Для обнаружения ошибок оператор применяют дублирование перфоносителя путем набивки второго комплекта другим оператором. Сравнение двух комплектов перфоносителя производится с помощью контрольно-считывающего устройства. При несовпадении кодов двух контролируемых комплектов перфоносителя происходит автоматический останов контрольно-считывающего устройства, с одновременной индикацией несовпадающих кодов.

Второй способ контроля осуществляется без дублирования перфоносителя. Подготовленный перфоноситель проверяется на контрольно-считывающем устройстве оператором, для чего заново набирается необходимая информация на клавиатуре. При соответствии кода на перфоносителе коду вводимого символа перфоноситель автоматически передвигается на одну позицию, при несоответствии кодов происходит останов с индикацией несовпадающих кодов. Такой способ контроля может быть осуществлен с использованием только одного УПД без контрольно-считывающего устройства.

Обнаружение ошибок устройств подготовки данных и устройств ввода производится с помощью набивки на перфо-

носитель, кроме основной информации, избыточной *контрольной информации* в виде контрольных разрядов или контрольных сумм.

При использовании перфокарт контрольная информация представляет собой контрольную сумму, подсчитываемую по всей карте во время ее набивки. Для этой цели в состав УПД вводится специальный сумматор, при этом любая информация (в том числе и буквенная) рассматривается как двоичное число. Результат суммирования всех информационных строк перфорируется в специально отведенной контрольной строке. (Возможны различные методы вычисления контрольных сумм.)

При вводе перфокарты в ЭВМ арифметическое устройство автоматически вычисляет контрольную сумму по информационным строкам каждой перфокарты и сравнивает ее с контрольной суммой, набитой на контрольной строке. При наличии расхождений в контрольных суммах ввод останавливается. Если при повторном вводе этой перфокарты контрольные суммы сойдутся, то это квалифицируется как случайный сбой и ввод автоматически продолжается. В противном случае необходимо определить место неисправности.

Обнаружение ошибок при работе с перфолентами, как правило, производится с использованием контрольного разряда по четности.

При работе с перфолентами контроль устройств ввода может быть осуществлен методом дублирования чтения каждой строки. Для этого в устройстве ввода применяют двоянный блок считывания (обычно двоянный блок фотодиодов), который дважды преобразует перфорации строки в электрические сигналы с некоторым временным сдвигом. Сравнение считанной информации производится в специальном регистре одновременно с передачей ее в память машины.

Вывод информации из ЭВМ может осуществляться либо в канал связи, либо на местные печатающие или перфорирующие устройства.

При выводе на перфорирующий механизм (а иногда и на печатающий механизм) находит применение метод *обратной проверки*, называемый методом *эхо-контроля*. Сущность этого метода состоит в том, что непосредственно при перфорации или при печати каждого символа электромеханическое кодирующее устройство, связанное с исполнительным органом перфоратора или печатающего устройства, воспро-

изводит код этого символа. При несовпадении кодов схема сравнения формирует сигнал ошибки. Недостатком такой схемы контроля является низкая надежность работы электромеханических кодирующих устройств, которая оказывается значительно ниже, чем надежность самого устройства.

В отдельных устройствах вывода на перфоноситель устанавливают дополнительный блок считывания, который воспринимает информацию после пробивки. Для реализации этого метода требуется дополнительное буферное запоминающее устройство для хранения определенного числа символов в зависимости от расстояния между блоком пробивок и блоком считывания. Применение в настоящее время волоконной оптики для блока считывания позволяет разместить его непосредственно в узле пробивок, при этом отпадает необходимость в буферном запоминающем устройстве.

В ряде устройств вывода применяют косвенный способ эхо-контроля, при котором в качестве признака срабатывания электромагнита печатающего рычага или электромагнита пробивного пуансона используют изменение тока или напряжения в обмотках электромагнита.

### *Контроль внутренних цепей передач информации*

В современных ЭВМ схемы, предназначенные для хранения и передачи цифровой информации, составляют значительную часть оборудования. Сюда входит как хранение и передача данных, так и передача команд и управляющих сигналов.

Основной вид контроля цепей передачи информации основан на применении информационной избыточности и использует коды с обнаружением (реже коррекцией) ошибок.

Отсутствие информационной связи между отдельными разрядами слова при его хранении или передаче позволяет применить для контроля способы, обеспечивающие небольшую схемную избыточность при достаточной эффективности работы.

Как правило, для контроля цепей передачи информации применяют коды с проверкой на *четность* (нечетность).

В современных ЭВМ передача информации производится байтами. Байт представляет собой совокупность 7 информационных и 1 контрольного разряда, несущего информацию о четности (нечетности) кода данного байта. Несколько



байтов, соединенных вместе, образуют слово. Применение байтов особенно удобно для реализации операций ввода-вывода и десятичной арифметики.

При передаче информации байтами обеспечивается более надежный контроль благодаря большому количеству разрядов в слове, однако нужно иметь в виду, что при этом растет количество дополнительного контрольного оборудования.

Вторым преимуществом побайтовой передачи информации является упрощение контроля операций десятичной арифметики вследствие меньшего количества необходимых разрядов арифметического устройства.

В ряде случаев передача двоичной информации с регистра на регистр осуществляется без использования информационной избыточности (контрольных разрядов). При таком способе контроля содержимое приемного регистра возвращается на передающий регистр с одновременной реализацией операции сложения по модулю 2, т. е. с подачей информации на счетные входы триггеров, входящих в состав регистра.

При отсутствии ошибок в передаче информации передающий регистр должен перейти в нулевое состояние. Признаком наличия ошибки будет наличие единицы хотя бы в одном разряде передающего регистра.

### *Контроль работы сумматоров*

Сумматором называется узел ЭВМ, осуществляющий арифметическое сложение кодов чисел. Различают сумматоры последовательного и параллельного действия. В сумматорах последовательного действия передача слагаемых и образование суммы производится последовательно разряд за разрядом. Данные сумматоры, несмотря на простоту технической реализации, не получили широкого распространения в силу низкого быстродействия: для сложения двух  $n$ -разрядных двоичных чисел требуется  $n$  тактов работы ЭВМ.

Преимущественное распространение в современных вычислительных машинах имеют сумматоры параллельного действия, в которых передача чисел и образование суммы происходит одновременно для всех разрядов. Значение цифры суммы  $S_i$  для  $i$ -го разряда зависит от соответствующих этому разряду цифр слагаемых  $a_i$  и  $b_i$ , а также от переноса из предыдущего разряда  $c_{i-1}$ . Функционирование

сумматора может быть описано таблицей истинности (табл. 5-3). В соответствии с данной таблицей работу одноразрядного полного сумматора можно описать следующей системой булевых функций:

$$S_i = a_i \oplus b_i \oplus c_{i-1};$$

$$c_i = a_i b_i \vee a_i c_{i-1} \vee b_i c_{i-1}.$$

Т а б л и ц а 5-3

Первое слагаемое $a_i$	Второе слагаемое $b_i$	Перенос из предыдущего разряда $c_{i-1}$	Сумма $S_i$	Перенос в старший разряд $c_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

На рис. 5-17 приведена функциональная схема одноразрядного полного сумматора комбинационного типа на основе логических элементов И — ИЛИ.

Различают два основных типа сумматоров в соответствии со способом образования переноса: с последовательным и с параллельным переносом.

В сумматорах первого типа (рис. 5-18) перенос, возникающий в  $i$ -м разряде, зависит от переноса из предыдущего разряда, который в свою очередь зависит от другого предыдущего переноса. Таким образом, полная схема формирования переносов представляет собой последовательную цепь, при распространении сигнала по которой существенно увеличивается общее время задержки, что снижает быстродействие данного типа сумматора.

Для ускорения процесса сложения нужно сократить путь распространения переноса. В этих случаях применяют схему параллельного переноса, используя формулу

$$c_i = a_i b_i \vee (a_i \vee b_i) a_{i-1} b_{i-1} \vee (a_i \vee b_i) (a_{i-1} \vee b_{i-1}) a_{i-2} b_{i-2} \vee \dots$$

В этой схеме каждый перенос формируется независимо от предыдущих переносов, что приводит к увеличению

быстродействия сумматора. Недостатком данной схемы является усложнение цепей переноса, особенно в многоразрядных сумматорах.

Неисправности схемы сумматора вызывают появление ошибок различного типа. Их можно разделить на два основных вида в зависимости от того, появляется ли в результате неисправности одиночная ошибка в разряде суммы или групповые ошибки в разряде суммы и переноса.

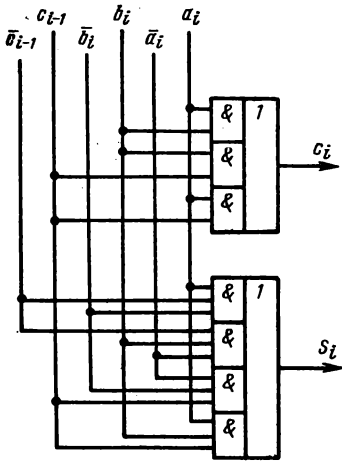


Рис. 5-17. Полный одноразрядный сумматор комбинационного типа.

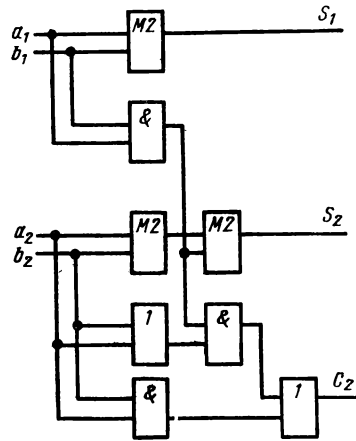


Рис. 5-18. Функциональная схема сумматора с последовательным переносом.

Одиночная ошибка в разряде суммы вызывается только неисправностью в схеме формирования разряда суммы (т. е. в полусумматоре). Групповая ошибка в разряде суммы и переноса вызывается только неисправностью схемы переноса.

В качестве примера рассмотрим сложение двух четырехразрядных двоичных чисел

$$\begin{array}{r}
 \phantom{+} 1 \phantom{0} 1 \phantom{0} 1 \\
 \phantom{+} \overline{\phantom{0}} \phantom{0} \overline{\phantom{0}} \overline{\phantom{0}} \phantom{0} \phantom{0} \\
 + 0 \phantom{0} 1 \phantom{0} 1 \\
 \hline
 1 \phantom{0} 1 \phantom{0} 0
 \end{array}$$

В случае неисправности схем формирования полусуммы, например второго разряда,

$$\begin{array}{r}
 \phantom{+} 0 \ 0 \ 1 \ 1 \\
 + 0 \ 1 \ 1 \ 1 \\
 \hline
 1 \ 0 \ 0 \ 0
 \end{array}$$

↑ \_\_\_\_\_ ошибка

При этом важно заметить, что данная ошибка не распространяется на старшие разряды.

При неисправности в схеме формирования переноса может произойти искажение одного разряда суммы

$$\begin{array}{r}
 \phantom{+} 0 \ 0 \ 1 \ 1 \\
 + 0 \ 1 \ 1 \ 1 \\
 \hline
 1 \ 0 \ 0 \ 0
 \end{array}$$

↑ \_\_\_\_\_ ошибочный разряд

или группы следующих друг за другом разрядов

$$\begin{array}{r}
 \phantom{+} 0 \ 0 \ 1 \\
 + 0 \ 1 \ 1 \ 1 \\
 \hline
 0 \ 1 \ 1 \ 0
 \end{array}$$

↑ \_\_\_\_\_ ошибочные разряды

Из множества разработанных к настоящему времени методов контроля сумматоров наиболее часто применяются контроль по четности и контроль с помощью остаточных кодов.

В качестве простейшего метода контроля работы сумматора может быть использован контроль по четности. При сложении чисел  $a$  и  $b$  разряды суммы  $S$  образуются в соответствии с выражениями:

$$\begin{aligned}
 S_1 &= a_1 \oplus b_1; \\
 S_2 &= a_2 \oplus b_2 \oplus c_1; \\
 &\dots \dots \dots \\
 S_n &= a_n \oplus b_n \oplus c_{n-1}.
 \end{aligned}$$

Сложив все  $n$  вышеприведенных равенств по модулю 2, получим:

$$P_s = P_a \oplus P_b \oplus P_c,$$

где  $P_s$  — четность суммы;  $P_a, P_b$  — четность слагаемых;  $P_c$  — четность переносов.

Данное уравнение можно использовать для контроля четности суммы  $S$ , сравнивая предсказанную четность в соответствии с этим уравнением с действительной четностью

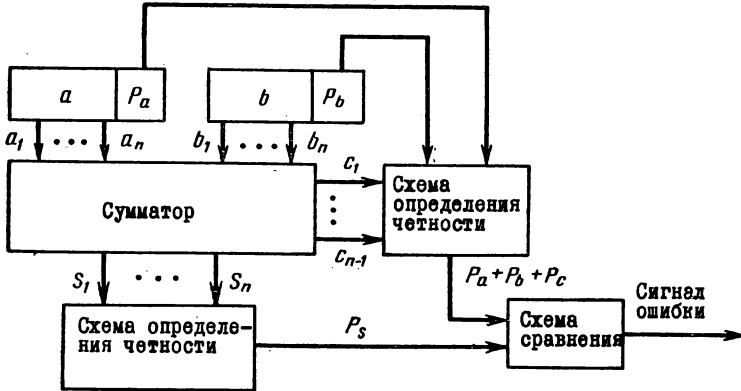


Рис. 5-19. Функциональная схема сумматора, контролируемого по четности.

суммы. Таким образом, для реализации контроля по четности необходимо иметь контрольные разряды четности входных слагаемых  $P_a$  и  $P_b$ , сформировать четность переносов  $P_c$  и четность суммы  $P_s$  и сравнить ее с выражением  $P_a \oplus P_b \oplus P_c$ .

В качестве примера рассмотрим сложение четырехразрядных чисел с контрольным разрядом:

$$\begin{array}{r|l}
 \begin{array}{cccc}
 & 1 & 1 & \\
 & \leftarrow & \leftarrow & \\
 A = & 0 & 0 & 1 & 1 & \\
 B = & 0 & 1 & 1 & 0 & \\
 \hline
 S = & 1 & 0 & 0 & 1 & 
 \end{array}
 & \begin{array}{l}
 0 \\
 0 \\
 P_c = 0 \\
 P_0 = 0
 \end{array}
 \end{array}$$

На рис. 5-19 приведена функциональная схема контроля сумматора по четности. Сигнал ошибки в этой схеме выда-

ется при несовпадении истинной четности суммы  $P_s$  с предсказанной четностью.

Данная схема контроля позволяет обнаружить единичную и любое нечетное число ошибок. Как уже указывалось выше, неисправности в цепи переноса вызывают появление четного числа ошибок — в разряде переноса и в разряде суммы, поэтому четность суммы при этом не изменится. Таким образом, неисправности в цепи переноса данной схемой контроля не обнаруживаются. Так как схема формирования переноса составляет значительную часть схемы сумматора, особенно для сумматоров с параллельным переносом, схема контроля по четности для сумматора будет обладать малой обнаруживающей способностью.

Распространенным методом обнаружения ошибки в схеме формирования переноса является применение дублированного переноса с использованием дополнительной аппаратуры, не зависящей от схемы формирования истинного переноса. Ошибки в схеме формирования переноса можно обнаружить путем поразрядного сложения по модулю 2 дублированного и истинного переноса. На рис. 5-20 представлена функциональная схема контроля по четности сумматора с последовательным переносом на основе схемы дублирования переноса. Данная схема позволяет обнаружить одиночные и нечетное число ошибок как в схеме формирования суммы, так и в схеме формирования переносов.

В сумматорах с параллельным переносом формирование дублированного параллельного переноса не эффективно, поскольку это приводит к резкому увеличению аппаратурных затрат.

Контроль работы сумматоров можно осуществить с использованием *парафазной* или *двухпроводной* логики. При этом для представления каждой переменной используются две линии: прямой сигнал и инверсный. Поскольку в двухпроводной логике для передачи двоичной информации используются две линии, то сигналы на этих линиях могут принимать четыре возможных состояния: 00, 01, 10, 11. Разрешенными сигналами являются 01 и 10, т. е. сигнал на одной линии должен быть инверсным, находиться в противофазе по отношению к сигналу на второй линии. Состояния 00 и 11 являются запрещенными, как неудовлетворяющие условию инверсности, и появление их на выходе логической схемы свидетельствует об ошибке в схеме.

Преимуществом двухпроводной логики является возможность при разработке логических схем строить более

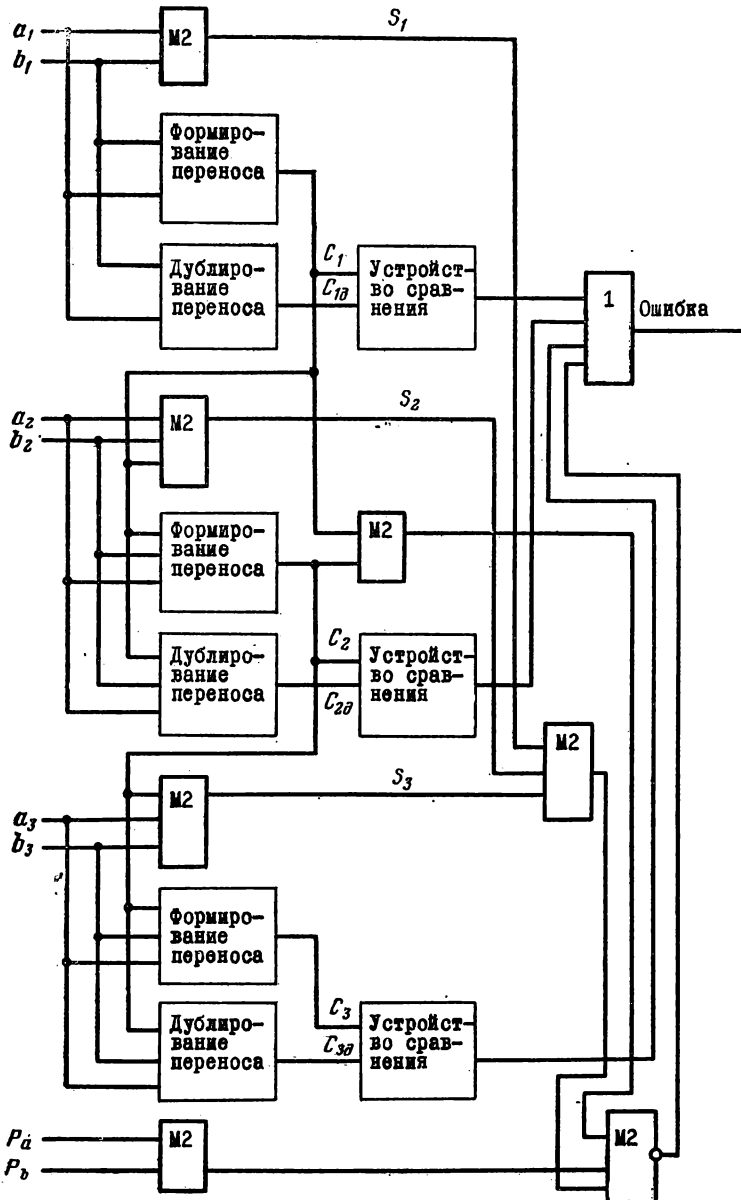


Рис. 5-20. Функциональная схема контроля сумматора по четности с дублированием переносов.

быстродействующие схемы за счет отсутствия временных задержек на инвертирование сигналов.

Сумматоры на основе парафазной логики строятся с учетом выражений, приведенных в табл. 5-4.

Таблица 5-4

$a_i$	$b_i$	$c_{i-1}$	$S_i$	$\bar{S}_i$	$c_i$	$\bar{c}_i$
0	0	0	0	1	0	1
0	0	1	1	0	0	1
0	1	0	1	0	0	1
0	1	1	0	1	1	0
1	0	0	1	0	0	1
1	0	1	0	1	1	0
1	1	0	0	1	1	0
1	1	1	1	0	1	0

В соответствии с этой таблицей

$$S_i = (a_i \bar{b}_i \vee \bar{a}_i b_i) \bar{c}_{i-1} \vee (\overline{a_i \bar{b}_i \vee \bar{a}_i b_i}) c_{i-1};$$

$$\bar{S}_i = (a_i \bar{b}_i \vee \bar{a}_i b_i) c_{i-1} \vee (\overline{a_i \bar{b}_i \vee \bar{a}_i b_i}) \bar{c}_{i-1};$$

$$c_i = a_i b_i \vee (a_i \vee b_i) c_{i-1};$$

$$\bar{c}_i = \overline{a_i b_i} (\overline{a_i \bar{b}_i \vee \bar{a}_i b_i}) \vee \overline{a_i b_i} \bar{c}_{i-1}.$$

На рис. 5-21 приведена функциональная схема полного одноразрядного сумматора с парафазной логикой на основе элементов И—ИЛИ—НЕ.

Ошибки в схеме сумматора с парафазной логикой обнаруживаются на основе сравнения выходов  $S_i$  и  $\bar{S}_i$ ,  $c_i$  и  $\bar{c}_i$ . В данной схеме обнаруживаются все ошибки, вызванные одиночной неисправностью.

Рассмотренные схемы контроля сумматоров по четности и на основе парафазной логики рассчитаны на обнаружение одиночных ошибок. Для увеличения обнаруживающей способности схем контроля используют *остаточные* коды, принцип формирования которых рассмотрен в предыдущих главах.

Остаточные коды можно использовать для обнаружения или исправления одиночных, групповых или пакетов ошибок. Но, как правило, остаточные коды для исправления ошибок применяются редко, что связано с резким увеличе-



нием объема контрольного оборудования. В основном остаточные коды применяют для обнаружения ошибок; при этом чаще всего используется контроль по модулю 3.

Контроль сумматора по модулю 3 рассмотрен в гл. 4.

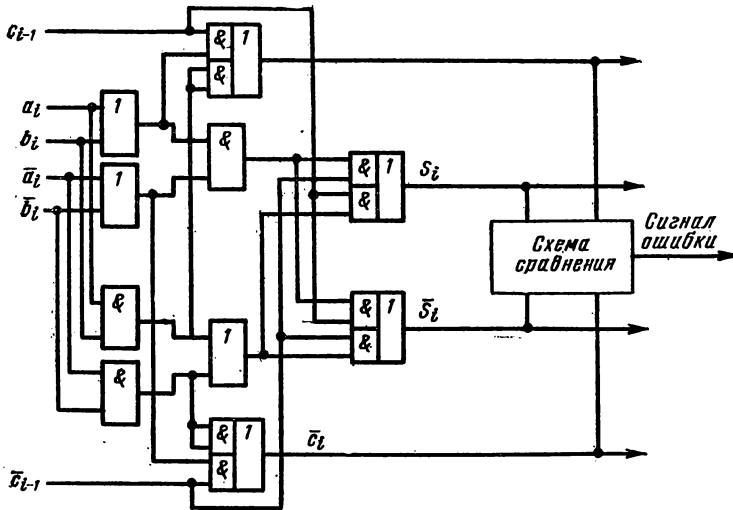


Рис. 5-21. Одноразрядный полный сумматор с парафазной логикой.

### Контроль работы счетчиков

Счетчиком называется узел ЭВМ, предназначенный для подсчета числа входных сигналов. Различают суммирующие счетчики (изменяющие состояние на  $+1$ ), вычитающие (изменяющие состояние на  $-1$ ) и реверсивные, которые под воздействием дополнительного входа могут работать как суммирующие или вычитающие.

В качестве счетчика может быть использован обычный двоичный сумматор. Связь сумматоров со счетчиками определяет общие методы контроля, применяемые для обеих схем. Применение специальных двоичных счетчиков упрощает их схемную реализацию по сравнению с полным сумматором и схемы контроля.

Наиболее распространенным методом контроля счетчиков является контроль по четности результата. Четность результата при прибавлении единицы к содержимому счетчика можно вычислить (предсказать) аналогично тому,

как это делается в сумматоре:

$$P_n = P_0 \oplus P_c \oplus 1,$$

где  $P_0$  — четность старого содержимого счетчика;  $P_c$  — четность переносов;  $P_n$  — четность нового содержимого счетчика.

Практические схемы контроля счетчиков различаются по способу определения четности возникающих переносов.

На рис. 5-22 представлена функциональная схема двоичного счетчика, контролируемого по четности. Собственно

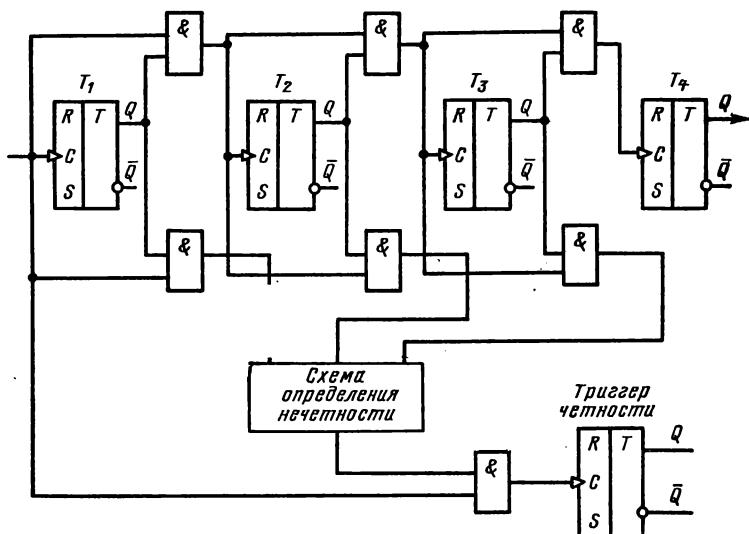


Рис. 5-22. Контроль двоичного счетчика дублированием переносов.

счетчик выполнен на  $T$ -триггерах с последовательным переносом. Контрольная схема определяет четность возникающих переносов. В данной схеме по аналогии с рассмотренной в предыдущем параграфе схемой контроля по четности сумматора применено дублирование переносов для обнаружения ошибок, вызванных неисправностями схемы формирования переносов. Напомним, что неисправность переноса вызывает четное число ошибок (в разряде переноса и в следующем разряде счетчика).

Кольцевые счетчики, или иначе сдвигающие регистры, представляют собой последовательное соединение  $n$  триггеров, причем сигналы с выхода последнего  $n$ -го триггера

поступают на вход первого триггера. Как правило, в кольцевом счетчике только один разряд находится в состоянии 1. При поступлении сигнала счета переходит в соседний разряд, данный же разряд сбрасывается в 0.

Таким образом, в кольцевом счетчике постоянно циркулирует одна единица.

Кольцевые счетчики находят широкое применение в различных узлах ЭВМ: они используются в схемах синхронизации, в схемах формирования сигналов управления, в распределителях импульсов и т. д.

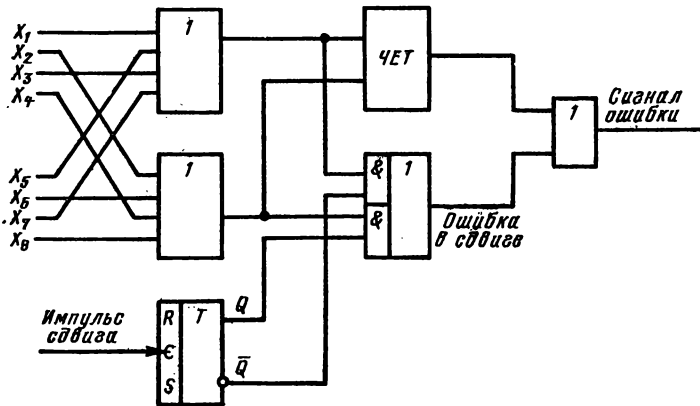


Рис. 5-23. Схема контроля восьмиразрядного кольцевого счетчика.

Существует много разновидностей кольцевых счетчиков, отличающихся схемами движения единицы от одного разряда к другому. Так, например, в реверсивных кольцевых счетчиках единица может сдвигаться как влево, так и вправо в зависимости от наличия соответствующего дополнительного сигнала управления.

Методы контроля кольцевых счетчиков основаны на использовании того факта, что если в начале счета единица имела только в одном разряде счетчика, то и при дальнейшей его нормальной работе в счетчике сохраняется единица в одном разряде. Одиночная неисправность в какой-либо цепи схемы приведет к тому, что или все разряды будут находиться в 0 (пропадание единицы), или два разряда будут находиться в 1 (лишняя единица).

На рис. 5-23 представлена одна из возможных схем контроля, основанная на методе обнаружения ошибки «1 из

*N*» входов. В данной схеме все разряды счетчика разделены на два несвязных ряда: разряды с четными номерами и разряды с нечетными номерами. Выходы каждого ряда разрядов счетчика объединяются схемой ИЛИ. При правильной работе счетчика единичный сигнал будет только на выходе одной схемы ИЛИ. Наличие неисправности в схеме счетчика, вызывающей или появление единиц на выходах двух схем ИЛИ или появление двух нулей, фиксируется схемой ЧЕТ. С другой стороны, при нормальной работе счетчика единица должна появляться попеременно на выходах двух схем ИЛИ. Неисправности, вызывающие нарушение этой последовательности, можно обнаружить второй частью схемы контроля. Основу ее составляет триггер со счетным входом, на который подаются импульсы сдвига, при этом триггер попеременно проходит состояния 0 и 1. Выходные сигналы триггера используются для коммутации выходов двух схем ИЛИ так, чтобы единица на выходе триггера соответствовала нулю на выходе схемы ИЛИ, что выполняется начальной установкой триггера в нужное состояние.

При правильной работе счетчика такое соответствие всегда будет сохраняться и сигнал ошибки при этом не формируется. Неисправности, приводящие к нарушению данного соответствия, будут вызывать появление ошибки.

Заметим, что рассмотренная схема контроля кольцевого счетчика не выявляет все ошибки: в данном случае необнаруживаемыми ошибками будут, например, ошибки в виде нескольких единиц, если соответствующие разряды подключены к одной схеме ИЛИ. Однако вероятность появления таких ошибок мала. Более вероятными ошибками являются ошибки в соседних разрядах счетчика.

Схемы контроля, применяемые для обычных двоичных счетчиков, также можно использовать для контроля кольцевых счетчиков, но при этом будут выше аппаратные затраты.

Схемы контроля необходимо защищать от возникновения ложных сигналов, связанных с различным временем установления переходных процессов в различных точках контролируемой схемы. Это осуществляется или стробированием сигнала ошибки подачей от схемы управления специального синхронизирующего сигнала, или применением сглаживающих фильтров на выходе схемы контроля.

## Контроль комбинационных логических схем

Преобразование информации в ЭВМ осуществляется логическими схемами двух типов: *цифровыми автоматами* и *комбинационными логическими схемами*.

В комбинационных логических схемах (КС) совокупность выходных сигналов  $Y = (y_1, y_2, \dots, y_m)$  в дискретный момент времени однозначно определяется совокупностью входных сигналов  $X = (x_1, \dots, x_n)$ , поступивших на вход

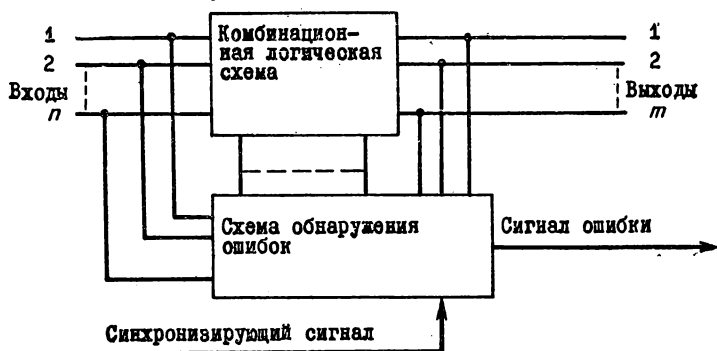


Рис. 5-24. Структурная схема обнаружения ошибок в работе комбинационных логических схем.

в тот же самый момент времени. Название «комбинационная схема» объясняется тем, что результат обработки информации зависит только от комбинации входных сигналов в данный момент времени (естественно, с учетом времени установления переходных процессов в схеме).

Функционирование КС задается либо в виде таблицы истинности, имеющей  $2^n$  строк (по числу комбинаций входных сигналов), либо в виде аналитической зависимости

$$\begin{aligned} y_1 &= f_1(x_1, \dots, x_n); \\ &\vdots \\ y_m &= f_m(x_1, \dots, x_n), \end{aligned}$$

где  $f_i$  — некоторые булевы функции.

На рис. 5-24 приведена наиболее общая структурная схема обнаружения ошибок комбинационных логических схем. Схема обнаружения ошибок комбинационных логических схем может оперировать с входными, выходными

сигналами КС, а также в ряде случаев для более полного контроля и с промежуточными сигналами, не являющимися выходными для данной КС.

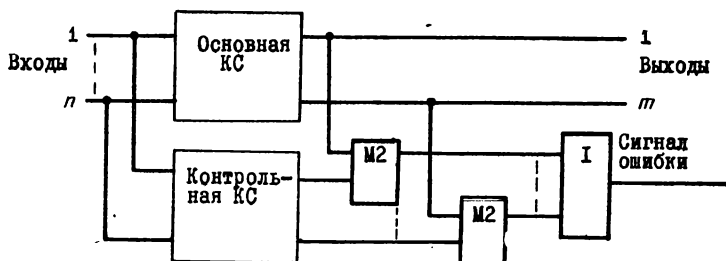


Рис. 5-25. Схема контроля комбинационной логической схемы дублированием.

В соответствии с этим имеется несколько способов построения схем контроля для комбинационных логических схем. Одним из таких способов является дублирование. При этом используются две идентичные комбинационные логические схемы с параллельными входами, выходы которых поступают на схему сравнения. При несовпа-

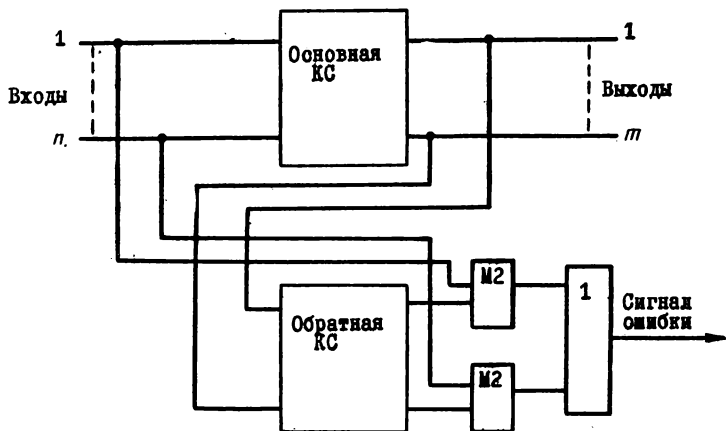


Рис. 5-26. Схема контроля комбинационной логической схемы методом восстановления входов.

дении одноименных выходных сигналов формируется сигнал ошибки (рис. 5-25).

Такой способ контроля применим для любой КС, но он обладает большой схемной избыточностью.

Второй способ контроля состоит в восстановлении входных сигналов КС по значениям выходных сигналов, Входные сигналы основ-

ной логической схемы и восстановленные входные сигналы с помощью обратной логической схемы сравниваются по модулю 2 и в случае несравнения формируется сигнал ошибки (рис. 5-26).

Данный способ применим только для таких логических функций, для которых можно определять входные сигналы, используя только выходные. Если ослабить требования к степени обнаружения ошибок, а ограничиться только обнаружением одиночных ошибок, то для реализации указанного способа контроля достаточно восстановить только одну входную переменную  $x_1$ .

Т а б л и ц а 5-5

$x_1$	$x_2$	$x_3$	$y_1$	$y_2$
0	0	0	1	0
0	0	1	1	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

При этом любая неисправность в основной логической схеме будет вызывать ошибку на одном из выходов  $y_j$ , которая, поступив на обратную логическую схему, вызовет изменение в  $x_i$ , что будет зафиксировано схемой сравнения. Достоинством такого способа контроля являются меньшие аппаратные затраты по сравнению с дублированием.

В качестве примера рассмотрим комбинационную логическую схему, функционирование которой задается следующей таблицей истинности (табл. 5-5).

Для данной КС в качестве восстанавливаемого входа нельзя выбрать  $x_1$ , поскольку значения  $x_1$  различны при одних и тех же значениях выходов  $y_1$  и  $y_2$  (при значениях выходов  $y_1$  и  $y_2$ , равных 10 — первая и шестая строка таблиц значения  $x_1$  различны). По тем же соображениям неоднозначности обратной функции нельзя применить в качестве восстанавливаемого входа  $x_3$  (первая и вторая строка таблицы).

Для входа  $x_2$  условие однозначности соблюдается. На основе табл. 5-5 можно построить таблицу истинности для обратной логической схемы (табл. 5-6).

В соответствии с данной таблицей обратная логическая схема описывается выражением

$$x_2^* = \bar{y}_1 \bar{y}_2 \vee y_1 y_2 = (\bar{y}_1 \oplus \bar{y}_2),$$

что соответствует схеме равнозначности.

Т а б л и ц а 5-6

$y_1$	$y_2$	$x_2^*$
0	0	1
0	1	0
1	0	0
1	1	1

На рис. 5-27 представлена схема контроля комбинационной логической схемы, соответствующая таблице истинности (табл. 5-6).

Контроль комбинационной логической схемы может быть также осуществлен только по выходным сигналам (рис. 5-28).

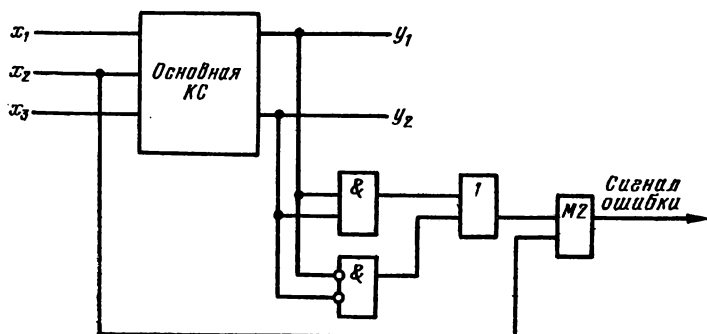


Рис. 5-27. Схема контроля комбинационной логической схемы.

Возможность такого способа контроля рассмотрим на примере полного дешифратора трехразрядного двоичного кода.

Особенностью работы дешифратора является наличие единицы только на одном выходе (табл. 5-7) при любой комбинации входных сигналов. Учет этого факта позволяет



Рис. 5-28. Контроль комбинационной логической схемы путем анализа выходов.

построить достаточно экономичную схему контроля, которая будет вырабатывать сигнал ошибки во всех случаях появления в результате неисправности схемы на всех выходах дешифратора нулей или нескольких единиц. Такая



схема контроля будет выявлять все ошибки, за исключением тех, в результате которых неправильно выбирается нужный выход.

Таблица 5-7

$x_3$	$x_2$	$x_1$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$	$y_8$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Представленная на рис. 5-29 структурная схема контроля трехразрядного дешифратора представляет собой пороговую схему «1 из 8» и состоит из двух частей: схема ИЛИ выявляет случай отсутствия единиц на выходе дешифратора,

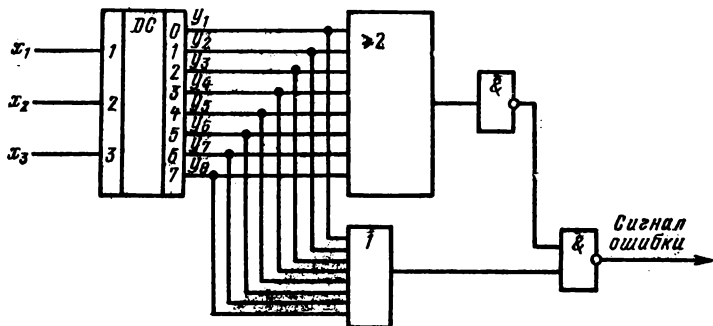


Рис. 5-29. Контроль полного трехразрядного дешифратора по выходным сигналам.

пороговая схема с порогом 2 анализирует случай, когда на выходе будет более одной единицы. Сигналы с обеих частей схемы контроля объединяются и, таким образом, формируется сигнал ошибки.

### Контроль запоминающих устройств

Комплекс запоминающих устройств ЭВМ предназначен для записи, хранения и выдачи информации. В общем слу-

чае он может содержать: сверхоперативное ЗУ (СОЗУ), ОЗУ, внешнее ЗУ (ВЗУ) с произвольным обращением (магнитные барабаны, магнитные диски) и внешние ЗУ с последовательным поиском информации (ЗУ на магнитных лентах).

Обнаружение ошибок в ЗУ включает задачи: 1) проверку правильности данных, записываемых в ЗУ; 2) обнаружение ошибок при чтении информации из ЗУ; 3) проверку правильности обращения по заданному адресу; 4) проверку правильности работы аппаратуры ЗУ.

Из многих типов ОЗУ наибольшее распространение в современных ЭВМ получили запоминающие устройства на магнитных сердечниках (МОЗУ) и запоминающие устройства на интегральных схемах с использованием триггеров в качестве запоминающих элементов.

МОЗУ состоит из куба памяти, аппаратуры адресной части и аппаратуры разрядной части. Большую часть электронного оборудования составляет аппаратура адресной части (дешифратор адреса и выходные адресные формирователи). Как правило, оконечная ступень дешифратора выполняется в виде диодной сетки, насчитывающей несколько тысяч диодов. Кроме того, активные элементы в выходных цепях адресной части используются в тяжелых режимах. Это все обуславливает большую вероятность появления неисправностей в адресной части МОЗУ.

Для аппаратурного контроля адресной части МОЗУ можно применить способ восстановления входных переменных, описанный ранее. Применительно к МОЗУ его реализация основана на использовании дополнительного шифратора, преобразующего позиционный код на выходе адресной части в двоичный код. Полученный таким образом код сравнивается с исходным кодом адреса и делается заключение о наличии или отсутствии ошибки.

С целью уменьшения дополнительного контрольного оборудования используется не полный код адреса, а какой-либо контрольный код (код четности или код по модулю), однако необходимо иметь в виду, что при этом вероятность обнаружения ошибки уменьшается.

Можно построить схему контроля, обнаруживающую одиночную ошибку адресной части, учитывая то, что при одиночной неисправности (исключая неисправности регистра адреса) может оказаться или два выбранных слова, или не одного. Для того чтобы был выбран один, но неправильный адрес, должна произойти двойная ошибка. Так

как вероятность этого мала, схема контроля достаточно эффективна.

Для преобразования позиционного кода в двоичный используется шифратор, состоящий из схем объединения, по одной на каждый разряд адреса. Входы каждой схемы объединения (схемы ИЛИ) подключаются к тем адресам, которые в данном разряде имеют 1 в коде адреса. Код, обратный коду адреса, может быть получен, если входы шифратора подключить к тем адресам, которые имеют 0 в соответствующем разряде кода адреса.

Такие два шифратора могут быть использованы для контроля адресной части, так как только при правильной работе коды на их выходах будут взаимно обратны. В случае, если не выбран ни один адрес, на всех выходах обоих шифраторов будет код 0. В случае, если выбрано одновременно два адреса, на одном шифраторе получится результат логического сложения прямых кодов, а на другом — результат логического сложения обратных кодов. При этом коды на выходах шифраторов не будут взаимно обратными, что свидетельствует о наличии ошибки.

Если шифратор выполнен на магнитных сердечниках, то отпадает необходимость в двух шифраторах. Прошивка выходных обмоток шифратора может быть сделана таким образом, чтобы коды 0 и 1 отличались полярностью сигналов на этих обмотках. Сигналы с выходных обмоток шифратора подаются на усилитель, одинаково реагирующий на сигналы любой полярности. При правильной работе адресной части на выходах усилителей всех разрядов должны быть сигналы. Выбор одновременно двух адресов приведет к тому, что в тех разрядах, в которых коды этих адресов не совпадают, будут сигналы обеих полярностей. Эти сигналы скомпенсируются, и на выходе усилителя сигнала не будет.

Отсутствие сигнала на выходе хотя бы одного усилителя свидетельствует о неисправной работе адресной части. Для надежной работы такой схемы контроля разброс сигналов с сердечников шифратора должен быть небольшим.

Если вместо полного кода адреса использовать контрольный код, то так же, как и в схемах контроля, использующих исходный адрес, может быть получено значительное сокращение оборудования.

Так, при использовании кода четности шифратор должен быть одноразрядным, причем адреса, имеющие четное число единиц в коде, должны давать сигнал одной полярности, а нечетное — другой.

Если выбраны одновременно два адреса с разными кодами четности или не выбран ни один, то на выходе такого шифратора сигнала не будет, тем самым фиксируется ошибка в адресной части.

Наиболее вероятной причиной выбора одновременно двух адресов являются неисправности в дешифраторе адреса, при этом в случае одиночной ошибки коды выбранных адресов будут отличаться только одним разрядом и, следовательно, иметь несовпадающие коды четности.

Другой вид неисправности адресной части состоит в том, что адресный формирователь выдает импульсы неправильной амплитуды, что может привести к ошибке в считывании информации. Такого типа неисправность обнаруживается с помощью схемы, чувствительной к изменению амплитуды выходного сигнала.

Один из способов осуществления такой схемы контроля состоит в использовании двух магнитных сердечников, работающих как пороговые детекторы. Все шины возбуждателей прошивают оба сердечника. Кроме того, через каждый сердечник пропущен провод с током смещения: для одного из сердечников ток смещения выбирается больше тока возбуждения на пороговое значение, а для второго сердечника — на это же значение меньше тока возбуждения. Уровень порога выбирается таким образом, чтобы сердечники не перемагничивались, когда ток возбуждения лежит в установленных пределах. При выходе амплитуды тока возбуждения за установленные пределы один из сердечников перемагничивается, что фиксируется как неисправность.

Неисправности в разрядной части магнитных запоминающих устройств обнаруживаются, как правило, с использованием кода четности информации. Если код слова нечетный, то этот контроль будет также обнаруживать отказ в выборе слова вообще.

Контроль соответствия информации в памяти записываемой информации может быть осуществлен с помощью усилителей считывания при формировании сигналов при записи во время перемагничивания сердечника.

Ошибки, возникающие в работе внешних магнитных запоминающих устройств (НМЛ, НМБ, НМД), могут носить случайный или систематический характер. Случайные ошибки возникают в основном в результате мгновенной потери контакта магнитного носителя с магнитной головкой. Как правило, при повторении операции считывания эти ошибки устраняются.

Систематические ошибки обычно возникают в результате механического повреждения магнитного носителя. Постоянные дефекты магнитного носителя, вызванные нарушением технологии его изготовления, приводят к тому, что невосстанавливаемо теряется информация.

В процессе длительной эксплуатации магнитного носителя вследствие износа магнитного слоя, его отслаивания, ухудшения физико-химических свойств подложки увеличивается вероятность систематических ошибок.

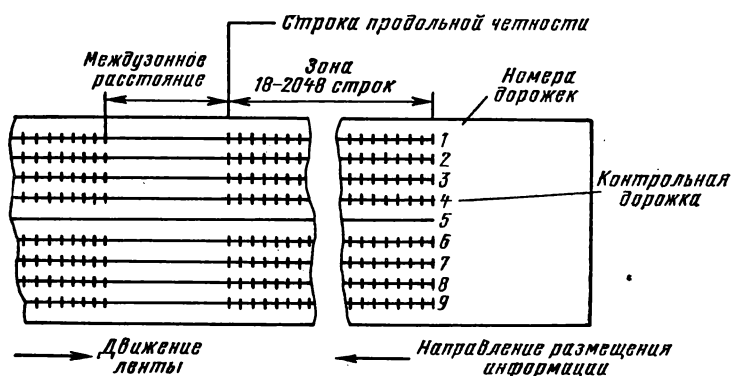


Рис. 5-30. Размещение информации на магнитной ленте.

Частой причиной ошибок при считывании является попадание пыли на магнитную поверхность. При плотностях записи, применяемых в настоящее время, размеры пылинок оказываются намного больше участка, занимаемого одним двоичным разрядом хранимой информации. Поэтому в этих устройствах относительно более вероятны групповые ошибки в смежных разрядах, т. е. так называемые «пачки» ошибок, подобно тому, как это имеет место в каналах связи за счет различного рода помех.

При контроле считывания информации с магнитной ленты помимо проверки на четность применяют дублирование информационных и служебных дорожек, а сама проверка на четность производится в двух измерениях (рис. 5-30).

Сущность этого метода контроля состоит в том, что каждое слово записывается на ленту параллельно-последова-

тельным способом. По ширине ленты размещается строка, состоящая из восьми двоичных знаков, включая один знак проверки на четность (контрольная дорожка). В конце слова или нескольких слов записывается контрольная строка, состоящая из знаков проверки на четность по столбцам, т. е. по длине каждой магнитной дорожки. Необходимое при этом усложнение кодирующей и декодирующей аппаратуры компенсируется значительным повышением эффективности контроля. Во-первых, кодирование в строках производится с большой избыточностью в силу того, что строка короткая. Во-вторых, вследствие двойной проверки необнаруженными остаются только такие ошибки, которые составляют четное число как в строках, так и в столбцах, что маловероятно.

Обнаружение ошибок при записи информации на магнитный носитель осуществляется в основном двумя методами — методом эхо-контроля или методом чтения после записи.

Эхо-контроль основан на том, что при записи информации любого символа хотя бы один усилитель записи подает на магнитную головку записи сигнал 1. Применяв такой метод контроля, можно выяснить, имел ли место процесс записи в данный момент времени.

Для контроля записи методом чтения информации после записи применяют sdвоенный блок головок (головка записи и головка считывания). Этот вид контроля, кроме того, может проверить наличие или отсутствие дефектов в области записи данных.

В запоминающих устройствах на магнитных дисках и барабанах необходимо еще контролировать правильность выбора дорожек. Это осуществляется с помощью механических и электрических обратных связей с механизмов выбора дорожек. Другой способ заключается в том, что каждая считываемая зона имеет свой опознавательный знак. При неправильном выборе дорожки опознавательный знак не будет соответствовать знаку, ожидаемому системой.

Так как на магнитных дисках и барабанах данные записываются последовательным кодом, то эффективно использовать для обнаружения ошибок циклические коды, рассмотренные в гл. 5, которые кодируются и декодируются при помощи сдвиговых регистров.

## ГЛАВА ШЕСТАЯ

### ПРОВЕРОЧНЫЕ (КОНТРОЛИРУЮЩИЕ) ТЕСТЫ

Все возникающие в ЭВМ ошибки проявляются в виде отказов (систематических ошибок) и случайных ошибок (сбоев) (см. гл. 2).

При отказе происходит полная или частичная утрата системой (или ее отдельным элементом) работоспособности из-за выхода за допустимые пределы значений одного или нескольких параметров этой системы. При сбое происходит кратковременное нарушение нормальной работы ЭВМ, при котором один или несколько ее элементов срабатывают неправильно, причем после сбоя эти элементы могут продолжительное время функционировать нормально.

Для выявления элементов, приведших к отказу, и элементов, параметры которых близки к граничным, применяются аппаратные, программные и аппаратурно-программные методы контроля. Для сигнализации о происшедшем сбое используются обычно методы программного контроля, которые будут рассмотрены в этой и следующей главах книги.

#### 6-1. ПРОГРАММНО-ЛОГИЧЕСКИЙ КОНТРОЛЬ

Программно-логический контроль служит для проверки правильности решения задачи при выполнении рабочей программы. Основным преимуществом этого метода контроля является возможность обнаружить ошибки выполнения программы, обусловленные сбоем. Обычно программно-логический контроль применяется для проверки решения задачи в целом или ее отдельных достаточно самостоятельных этапов. Наиболее известными способами программно-логического контроля являются: двойной счет, введение математических и смысловых проверок и т. д.

Контроль методом двойного счета (на современных универсальных ЭВМ применяется довольно редко) основан на двукратном решении одной и той же задачи. Результаты решений сравниваются, и их совпадение считается признаком достоверности. В зависимости от типа задач применяют различные способы сравнения результатов решения.

Для статистических и экономических задач, где результат обычно получается в виде таблиц, подсчитываются контрольные суммы при каждом варианте просчета и контроль

ведется по сравнению этих сумм. Метод двойного счета прост в исполнении, но требует больших затрат машинного времени. Однако, если применение метода двойного счета для контроля решений задач в современных универсальных ЭВМ невыгодно из-за больших затрат машинного времени и высокой его стоимости, то применение этого метода весьма эффективно для контроля работы устройства и программы ввода данных с колоды перфокарт. Контроль при этом ведется по контрольной сумме.

Введение в программу математических и логических проверок промежуточных результатов позволяет в значительной мере сэкономить машинное время по сравнению с методом двойного счета, но требует значительных предварительных затрат времени математиком или программистом на анализ алгоритма решения и формулировку дополнительных математических и логических проверочных условий.

Рассмотрим конкретный пример.

**Пример 6-1.** Составим блок-схему алгоритма для отыскания начальных приближений (отделения) всех положительных действительных корней уравнения  $f(x) = 3x^8 - 5x^7 - 6x^6 - x - 9 = 0$  с точностью до 0,1.

Отыскание начальных приближений корней уравнения выполняется с тем, чтобы в дальнейшем уточнить их значения одним из существующих методов. Для этого предварительно находится верхняя и нижняя границы нахождения всех положительных действительных корней.

Для алгебраического уравнения с действительными коэффициентами  $a_0, a_1, a_2, \dots, a_n$  ( $a_0 > 0$ ) верхняя граница положительных действительных корней  $R_B^+$  определяется по Лагранжу как

$$R_B^+ = 1 + \sqrt[k]{B/a_0},$$

где  $k$  — номер первого из отрицательных коэффициентов полинома  $f(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_n$ ;  $B$  — наибольшее из абсолютных значений отрицательных коэффициентов  $f(x)$  (для нашего случая при определении  $R_B^+ k = 1, B = 9, a_0 = 3$ ), а нижняя граница положительных действительных корней  $R_H^+$  находится по той же формуле, что и  $R_B^+$ , но значения  $k, B$  и  $a_0$  берутся из вспомогательного уравнения

$$f_1(x) = x^n f(1/x) = 0.$$

Находим  $f_1(x) = 9x^8 + x^7 + 6x^6 + 5x - 3 = 0$  (отсюда для  $R_H^+$  имеем  $k = 8, B = 3, a_0 = 9$ ).

Решение поставленной задачи начинается с вычисления  $f(x)$  в точке  $x = R_B^+ = 1 + 9/3 = 4$  и продолжается до  $R_H^+ = 1 + \sqrt[8]{3/9} = 1,87$  с шагом, равным допустимой ошибке отделения корня:  $n = 0,1$ . Нахождение корня  $z$  в интервале  $(x, x + n)$  проверяется по условию изменения знака произведения  $fp$ , где  $f$  — значение функции  $f(x)$  в точке  $x$ , а  $p$  — значение функции  $f(x)$  в точке  $x + n$ .



Блок-схема алгоритма приведена на рис. 6-1. В алгоритме блоки 2—7 и 11 служат для организации вычислений и печати всех положительных корней уравнения. Заметим, что формула Лагранжа, определяющая верхнюю и нижнюю границы действительных корней, не гарантирует обязательного нахождения корня (в уравнении может не быть ни одного действительного положительного корня), в этом случае алгоритм выведет на печать значение верхней границы  $R_в^+$ .

Блоки 8—10 и 12 служат для организации контроля решения задачи методом математической проверки и печати фразы «Проверь решение». Так как в алгоритме ищутся корни уравнения, то проверкой отсутствия сбоя или отказа является подстановка каждого найденного корня в уравнение. При этом поскольку корни отыскиваются с некоторой ошибкой, то и уравнение проверяется с заданной точностью  $\epsilon$ . Если же условие  $f \leq \epsilon$  не выполняется для заданного значения корня  $z$ , то печатается фраза «Проверь решение». После этого решение надо повторить. Если повторное решение пройдет нормально, то был сбой, если же после повторного решения вновь будет напечатана эта же фраза, значит, в ЭВМ имеется отказ (предполагается, что алгоритм решения отлажен и выход на печать фразы не обусловлен ошибками алгоритма).

Контроль методом математической или смысловой проверки требует значительно меньших затрат машинного времени, чем двойной счет, и позволяет выявить не только случайные ошибки, но и постоянные. Однако такая форма контроля требует больших предварительных затрат времени и зависит от характера задачи и искусства программиста.

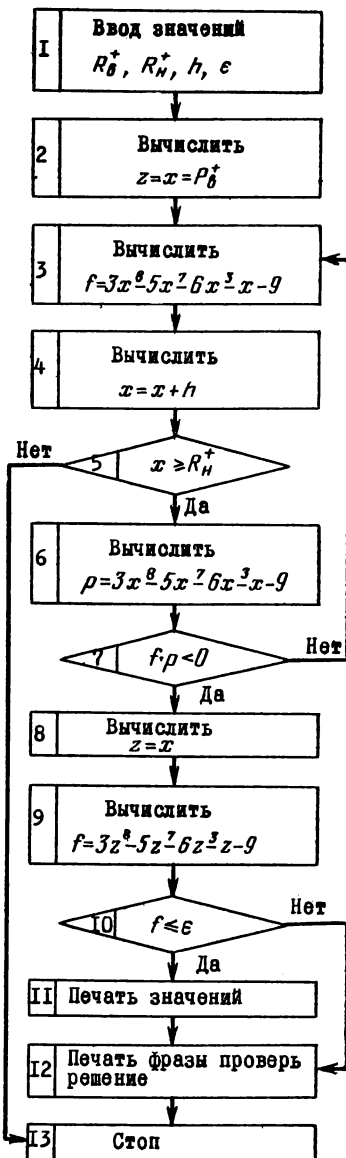


Рис. 6-1. Блок-схема алгоритма вычисления начального приближения положительных действительных корней алгебраического выражения

## 6.2. ТЕСТОВЫЙ КОНТРОЛЬ

Основным видом контроля, позволяющим не только обнаружить отказ ЭВМ в ходе ее эксплуатации, но и проверить работоспособность машины в целом и отдельных ее функциональных блоков в период наладки и после ремонта, является тестовый контроль, осуществляемый специальными программами, называемыми тестами. Идея проведения тестового контроля сводится к выполнению вычислительной машиной определенной последовательности действий над исходными числами, сравнению полученных результатов с подготовленными ответами и к регистрации ошибки в случае их несовпадения.

По назначению тестовые программы делятся на проверочные (контролирующие) и диагностические.

Проверочные тесты предназначены для установления факта наличия неисправности в машине (или ее отдельном устройстве), диагностические — для определения и указания конкретного модуля (или элемента), в котором произошел отказ. В отличие от программно-логического тестовый контроль проводится не в процессе выполнения машинной основных программ, а до или в перерывах между их выполнением.

Тестовые программы обычно записываются на машинном языке или на языке АССЕМБЛЕР и хранятся во внешних запоминающих устройствах ЭВМ (чаще всего в накопителях на магнитной ленте), однако практически всегда имеются дубликаты тестовых программ, записанные на перфокартах или перфолентах для ввода при первоначальной наладке машины либо при отказе внешних запоминающих устройств.

Для оценки эффективности контроля ЭВМ с помощью тестовых программ предложены следующие основные критерии (характеристики).

*Полнота охвата*, представляющая собой вероятность определения того элемента, из-за неисправности которого произошел отказ его или связанных с ним элементов. Полнота охвата (машины в целом или отдельного устройства) вычисляется по формуле

$$L = \frac{m}{n}, \quad (6-1)$$

где  $m$  — число ошибок, определяемых данным тестом;  $n$  — число всех возможных ошибок в машине или ее отдельном устройстве.

*Глубина контроля* служит для оценки вероятности обнаружения неисправности и вычисляется так

$$R = \frac{S}{M}; \quad (6-2)$$

где  $S$  — число физических неисправностей, выявляемых тестом;  $M$  — число физических неисправностей, которые могут вызвать любую ошибку из числа  $m$  ошибок, определяемых данным тестом. Отметим, что число физических неисправностей обычно не совпадает с числом ошибок, так как одна физическая неисправность (электрическая или механическая неисправность какого-либо элемента) может вызвать несколько ошибок как в самом неисправном элементе, так и в связанных с ним.

*Эффективность контроля*, представляющая собой вероятность обнаружения тестом любой физической неисправности при всех возможных в машине или отдельном ее устройстве физических неисправностях:

$$E = \frac{S}{N}, \quad (6-3)$$

где  $N$  — число всех возможных физических неисправностей в контролируемом устройстве (или в ЭВМ в целом).

Ряд характеристик служит не для оценки качества системы контроля, а для непосредственной оценки тестовой программы. Прежде чем дать формулировки этим характеристикам, остановимся на составе команд в тестовой программе. Все команды в тестовой программе можно разбить на две группы: основные и вспомогательные.

*Основными* называются команды, при выполнении которых срабатывают те или иные элементы контролируемого устройства и на основе результатов выполнения этих команд выявляется неисправность. Все остальные команды в программе называются *вспомогательными*. В идеальном варианте тестовая программа какого-либо устройства должна была бы включать в себя только основные команды, причем каждая основная команда позволяла бы проверить возможную неисправность одного элемента. В этом случае такой тест занимал бы минимальное место в памяти ЭВМ и поиск неисправности длился бы минимальное время. Однако очевидно, что подобную идеальную тестовую программу создать нельзя.

*Коэффициент полезного действия теста* оценивается как

$$k = \frac{t}{T}, \quad (6-4)$$

где  $t$  — время выполнения основных команд теста;  $T$  — время выполнения всех его команд.

*Относительное быстроедействие тестовой программы* показывает, сколько различных неисправностей определяется ею в единицу времени:

$$B = \frac{S}{T}. \quad (6-5)$$

*Надежность тестовой программы* представляет собой вероятность того, что имеющаяся в машине неисправность не повлияет на выполнение вспомогательных команд программы и на работу тех элементов, которые не принадлежат контролируемой части устройства, и вычисляется по формуле

$$H = \frac{b}{N}, \quad (6-6)$$

где  $b$  — число неисправностей в контролируемом устройстве, не влияющих на работу вспомогательных команд программы и на работу элементов, не принадлежащих контролируемой части устройства.

Надежность тестовой программы всегда меньше единицы. Для увеличения надежности всей системы контроля применяют не набор тестов, а их систему.

*Системой тестов*, служащих для наладки или проверки ЭВМ, называют набор тестовых программ, позволяющий (при определенном порядке их выполнения и устранении неисправностей по мере их нахождения) провести практически полную проверку элементов машины за приемлемое время.

Для того чтобы система тестов имела высокую надежность, необходимо, чтобы каждая тестовая программа имела высокую условную надежность. *Условная надежность* определяется соотношением

$$H' = \frac{b'}{N},$$

где  $b'$  — число неисправностей, которые не влияют на выполнение вспомогательных команд программы и на работу элементов, действующих при выполнении основных команд, но не входящих в проверяемую часть устройства при условии исправления всех неисправностей, выявленных предыдущими тестами, т. е.

$$b' = b - G,$$

где  $G$  — число устраненных неисправностей.

### 6-3. ФОРМИРОВАНИЕ СИСТЕМЫ ТЕСТОВ

Выше отмечалось, что все тесты делятся на две большие группы: тесты проверочные и диагностические. В современных системах проверочным тестам передана часть функций диагностики. Обычно система проверочных тестов строится так, чтобы выполнить принцип *расширяющихся областей*, а именно:

ранее проверенные элементы используются при работе последующих подпрограмм; при этом допустимый режим их использования определяется той глубиной контроля, с которой они были проверены;

ранее проверенные элементы проверяются с большей глубиной;

контролю последовательно подвергаются все новые группы элементов, связанных с уже проверенными.

Так как диагностические тесты в отличие от проверочных должны сузить район поиска и локализовать неисправность, то система диагностических тестов использует принцип *сужающихся областей*. Входной информацией для системы диагностических тестов является сигнал об ошибке и дополнительная информация об отказе, сообщаемая проверочной программой и оператором ЭВМ на основе данных проверочных программ.

Рассмотрим на некотором условном примере, как формируется система тестов для проверки ЭВМ.

**Пример 6-2.** Необходимо сформировать последовательность тестов, позволяющих на основе приведенных выше критериев осуществить эффективную полную проверку ЭВМ. Пусть в результате предварительных расчетов для данной машины получено количество возможных ошибок ОЗУ, АУ и УУ:  $n_{\text{ОЗУ}} \approx 4,6 \cdot 10^6$ ;  $n_{\text{АУ}} \approx 1 \cdot 10^4$ ;  $n_{\text{УУ}} \approx 1,3 \cdot 10^4$  соответственно.

Последовательность тестов будем выбирать так, чтобы обеспечить достаточно высокую условную надежность всех программ системы.

Обозначим через  $i = 1, 2, 3, \dots$  номера тестовых программ в системе. Предположим далее, что в системе имеется некоторая программа (обозначим ее номером  $i = 1$ ), позволяющая проверить все ячейки ОЗУ с полной охвата  $L_{\text{ОЗУ}} = 1$  (в принципе эта первая проверка может быть выполнена и специальной схемой). Пусть эта проверка обнаружила  $\Delta b_1$  неисправностей, тогда для программы  $i$  можно найти условную надежность как

$$H'_{i1} = \frac{\Delta b_1 + b_i}{N} > \frac{b_i}{N}. \quad (6-7)$$

Здесь  $H_{i1}$  — условная надежность выполнения программы  $i$  при условии, что была выполнена 1-я программа.

Найдем  $\Delta b_1$  как

$$\Delta b_1 = \Delta b_{OЗУ} = l_1^{OЗУ} n_{OЗУ} \geq l_1^{OЗУ} \cdot 2,5 \cdot 10^6.$$

Здесь  $l_1^{OЗУ}$  — некоторый коэффициент, учитывающий число неисправностей, вызывающих ошибки (из общего числа  $n_{OЗУ}$ ) при первой проверке ( $l_1^{OЗУ} > 1$ ).

Легко показать, что для заданных конкретных условий более эффективно проведение первой проверки начинать с ОЗУ, чем с других устройств процессора. В самом деле, в этом случае

$$\Delta b_1 = \Delta b_{AY} \geq l_1^{AY} \cdot 10^4;$$

$$\Delta b_1 = \Delta b_{YУ} \geq l_1^{YУ} \cdot 1,3 \cdot 10^4,$$

где  $\Delta b_{AY}$  и  $\Delta b_{YУ}$  — число неисправностей АУ и УУ.

Если считать, что коэффициенты  $l_1^{AY}$ ,  $l_1^{YУ}$  аналогичны коэффициенту  $l_1^{OЗУ}$  — примерно одного с ним порядка, то  $\Delta b_{OЗУ} > \Delta b_{AY}$  и  $\Delta b_{OЗУ} > \Delta b_{YУ}$ . Таким образом, доказано, что программа 1 (или специальная схема) должна проверять ОЗУ.

Остановимся далее на выборе программы 2 ( $i = 2$ ). В любой из программ, выполняющихся с выдачей команд из ОЗУ, принимают участие такие блоки УУ, как регистр команд, счетчик команд и дешифратор УУ, но проверка этих блоков невозможна без одновременной проверки управляемых дешифратором элементов, входящих в состав АУ. Поэтому эта программа с  $i = 2$  обычно называется программой частичной проверки АУ.

Число неисправностей УУ, определяемых этой программой, находится как

$$\Delta b_2^{YУ} = l_2^{YУ} n'_{YУ}, \quad (6-8)$$

где  $n'_{YУ}$  — число ошибок УУ, проверяемых программой. При этом  $n'_{YУ} < n_{YУ}$ , так как  $L_{YУ} < 1$ . Коэффициент  $l_2^{YУ}$  учитывает число неисправностей, вызывающих ошибки из числа  $n'_{YУ}$  при второй проверке.

Выбор программ 3 и 4 организован таким образом, чтобы провести полный контроль регистра и счетчика команд устройства управления. Для этого используются тяжелые режимы работы ЭВМ (подробно режимы работы будут рассмотрены в § 6-4). После выполнения полного контроля устройства управления  $L_{YУ}$  и  $R_{YУ}$  будут равны единице.

Пятая программа должна выполнить полную проверку АУ процессора. После выполнения программы 5

$$L_{AY} = 1 \text{ и } R_{AY} = 1.$$

Программы 6 и 7 служат для полной проверки ОЗУ и каналов. Программа 8-я проверяет работу всех внешних устройств и, наконец, программа 9 проводит полный комплексный контроль ЭВМ.

#### 6-4. СИСТЕМА ТЕСТОВ ТЕСТ-МОНИТОР ЕДИНОЙ СИСТЕМЫ ЭВМ

Как уже отмечалось, тестовые программы представляют собой набор специально подобранных примеров и эталонных ответов их решения. Проверка всей машины в целом состоит из последовательных проверок каждого функционального устройства (процессора с ОЗУ, ВЗУ, устройств ввода-вывода и т. д.). Для каждого устройства имеется свой тест или свой набор тестов. Управление запуском тестов и выводом на контроль (на пульт управления или на устройства печати) результатов проверки осуществляется управляющей программой. На рис. 6-2 приведена укрупненная структурная схема системы тестов ТЕСТ-монитор Единой системы (ЕС) ЭВМ.

Комплект тестовых программ системы ТЕСТ-монитор предназначен для проверки правильности функционирования технических средств ЕС ЭВМ и включает в себя:

- базовый тест;
- управляющую программу;
- программу обслуживания тестовой магнитной ленты;
- тестовые программы (тест-секции).

*Базовый тест* предназначен для проверки той части оборудования, которая необходима для работы управляющей программы ТЕСТ-монитора. Базовый тест проверяет правильность выполнения команд, используемых управляющей программой, работоспособность пишущей машинки, на которой будут печататься результаты проверки, и накопителя на магнитной ленте, на котором установлена тестовая магнитная лента.

*Программа обслуживания тестовой ленты* выполняет следующие функции:

- формирование тестовой ленты;
- дублирование тестовой магнитной ленты;
- внесение изменений (добавление, исключение, замена тестов) на тестовой магнитной ленте;
- вывод программы с тестовой магнитной ленты на перфокарты;

распечатка списка программ, находящихся на тестовой магнитной ленте, и текста самих программ.

*Управляющая программа* ТЕСТ-монитора выполняет следующие функции:

загружает (записывает) тест-секции с магнитной ленты в ОЗУ. Загрузка тест-секций осуществляется последова-

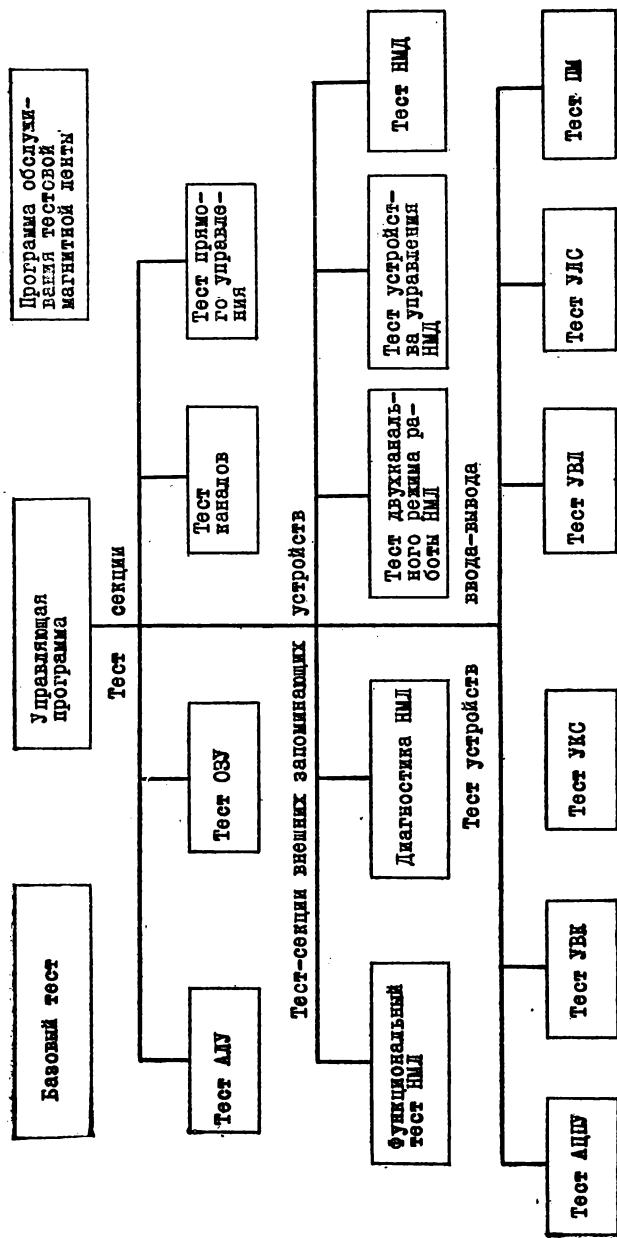


Рис. 6-2. Структурная схема ТЕСТ-монитора.

АЛУ — арифметическо-логическое устройство; НМД — накопитель на магнитной ленте; НМД — накопитель на магнитных дисках; УВК — устройство вывода на перфокарты; АЦПУ — алфавитно-цифровое печатающее устройство; УКС — устройство считывания (ввода) с перфокарт; УЛС — устройство считывания с перфоленты; ПМ — пишущая машинка; УВД — устройство вывода на перфоленту.



тельно, т. е. вначале выполняется одна тест-секция, а затем загружается другая;

предоставляет тест-секциям данные о конкретной ЭВМ и проверяемых устройствах ввода-вывода.

Для этого управляющая программа использует ряд таблиц. Основная связь между управляющей программой и тест-секциями осуществляется с помощью специальной таблицы. В эту таблицу заносятся данные, характеризующие конкретную ЭВМ и особенности ее устройства и состава оборудования. Эта таблица записана, как и все программы, на тестовой магнитной ленте и сведения в нее вносятся на заводе при формировании системы ТЕСТ-монитор для конкретной ЭВМ.

В таблице помещается следующее:

тип ЭВМ (система может быть использована для ЭВМ ЕС-1020, ЕС-1022, ЕС-1030, ЕС-1035, ЕС-1040, ЕС-1050, ЕС-1060);

емкость основной памяти (8, 16, 32, 64, 128, 256, 512, 1024 Кбайт); отметим, что в настоящее время ЭВМ с объемом памяти 8, 16, 32, 64 и 128 Кбайт практически не выпускаются;

быстродействие основной памяти (от 10 нс до 2,55 мкс) и ряд сведений, характеризующих управляющую программу данной ЭВМ.

Для описания характеристик устройств ввода-вывода, входящих в состав конкретной машины, используется соответствующая таблица, включающая следующие сведения:

тип устройства ввода-вывода;

дополнительные функции, выполняемые отдельными устройствами (например, наличие блока чтения без преобразования в устройстве чтения с перфокарт, наличие универсального набора символов в алфавитно-цифровом печатающем устройстве и т. д.).

Всего в состав управляющей программы входит шесть таблиц, с помощью которых полностью описывается состав данной ЭВМ, ее основные характеристики и особенности отдельных устройств, входящих в комплект, и самого ТЕСТ-монитора. При изменении состава устройств машины, замене отдельных устройств или добавлении новых в таблицы вносятся соответствующие изменения, которые

обеспечивают тест-секции стандартным программным обслуживанием;

выдают оператору сообщения о ходе проверки;

передают указания оператора отдельным тест-секциям или их группе.

*Тест-секции* служат для проверки различных устройств. При этом для проверки каждого устройства используется несколько тест-секций.

Каждая тест-секция состоит из заглавия и серии логически независимых подпрограмм, называемых примерами. Перед каждым примером записано его заглавие. Заглавие теста включает в себя код (идентификатор) тест-секции, переключатель прерываний и справочную информацию, которая используется для организации взаимодействия между управляющей программой и данной тест-секцией.

Все тест-секции ТЕСТ-монитора перечислены в табл. П-1. В перечне приведены наименование теста, код теста и поисковый номер. Код тест-секции состоит из пяти шестнадцатиричных цифр — TNNNS, где T — признак, характеризующий возможность выполнять данную тест-секцию на различных ЕС ЭВМ. Если данная тест-секция может выполняться на всех ЕС ЭВМ, то она имеет признак F, если только на некоторых — признак E; NNN — трехзначный номер тест-секции, S — номер редакции данного теста, изменяемый при замене теста.

Например, тест с номером F6010 может быть использован на любой ЕС ЭВМ; он имеет номер 601 и нулевую редакцию.

Для поиска нужной тест-секции при задании оператором определенного приказа управляющей программе используются поисковые номера тест-секций, состоящие из трех шестнадцатиричных цифр.

Программы ТЕСТ-монитора расположены на тестовой магнитной ленте в следующем порядке:

- базовый тест;
- управляющая программа;
- программа обслуживания тестовой магнитной ленты;
- тест-секции.

Первые три программы не связаны программно друг с другом и могут работать самостоятельно. Тест-секции и управляющая программа тесно взаимосвязаны, и тест-секции без управляющей программы работать не могут.

Система ТЕСТ-монитор автоматически, если нет специальных заданий оператора, проводит проверку всех устройств машины, задавая в соответствии с указаниями управляющей программы необходимую последовательность тестов.

## 6-5. РАБОТА ОПЕРАТОРА С СИСТЕМОЙ ТЕСТ-МОНИТОР

Различают два режима тестового контроля ЭВМ: *наладочный* и *профилактический*. Наладочный режим используется при запуске ЭВМ в работу после монтажа, ремонта, замены отдельных устройств или их модернизации. В наладочном режиме обычно выполняется полный комплексный контроль ЭВМ с использованием всей системы тестов. Профилактический контроль проводится периодически, при ежедневном запуске ЭВМ в работу, если она работает не круглосуточно, или в заданное время при круглосуточной работе; после замены или наладки отдельных элементов устройств; после сбоя при решении задач. При профилактическом контроле обычно используют не всю систему тестов, а лишь часть ее, задаваемую оператором. При профи-

лактическом контроле проверяемые устройства обычно переводятся в тяжелый режим работы.

*Тяжелым режимом* работы некоторого устройства называется такой режим, при котором вероятность проявления либо всех, либо некоторых физических неисправностей схемы выше, чем при других режимах. Тяжелые режимы используются для выявления элементов, параметры которых приближаются к своим граничным значениям. В нормальном режиме работы наличие таких элементов приводит к сбоям. Применение тяжелых режимов, задаваемых либо вручную оператором, например изменением питающего напряжения устройства, либо автоматически с помощью теста, использующего задание соответствующих исходных данных, ставит эти элементы в критические условия, что приводит к их отказу, который и выявляется системой контроля. Примеры создания тяжелых режимов работы отдельных устройств будут рассмотрены в разделах, посвященных контролю этих устройств.

Рассмотрим порядок работы оператора при проверке ЭВМ. Так как эта работа во многом определяется конкретным видом ЭВМ, то будем рассматривать этот порядок на примере описанной ранее системы тестов ТЕСТ-монитор.

Вначале оператор устанавливает тестовую магнитную ленту на лентопротяжный механизм и включает процессор и все периферийные устройства ЭВМ.

Далее приводятся в состояние ГОТОВО (один из режимов работы устройств) устройство управления накопителями на магнитных дисках, устройство управления накопителями на магнитных лентах, сами накопители на магнитных дисках и магнитных лентах, печатающие устройства ввода-вывода и перфокарточные устройства ввода-вывода.

После этого можно вводить в ОЗУ ЭВМ систему ТЕСТ-монитор. Ввод начинается с базового теста. Для этого на переключателях пульта управления процессора набирается адрес накопителя на магнитной ленте, где находится тестовая лента, и нажимается клавишный переключатель ЗАГРУЗКА. Если какая-либо из цепей, которые используются для команд базового теста, неисправна, то базовый тест до конца не пройдет. Возможны два варианта: останов прохождения базового теста и появление соответствующей сигнализации на регистре ошибок пульта управления и закливание базового теста. При этом сигнала на пульте управления не появляется и оператор обнаруживает факт закливания по времени прохода базового теста. Если базовый тест не прошел, то для поиска ошибки можно либо использовать сигнализацию об ошибке, появившуюся на пульте управления, либо попытаться запустить с пульта управления диагностические тесты процессора, которые хранятся в постоянной памяти ОЗУ, либо отключить базовый тест и ввести управляющую программу ТЕСТ-монитора.

Какое из указанных решений принять, зависит от характера ошибки, вызвавшей отказ базового теста.

При успешном прохождении базового теста на пульте управления загорается индикатор ОЖИДАНИЕ. Следующий этап — ввод управляющей программы. Для этого вновь нажимается клавишный переключатель ЗАГРУЗКА. После ввода управляющей программы пишущая машинка напечатает следующий текст:

ОЖД НЗП ТМЕС ВЕРСИЯ О

что означает: режим работы процессора ОЖД — ОЖИДАНИЕ; НЗП — программа начальной загрузки; ТМЕС — ТЕСТ-монитор Единой системы и его версия.

Дальнейшая работа с управляющей программой происходит путем обмена информацией между этой программой и оператором. Оператор может воздействовать на выполнение управляющей программы тремя способами:

1) через клавишный переключатель «ПРЕРЫВАНИЕ» на пульте управления;

2) через глагольный язык входных сообщений с пишущей машинки;

3) через буферный язык входных сообщений с пульта управления. Переключатель ПРЕРЫВАНИЕ используется для прекращения выполнения соответствующей тест-секции или для передачи с пульта управления указания на буферном языке входных сообщений. Буферный язык входных сообщений служит для передачи указаний управляющей программе с пульта управления. Это сообщение можно ввести в любое время, установив ЭВМ в состояние ОСТАНОВ и нажав переключатель ПРЕРЫВАНИЕ. После ввода сообщение с буферного языка будет переведено в глагольный язык входных сообщений. Этот язык определяет набор функций, выполнение которых оператор может потребовать от управляющей программы. Каждой функции соответствует глагол, описываемый буквой. Каждое глагольное сообщение состоит только из глагола или из глагола и переменного поля. В переменном поле, которое в свою очередь может состоять из подполей, записываются типы устройств, их адреса и т. д. Для разделения подполей используется точка. Можно сформировать цепочку глагольных сообщений длиной до 80 символов. Тогда каждое сообщение внутри цепочки отделяется от другого наклонной чертой. После последнего сообщения наклонная черта не ставится.

Существует несколько категорий глаголов, например глаголы, управляющие работой тест-секций:

F — запрос на прекращение выполнения;

O — запрос на вмешательство оператора;

S — запрос на включение и указание режима работы управляющей программы с данной тест-секцией;

Z — запрос на выключение;

глаголы, управляющие работой управляющей программы (УП):

V — запрос на начало или продолжение работы;

H — запрос на останов;

I — запрос на инициализацию (под инициализацией понимается настройка управляющей программы на конкретную ЭВМ)

и другие запросы.

Например, глагольное сообщение

I/L700. 7FF/075 A/B

означает:

I — инициализировать управляющую программу;

L700.7FF — загрузить тест-секции с поисковыми номерами от 700 до 7FF;

075 A — после загрузки секции 75 A остановиться для вмешательства оператора;

В — начать выполнение.

Вернемся к работе ТЕСТ-монитора после ввода управляющей программы в ОЗУ. На пишущей машинке после того, как машина печатает сообщение «ОЖД НЗП ТМЕС ВЕРСИЯ О», оператор должен напечатать следующее глагольное входное сообщение:

V22 OXFFFF/I/L001.049/S01B.00/S024.14.15/B, которое обозначает:

V22 — вид ЭВМ (здесь — ЭВМ ЕС-1022);

OXFFFF — максимальный адрес ОЗУ (для ЭВМ с емкостью ОЗУ 128 К пишется 01FFFF, 256 К — 03FFFF, 512 К — 07FFFF);

I — инициализировать управляющую программу;

L001.049 — загрузить тест-секции с поисковыми номерами от 001 до 049;

01B.00 — включить тест-секцию с поисковым номером 01B с заданным режимом работы с УП, обозначенным 00;

024.14.15 — включить тест-секцию с поисковым номером 024 с режимами работы с УП, обозначенными 14 и 15.

Режимы работы тест-секций с управляющей программой задаются специальной таблицей, входящей в состав УП.

После печати на пишущей машинке текста фразы на блоке управления пишущей машинки нужно нажать кнопку КТ (конец текста) и управляющая программа начнет поочередную загрузку тест-секций и их выполнение.

В ходе выполнения тест-секций, если нет специальных указаний оператора (а любое указание вводится, как отмечалось выше, с помощью переключателя ПЕРЕРЫВАНИЕ и буферного языка сообщений), последовательно проверяются все устройства ЭВМ и на печатающей машинке печатаются тексты вида (см. табл. П-1):

I I F1700

К

H F1710

К

H F1720

К

· · · · ·  
H F1770

К

H F1800

К

H F1810

К

H F1820

К

H F1900

К

H E2000

К

H E2010

К

· · · · ·  
H E2060

К

Здесь Н — начало просмотра соответствующего теста; К — конец просмотра.

После прогона теста стандартного набора команд, арифметики с плавающей запятой, десятичной арифметики и прерываний управляющая программа включает тест основной памяти. В ответ на пишущей машинке в случае нормального прохождения этого теста будет напечатано:

ИНИЦИАЛИЗАЦИЯ ТС ЗАКОНЧЕНА  
ПРОВЕРКА ЗАПИСИ И ЧТЕНИЯ КЗП ЗАКОНЧЕНА  
ПРОВЕРКА АДРЕСАЦИИ ЗАКОНЧЕНА  
ПРОВЕРКА СРАВНЕНИЯ КЗП ЗАКОНЧЕНА  
ПРОВЕРКА СЗП ПРИ РАБОТЕ КАНАЛОВ В-В ЗАКОНЧЕНА  
КЗП ВОССТАНОВЛЕННЫ

К

Здесь ТС — тест-секция; КЗП — ключ защиты памяти; СЗП — средства защиты памяти; В-В — ввод-вывод.

Кроме сообщений о том, какие тест-секции прошли и какие проверочные работы выполнены, управляющая программа печатает и указания оператору о его необходимых действиях при проверке отдельных устройств, например при проверке перфокарточного устройства вывода печатается следующий текст:

В

ПРИМЕР 1 — ПРОВЕРКА НЕГОТОВОГО УСТРОЙСТВА  
ПЕРЕВЕДИ В НЕ ГОТОВО

Прочитав этот текст, оператор должен подойти к перфокарточному устройству вывода и перевести его в состояние НЕ ГОТОВО. После проверки пишущая машинка отпечатает

КОНЕЦ ПРИМЕРА 1

Эти операции будут выполняться столько раз, сколько перфокарточных устройств вывода есть в составе конкретной ЭВМ.

Полная распечатка всех сообщений управляющей программы при полной комплексной проверке ЭВМ занимает около 2,5 ч.

Полностью ТЕСТ-монитор прогоняется обычно в наладочном режиме. При ежедневных профилактических проверках используются тест-секции отдельных устройств. Задание этих тест-секций или осуществляют с пульта управления с помощью буферного языка сообщений, или на пишущей машинке печатают глагольные входные сообщения с номерами тест-секций, необходимых для проверки выбранных устройств ЭВМ. Следует помнить, что независимо от того, сколько устройств ЭВМ мы хотим проверить с помощью ТЕСТ-монитора — одно или все, обязательно должна быть введена в начале проверки управляющая программа, так как без нее тест-секции не работают.

Рассмотрим, как же осуществляется проверка отдельных устройств ЭВМ и основных блоков этих устройств.

## 6-6. ТЕСТОВЫЙ КОНТРОЛЬ ПРОЦЕССОРА

Прежде чем рассматривать систему тестового контроля процессора, кратко напомним структуру процессоров современных ЭВМ и их основные функции.

Процессор представляет собой устройство для автоматического выполнения последовательности операций (арифметических, логических, обмена информацией с другими устройствами ЭВМ), предусмотренных программой решения задачи. Процессор связан с оперативным и внешними запоминающими устройствами машины и каналами устройств ввода-вывода. В современной литературе, в частности в справочно-информационной литературе об ЭВМ ЕС-1020 и ЕС-1022, в составе процессора указаны также ОЗУ, УУ и сами каналы ввода-вывода ЭВМ. В этом случае выделяют понятие *центральный процессор*, выполняющий те функции, о которых шла речь в начале раздела. В дальнейшем тексте слово «центральный» будет опущено и термин «процессор» будет применяться к устройству, не включающему ОЗУ и УУ каналами. Процессор состоит из арифметического и управляющего устройств.

Устройство управления управляет ходом вычислительного процесса в ЭВМ, т. е. выбирает машинные команды из ОЗУ, расшифровывает их, преобразуя в микрооперации (каждая микрооперация соответствует одному элементарному преобразованию в ЭВМ, например, очистке сумматора, сдвигу, изменению содержимого счетчика на единицу и т. д.). Кроме основных функций, связанных с организацией вычислительного процесса, УУ обеспечивает: пуск и останов ЭВМ; управление каналами и устройствами ввода-вывода; управление различными режимами работы машины; связь оператора (через пульт управления) с различными устройствами ЭВМ и т. д. Арифметическое устройство формирует из адресной части машинных команд (в адресной части машинной команды указаны адреса ячеек ОЗУ, где хранятся операнды, над которыми нужно выполнить указанную в операционной части команды операцию): исполнительные адреса и, получив из ОЗУ операнды, выполняет микропрограмму (набор микроопераций, соответствующих заданной машинной команде).

Обобщенная структурная схема процессора приведена на рис. 6-3.

Рассмотрим очень кратко цикл работы процессора при выполнении очередной команды:

- 1) счетчик команд передает в ОЗУ адрес очередной команды.
- 2) адресная часть команды из ОЗУ направляется в регистр адреса.
- 3) содержимое адресной части команды из регистра адреса передается в АУ.

- 4) АУ формирует исполнительный адрес.
- 5) из АУ исполнительный адрес передается в ОЗУ.
- 6) операнды исполняемой команды из ОЗУ передаются в информационный регистр.
- 7) из информационного регистра операнды следуют в АУ.

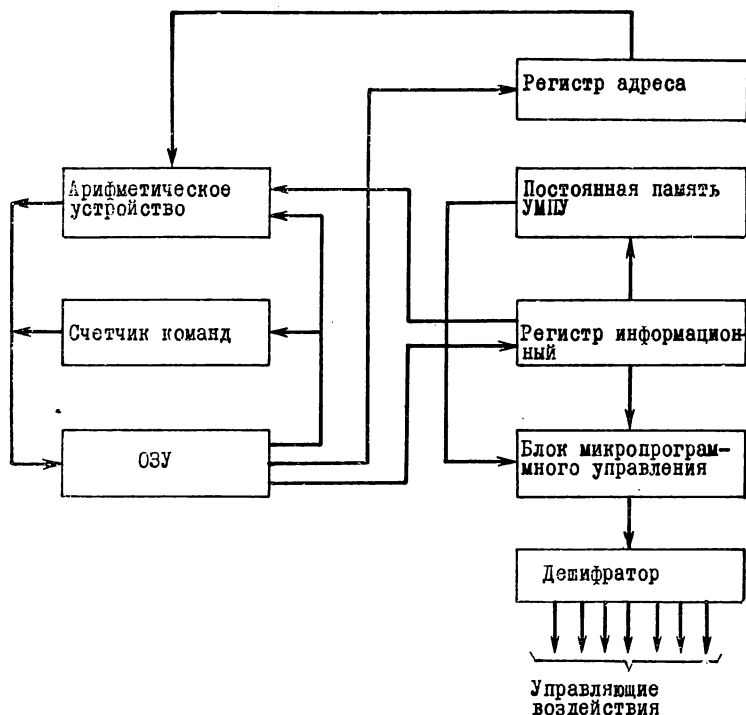


Рис. 6-3. Структурная схема процессора ЭВМ.

8) АУ расшифровывает код операции, а блок микропрограммного управления выдает управляющие сигналы путем последовательной выборки микроопераций, соответствующих расшифрованной машинной команде, из постоянной памяти устройства микропрограммного управления (УМПУ).

9) полученный результат записывается в информационный регистр;

10) результат вычислений из информационного регистра записывается в соответствующую ячейку ОЗУ.



Заметим, что в постоянной памяти УМПУ хранятся все микрокоманды, используемые процессором данной ЭВМ, а блок микропрограммного управления предназначен для управления арифметическим и другими функциональными устройствами в процессе выполнения микропрограмм всех машинных команд ЭВМ.

Из описания работы процессора ясно, что для создания эффективной системы контроля необходимо организовать контроль работы отдельных устройств и их взаимодействия и контроль выполнения всех операций, выполняемых процессором.

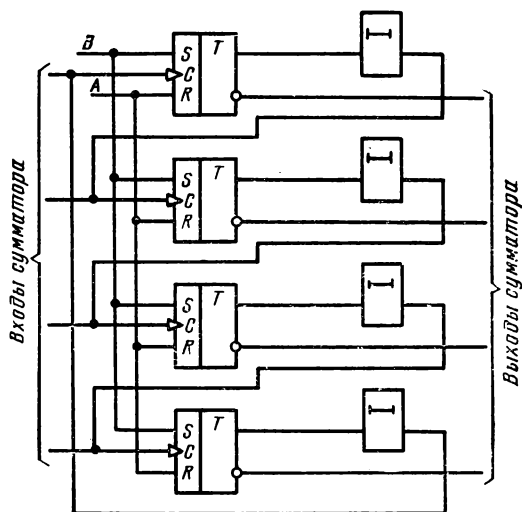


Рис. 6-4. Функциональная схема последовательного накапливающего сумматора.

Выше говорилось о том, что при профилактическом контроле обычно создается тяжелый режим работы отдельных устройств и их элементов. Рассмотрим примеры организации тяжелых режимов работы.

Основным элементарным блоком арифметического устройства ЭВМ является *сумматор*, выполняющий операцию арифметического суммирования двух чисел. По способу обработки чисел сумматоры можно разделить на последовательные, параллельные и последовательно-параллельные.

В последовательном сумматоре производится поразрядное сложение двух чисел, при этом сложение разрядов идет последовательно от младших к старшим (рис. 6-4).

В параллельном сумматоре сложение всех разрядов двух чисел производится одновременно (рис. 6-5).

Последовательно-параллельные сумматоры применяются обычно тогда, когда число разрядов в сумматоре меньше числа разрядов в складываемых числах. В этом случае складываемые числа делятся на группы разрядов (число разрядов в группе равно количеству разрядов сумматора) и на сумматор последовательно во времени подаются группы разрядов, которые складываются на нем параллельно.

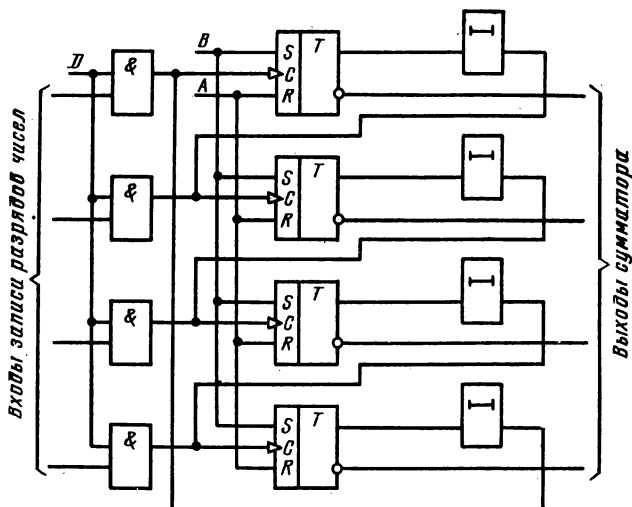


Рис. 6-5. Функциональная схема параллельного накапливающего сумматора.

Рассмотрим тяжелые режимы работы для сумматора параллельного типа и регистра сдвига информации.

**Пример 6-3.** Рассмотрим режим работы параллельного накапливающего сумматора, состоящего из четырех потенциальных триггеров, линий задержек и схем совпадения И (см. рис. 6-5).

Пусть в сумматор записано число 0010 и к нему требуется прибавить число 1111. Для этого все разряды числа 1111 одновременно подаются на входы схем совпадений И. При подаче синхронизирующего импульса на второй вход схемы совпадений (по шине D) коды разрядов второго слагаемого поступают на счетные входы триггеров сумматора. При перебросе триггера из единичного в нулевое состояние импульс переносимого разряда через схему линии задержки поступает на счетный вход триггера старшего разряда. Задержанный сигнал должен поступить позже, чем произойдет переключение триггера под действием сигнала, поступившего при подаче на входы триггеров второго слагаемого. Таким образом, сложение двух чисел, описан-

ных в этом примере, приводит к тяжелому режиму работы сумматора и позволяет выявить элементы с изменившимися параметрами в одной из цепочек: триггер — линия задержки — схема совпадений. Меняя код первого слагаемого, т. е. смещая единицу влево, проверяют все цепочки сумматора. Проверить работу четных триггеров с их линиями

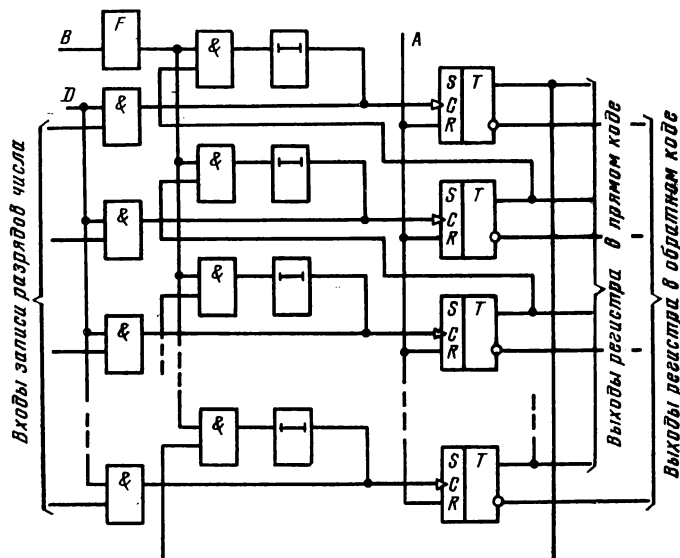


Рис. 6-6. Функциональная схема регистра сдвига, осуществляющего правый сдвиг.

задержки и схемами совпадений для  $m$ -разрядных сумматоров можно с помощью набора кодов

$$0 \dots 0101 \text{ и } 1 \dots 1111,$$

а нечетных триггеров с их линиями задержки и схемами совпадений

$$1 \dots 1010 \text{ и } 1 \dots 1111.$$

**Пример 6-4.** Рассмотрим режим работы регистра сдвига информации. Регистр сдвига информации представляет собой устройство для приема, промежуточного хранения и сдвига последовательно на один разряд всех разрядов записанного в нем числа либо от старших к младшим (правый сдвиг), либо от младших к старшим (левый сдвиг). В примере рассмотрим один из режимов работы регистра, осуществляющего правый сдвиг (рис. 6-6). Регистры сдвига позволяют получить информацию о записанных в них числах в прямом и обратном кодах.

Различают регистры с предварительной установкой всех триггеров в состояние 0 или без установки. Если в регистре с помощью логических схем имеется возможность записывать не только прямые, но и инвертированные значения каждого разряда числа, то установка в 0 не требуется. Если же такой возможности нет, то перед записью числа в регистр на входы триггеров всех разрядов подается сигнал

установки их в состояние 0 и лишь после этого записывается число и выполняется его сдвиг соответственно вправо или влево. Импульс сдвига формируется формирователем  $F$ .

Пусть во всех триггерах регистра записаны единицы, тогда нагрузка на формирователь будет максимальной и за счет падения напряжения на нем при изменении параметров какой-либо из схем совпадения в триггер может записаться сигнал помехи в интервал времени после прихода импульса записи числа в регистр и до прихода импульса сдвига. Таким образом, код числа 1 ... 1111 создает тяжелый режим работы регистра сдвига и позволяет проверить надежность работы регистра в целом. Шина  $A$  служит для подачи сигнала установки всех триггеров в состояние 0; шина  $B$  — для подачи импульса сдвига; шина  $D$  — для подачи тактового импульса записи числа через схемы совпадения.

В приведенных примерах показано, какие коды могут быть использованы в тестах для профилактического контроля отдельных блоков процессора, т. е. для создания в этих узлах процессора таких режимов, которые позволяют выявить элементы с параметрами, приближающимися к граничным.

Для проверки устройств процессора применяют тесты, использующие *постоянные и переменные коды*. Каждый тест в том и другом случае представляет собой набор примеров. В тестах на *постоянных кодах* решение всех примеров получено заранее. Проверка устройства или группы блоков устройства состоит в том, что на этом устройстве решаются последовательно (в автоматическом режиме, задаваемом тестом) один пример за другим и полученные результаты сравниваются с эталонными. Если хотя бы один из результатов не совпадает, то на пульт управления ЭВМ подается сигнал об ошибке. Обычно система тестов устроена таким образом, что в сообщении об ошибке, которое выводится на печатающее устройство ЭВМ, включается информация о блоке, на котором эта ошибка возникла. Для обеспечения более надежной проверки обычно предусматривается ручная или автоматическая возможность закливания теста, что дает возможность выявить не только постоянные, но и случайные ошибки.

Проверка на *переменных кодах* также использует идею набора примеров. Однако при такой проверке в тесты не заложены эталонные ответы решения примеров. Тесты на переменных кодах в зависимости от типа проверяемого устройства используют проверку полученного результата решения либо за счет восстановления первоначального набора исходных чисел, либо путем выполнения обратного действия.

В качестве примеров при проверке на переменных кодах очень часто используют проверку тождеств, например

$$\left| \frac{ab}{b} - a \right| \approx | \sqrt{aa} - |a| |.$$

Знак  $\approx$  показывает, что результаты вычисления левой и правой части тождества могут абсолютно точно не совпасть из-за погрешности округления. Поэтому после выполнения необходимых для решения примера операций умножения, деления и извлечения корня выполняется операция вычитания модулей двух найденных чисел  $|C|$  и  $|D|$ , где

$$|C| = \left| \frac{ab}{b} - a \right|;$$

$$|D| = | \sqrt{aa} - |a| |$$

и результат вычитания

$$E = |C| - |D|$$

сравнивается с заданной точностью вычислений  $\epsilon$ . Если окажется, что  $E \leq \epsilon$ , то программа переходит к следующему примеру, если же неравенство не выполняется, то печатается сообщение об ошибке.

При составлении примеров для программы на переменных кодах важную роль играет задание исходных чисел (в рассмотренном примере чисел  $a$  и  $b$ ). Коды этих чисел могут быть заданы заранее в виде таблиц либо вычисляться заново при каждом просмотре примера (при проверке решение каждого примера обычно выполняется многократно). В первом случае при очередном просмотре примера программа обращается к новой строке таблицы. Во втором случае при каждом новом обращении к примеру производится предварительное преобразование кодов исходных чисел, например, путем их сложения, или умножения, или сдвига и т. д. Таким образом, описанный пример проверки на переменных кодах выполняется путем многократного решения одного и того же примера, но с разными исходными данными.

Преимуществом тестов на постоянных кодах является большая степень локализации неисправности, но они занимают много места в памяти ЭВМ. Тесты на переменных кодах занимают гораздо меньший объем памяти ЭВМ, но не во всех устройствах эти тесты позволяют достаточно

точно локализовать неисправность. Поэтому в системе тестов обычно используют оба варианта тестов.

Тесты проверки блоков УУ обычно строятся таким образом, чтобы в соответствии с принципом работы процессора проверить в отдельности устройства, участвующие в расшифровке адресной и операционной частей команды. Проверка устройств, участвующих в обработке операционной части команды, проводится тестами, содержащими примеры с использованием всех машинных кодов данной ЭВМ и их различных комбинаций для создания тяжелых режимов работы проверяемых устройств.

Проверка блоков, участвующих в расшифровке адресной части команды, может быть проведена так: в ячейки ОЗУ записывают некоторую эталонную информацию, а затем с помощью теста в адресные регистры считывают адреса ячеек ОЗУ, где эта эталонная информация записана, и производят сравнение.

Тесты проверки АУ процессора строятся так, чтобы проконтролировать его работу с охватом максимально возможного числа цепей прохождения информации и при необходимости создать в проверяемых блоках тяжелые режимы. При проверке АУ основной задачей является проверка правильности выполнения всех арифметических и логических операций, осуществляемых данной машиной. Так же как и при проверке УУ, тест проверки блоков (а точнее, цепей выполнения различных операций) АУ состоит из набора примеров. Для простой проверки (первичной) правильности выполнения той или иной операции очень часто используют контроль по четности (хотя эта форма контроля наибольшее свое применение нашла в аппаратных методах контроля). Контроль по четности исключительно широко используется в ЭВМ на различных этапах обработки информации, начиная с поступления ее с устройств ввода.

Контроль по четности можно использовать, например, в качестве первого примера при проверке операций вычитания, сдвига и т. д.

Операция вычитания в ЭВМ обычно реализуется как выполнение операции сложения одного числа в прямом коде, а второго в обратном или дополнительном. Если, например, нужно от числа  $A$  отнять число  $B$ , то число  $A$  представляется в прямом коде, а число  $B$  в обратном или дополнительном коде, а затем эти два числа складываются. Поэтому для того, чтобы провести операцию  $A - B$ , число  $B$  необходимо предварительно перевести в обратный или

дополнительный код (напомним, что обратный код числа  $V$  получается путем инвертирования каждого разряда этого числа, а дополнительный код — добавлением к обратному единицы в младшем разряде). Контроль по четности при переводе числа в обратный код выполняется по следующему правилу: если количество разрядов в исходном числе четное, то и четность числа в обратном коде та же, что и у исходного числа; если же количество разрядов нечетное, то четность числа в обратном коде противоположна четности исходного числа. При задании контрольного примера четность числа, которое необходимо вычесть, вычисляется заранее и после перевода числа в обратный (или дополнительный) код выполняется сравнение по четности. Однако в подавляющем большинстве примеров при выполнении контрольного примера используют для сравнения эталонный результат решения (при работе с тестами на постоянных кодах). Контрольные примеры проверки выполнения арифметических и логических операций в тестах ЕС ЭВМ будут приведены в следующем параграфе. Здесь же остановимся несколько подробнее на контроле двух операций, одна из которых используется практически во всех примерах. Это операция условного перехода по признаку  $\omega$ . Вторая операция также встречается достаточно часто — это признак переполнения или аварийный останов по сигналу  $\phi$ . В тестовых примерах для системы ТЕСТ-монитор при проверке арифметических операций эти две операции используются совместно. Операция условного перехода, имеющая в мнемоническом коде на языке АССЕМБЛЕР наименование ВС, при выполнении арифметических операций осуществляет передачу управления в зависимости от выполнения следующих условий: код условия равен 0 — результат равен нулю; 1 — результат отрицательный; 2 — результат положительный; 3 — арифметическое переполнение.

Для проверки цепей АУ, работающих при выработке сигнала  $\phi$ , примеры теста подбираются таким образом, чтобы этот признак выполнялся. Например, в ТЕСТ-мониторе для операции арифметического сложения чисел с фиксированной запятой выбирают два числа, сумма которых дает переполнение разрядной сетки. Команду условного перехода при этом строят так, чтобы она выходила на останов при отсутствии признака  $\phi$ .

Проверку правильного выполнения команд условного перехода обычно проводят до проверки выполнения ариф-

метических и логических операций, поскольку, как это уже отмечалось, в контрольные примеры проверки выполнения операций в подавляющем большинстве случаев входит команда условного перехода. Так, например, в ТЕСТ-

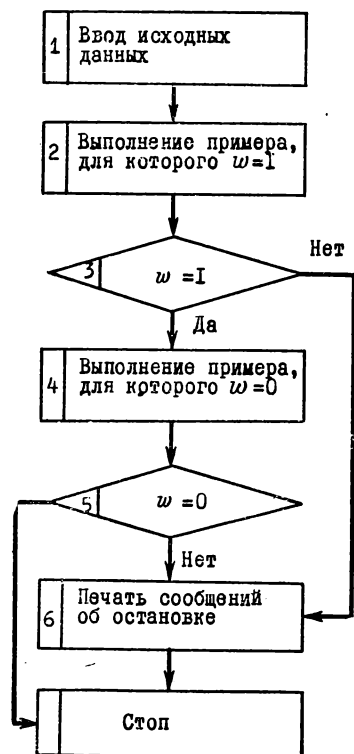


Рис. 6-7. Блок-схема программы проверки операции условного перехода.

мониторе проверка процессора начинается с теста проверки стандартного набора команд, а в этом тесте в первых примерах проверяются команды ВС — условного перехода и С — сравнения. Проверку правильного выполнения команды условного перехода обычно организуют так: после команды, вырабатывающей признак  $\omega$ , идут команды условного перехода по этому признаку и команды сравнения содержимого ячейки, в которую передано управление, с заданным значением. Если значения не совпадают или если управление передается не по адресу, то выполняется останов прохождения тестов и печатается информация об ошибке. Условно работу программы по проверке операции условного перехода, имеющей признаки  $\omega = 0$  и  $\omega = 1$ , можно представить с помощью блок-схемы рис. 6-7. Программа работает так. После ввода исходных данных выполняется пример, по резуль-

татам которого должен быть выработан признак  $\omega = 1$ ; если он вырабатывается, то программа переходит к выполнению второго примера, если же нет, то печатается сообщение о причине останова (блок 6) и производится останов. В результате решения второго примера должен быть выработан признак  $\omega = 0$ , если он получен, то программа заканчивает свою работу (выходит на СТОП), если же нет, то печатается сообщение о причине останова и выполняется останов.



## 6-7. ТЕСТЫ В СИСТЕМЕ ТЕСТ-МОНИТОР ДЛЯ ПРОВЕРКИ ПРОЦЕССОРА

Проверка различных блоков и устройств процессоров ЕС ЭВМ выполняется средствами аппаратурного и программного контроля. Тесты программного контроля процессора служат для проверки правильности выполнения всех арифметических и логических операций для данных, записанных в форме с фиксированной запятой, с плавающей запятой и для операций десятичной арифметики.

Для проверки выполнения команд используются следующие тесты:

- тест стандартного набора команд;
- тест десятичной арифметики;
- тест арифметики с плавающей запятой.

В свою очередь каждый из этих тестов состоит из набора тест-секций, а каждая тест-секция предназначена для проверки одной или группы команд ЭВМ.

Каждая тест-секция состоит из ряда примеров и подпрограмм. Примеры строятся по принципу сравнения выполнения проверяемой команды или группы команд с известным (эталонным) результатом. Если результаты не совпадают; то печатается стандартное сообщение об ошибке. Это сообщение имеет следующий вид:

```
AAA ... A
BBB ... B
XX YYYYY ZZZZ .
```

где AAA ... A — ожидаемый результат; BBB ... B — эталонный результат; XX — машинный код проверяемой команды; YYYYY — мнемонический код проверяемой команды на АССЕМБЛЕРЕ; ZZZZ — абсолютный адрес переключателя цикла.

Каждый тест предусматривает возможность закливания как отдельного примера, так и всей тест-секции.

Рассмотрим ряд примеров проверки отдельных команд и групп команд теста стандартного набора команд. Тест стандартного набора команд предназначен для проверки правильности выполнения команд с фиксированной запятой, логических команд, команд переключения состояния и команд переходов.

Так как все тесты в системе ТЕСТ-монитор написаны на языке АССЕМБЛЕР, то и в рассматриваемых ниже примерах программы они записаны на этом же языке. Для справки

в табл. П-2 приведен список команд на языке АССЕМБЛЕР, использованных в примерах, и перечень выполняемых ими операций.

**Пример 6-6.** Проверка арифметической операции сложения, имеющей мнемонический код на языке АССЕМБЛЕР AR. По этой операции число с фиксированной запятой из регистра первого операнда складывается с числом с фиксированной запятой, находящимся в регистре второго операнда. Результат записывается в регистр первого операнда.

Дадим описание алгоритма проверки выполнения операции арифметического сложения:

1) загружаются в регистры R0 и R1 исходные числа, записанные в шестнадцатиричной форме и имеющие числовые значения 00000000 и 00000001;

2) выполняется с помощью операции AR их сложение;

3) выполняется переход по команде условного перехода

BC 2, AR 01

По коду условия происходит переход к команде с меткой AR 01 (конец примера), если результат выполнения команды AR положительный. В любом другом случае печатается сообщение об ошибке.

Отметим, что перед командой

BC 2, AR 01

в программе записана команда

BC 0, L 00 P 12

В этой команде в качестве первого операнда записан 0, поэтому условного перехода не происходит, команда пропускается и в программе выполняется следующая команда, т. е.

BC 2, AR 01

Такие «пустые» команды условного перехода, т. е. команды, не вызывающие никаких действий, предусмотрены в программе для внесения нужных оператору ЭВМ изменений в тест с пульта машины. Для этого оператор вместо нуля записывает в операцию BC нужные ему для дополнительных проверок условия перехода.

Программа примера на языке АССЕМБЛЕР приведена на рис. 6-8. Мнемонические коды и расшифровка выполняемых ими операций приведены в табл. П-2. Необходимые пояснения приведены непосредственно в тексте программы.

Пример 2 построен по такому же алгоритму, что и первый. От первого он отличается исходными числами: 00000001 и FFFFFFFF (второе число представляет собой число минус единица в дополнительном коде) и условием перехода в операторе условного перехода. Переход в примере 2 осуществляется по признаку 0 (т. е. если результат операции сложения равен нулю).

В примере 3 исходными данными являются числа 00000000 и FFFFFFFF. После выполнения их сложения по команде AR проверяется условие отрицательности полученного результата.

В примере 4 складываются следующие два числа: 70000000 и 60000000 и после сложения их по операции AR проверяется, будет ли результат давать переполнение разрядной сетки. Если да, то про-

верка операции сложения с кодом AR закончена и можно переходить к проверке следующей операции, если же нет, то печатается сообщение об ошибке.

Проверка одной арифметической операции сложения с фиксированной запятой на четырех примерах позволяет выявить все возможные результаты выполнения этой операции (т. е. меньше 0, равно 0, больше 0, переполнение).

RTN12	DC	X'12'	НОМЕР ПРИМЕРА
	DC	X'00'	ПЕРЕКЛЮЧАТЕЛЬ ПРИМЕРА
	DC	AL2 (RTN13—PSN)	АДРЕС СЛЕДУЮЩЕГО ПРИМЕРА
	L	1, AR03	ЗАГРУЗИТЬ 00000001 В P1
LOOP12	L	0, AR02	ЗАГРУЗИТЬ 00000000 В P0
	AR	0, 1	ВЫПОЛНИТЬ СЛОЖЕНИЕ
NOP12	BC	0, LOOP12	ПЕРЕКЛЮЧАТЕЛЬ ЦИКЛА
	BC	2, AR01	ВЕТВИТЬСЯ К ВЫХОДУ ИЗ ПРИМЕРА, ЕСЛИ КУ РАВЕН 2
	SVC	X'D0'	ПЕЧАТЬ СООБЩЕНИЯ ОБ ОШИБКЕ, ЕСЛИ КУ НЕ РАВЕН 2
	DC	X'40'	
	DC	AL1 (CCERR3 — CCERR2)	
	DC	S (CCERR2)	
	MVC	PRINT (8), OP19	ПЕРЕСЛАТЬ КОД ОПЕРАЦИИ В ОБЛАСТЬ СООБЩЕНИЯ
*	LA	0, NOP12	ЗАГРУЗИТЬ АБСОЛЮТНЫЙ НОЛЬ
	BAL	10, SUBRT	ПЕРЕЙТИ К ПОДПРОГРАММЕ ПРЕОБРАЗОВАНИЯ
*	SVC	X'06'	ОКОНЧИТЬ ПРИМЕР
AR01	SNOP	0, 4	
AR02	DS	XL4'00000000'	
AR03	DS	XL4'00000001'	

Рис. 6-8. Программа примера для проверки выполнения операции AR.

**Пример 6-7.** Рассмотрим набор тестов для проверки операции умножения, имеющей на языке АССЕМБЛЕР мнемонический код MR. При этой операции множимое задается по первому адресу, а множитель — по второму. В результате умножения получается операнд двойной длины и он записывается в регистр, заданный первым адресом.

Рассмотрим описание алгоритма проверки выполнения команды MR:

- 1) загрузить исходные шестнадцатиричные числа 00000003 и 00000003 в регистры P4 и P5;
- 2) выполнить операцию умножения по команде MR4,4.

При выполнении этой команды число, содержащееся в регистре с нечетным номером, из пары общих регистров P4 и P5 умножается на число, содержащееся в регистре второго операнда, т. е. в P4. Результат записывается в оба регистра, при этом младшие разряды произведения занимают младшие позиции нечетного общего регистра.

На рис. 6-9 представлено содержимое регистров *P4* и *P5*. В верхних строках каждой таблицы записано двоичное представление чисел, а в нижних — шестнадцатиричное.

3) загрузить содержимое регистра *P5* (т. е. число 00000009 после выполнения операции умножения в регистр *P2*).

4) загрузить число 00000009 в регистр *P3*,

5) выполнить с помощью операции

CR 2,3

сравнение содержимого регистров *P2* и *P3*.

*P4*

0000	0000	0000	0000	0000	0000	0000	0011
0	0	0	0	0	0	0	3

а)

*P4*

0000	0000	0000	0000	0000	0000	0000	0000
0	0	0	0	0	0	0	0

б)

Рис. 6-9. Содержимое регистров *P4* и *P5* до (а)

6) осуществить условный переход по команде

BC 7,MR01

Если операнды в регистрах *P2* и *P3* равны, то перейти к следующей команде, если нет — печатать сообщение об ошибке по команде с меткой *MR01*.

7) загрузить в регистры *P2* и *P3* соответственно содержимое регистра *P4* (00000000 — после выполнения операции умножения) и число 00000000.

8) выполнить сравнение по операции CR и условный переход по операции BC

BC 8,MR02

В этом случае, если операнды в регистрах *P2* и *P3* равны, происходит переход к команде с меткой *MR02* — конец примера, если же нет, то печатается сообщение об ошибке.

На рис. 6-10 приведен текст этой программы.

Алгоритм следующих двух примеров проверки операции MR такой же, как и описанный выше, только в качестве исходных чисел в примере 2 используются — множимое отрицательное FFFFFFFF, а множитель положительный (00000003); в примере 3 — множитель и множимое отрицательные и равные FFFFFFFF.

Таким образом, три приведенных теста дают возможность проверить правильность выполнения операции умножения MR для всех видов чисел с фиксированной запятой.

**Пример 6-8.** Проверка операции деления. Рассмотрим тесты для проверки операции деления с мнемоническим кодом DR. По этой операции делимое (операнд двойной длины), заданное первым адресом, делится на делитель (операнд нормальной длины), заданный вторым адресом. В результате операции получается частное от деления и остаток. Частное и остаток помещаются в регистрах, где находилось делимое. При этом частное от деления находится в регистре с нечетным номером, а остаток — в регистре с четным номером.

Рассмотрим описание алгоритма выполнения операции DR для случая, когда делимое отрицательно и равно FFFFFFFF FFFFFFFF, а делитель положителен и равен 00000002

P5

0000	0000	0000	0000	0000	0000	0000	0011
0	0	0	0	0	0	0	3

P5

0000	0000	0000	0000	0000	0000	0000	1001
0	0	0	0	0	0	0	9

п после (б) выполнения операции умножения.

- 1) загрузить делитель в регистр P7 и делимое в регистры P5 и P4;
- 2) выполнить операцию деления по команде

DR 4,7

3) загрузить частное, полученное в регистре P5, в регистр P2, а ожидаемое частное, равное 00000000, в регистр P3;

4) по операции CR выполнить сравнение содержимого регистров P2 и P3 и по операции условного перехода

BC 7,DR01

выполнить переход к следующей команде, если операнды равны, и к команде с меткой DR01, печатающей сообщение об ошибке, если не равны;

5) загрузить полученный остаток в регистр P2, а ожидаемый, равный FFFFFFFF, в регистр P3;

6) по операции CR провести сравнение операндов в регистрах P2 и P3 и по операции условного перехода BC перейти к концу примера, если операнды равны, или напечатать сообщение об ошибке, если нет.

Программа примера на языке АССЕМБЛЕР приведена на рис. 6-11.

В примере 2 делимое и делитель отрицательны. При этом делитель равен FFFFFFFF, а делимое — FFFFFFFF 80000000. Алгоритм проверки тот же, что и в предыдущем примере.

RTN31	DC	X'31'	НОМЕР ПРИМЕРА
	DC	X'00'	ПЕРЕКЛЮЧАТЕЛЬ
	DC	AL2 (RTN32—PSN)	ПРИМЕРА
			АДРЕС СЛЕДУЮЩЕГО
LOOP31	L	4,MR03	ПРИМЕРА
	L	5,MR03	ЗАГРУЗИТЬ 00000003 В P4
	MR	4,4	ЗАГРУЗИТЬ 00000003 В P5
NOP31	BC	0,LOOP31	УМНОЖИТЬ
	LR	2,5	ПЕРЕКЛЮЧАТЕЛЬ ЦИКЛА
			ЗАГРУЗИТЬ СОДЕРЖИМОЕ
	L	3,MR04	P5 В P2
	CR	2,3	ЗАГРУЗИТЬ 00000009 В P3
			Q — РЕЗУЛЬТАТЫ
	BC	7,MR01	СОВПАДАЮТ
			НЕТ — ВЕТВИТЬСЯ НА
	LR	2,4	ПЕЧАТЬ СООБЩЕНИЯ
			ЗАГРУЗИТЬ СОДЕРЖИМОЕ
	L	3,MR05	P4 В P2
	CR	2,3	ЗАГРУЗИТЬ 00000000 В P3
			Q — РЕЗУЛЬТАТЫ
	BC	8,MR02	СОВПАДАЮТ
			ДА — ВЕТВИТЬСЯ К
MR01	SVC	X'D1'	ВЫХОДУ ИЗ ПРИМЕРА
	MVC	PRINT (8),OP63	ПЕЧАТАТЬ СООБЩЕНИЕ
			ОБ ОШИБКЕ
			ПЕРЕСЛАТЬ КОД
			ОПЕРАЦИИ В ОБЛАСТЬ
*	LA	9,NOP31	СООБЩЕНИЙ
	BAL	10,SUBRT	ЗАГРУЗИТЬ АБСОЛЮТНЫЙ
			АДРЕС ПЦП
			ПЕРЕЙТИ К
			ПОДПРОГРАММЕ
			ПРЕОБРАЗОВАНИЯ
MR02	SVC	X'D6'	ЗАКОНЧИТЬ ПРИМЕР
	SNOP	0,4	
MR03	DC	XL4'00000003'	
MR04	DC	XL4'00000009'	
MR05	DC	XL4'00000000'	
	CNOP	0,4	

Рис. 6-10. Программа примера для проверки выполнения операции MR.

Проверку выполнения *логических* операций рассмотрим на примере операции логического умножения NR. После выполнения этой операции результат помещается в регистр первого операнда.

Описание алгоритма.

1) загрузить в регистр P3 ожидаемый результат, равный 800F0107, а в регистры P0 и P2 исходные числа, соответственно равные

8F5F017F и 80AF0F87

2) выполнить операцию логического умножения чисел в регистрах P0 и P2

NR 2,0

3) по команде условного перехода

BC 4,NR03

RTN36	DC	X'36'	НОМЕР ПРИМЕРА
	DC	X'00'	ПЕРЕКЛЮЧАТЕЛЬ ПРИМЕРА
	DC	AL2 (RTN37—PSN)	АДРЕС СЛЕДУЮЩЕГО ПРИМЕРА
	L	7,DR03	ЗАГРУЗИТЬ ДЕЛИТЕЛЬ 00000002 В Р7
LOOP36	L	5,DR04	ЗАГРУЗИТЬ FFFFFFFF В Р5
	L	4,DR04	ЗАГРУЗИТЬ FFFFFFFF В Р4
	DR	4,7	ДЕЛИТЕЛЬ
NOP36	BC	0,LOOP36	ПЕРЕКЛЮЧАТЕЛЬ ЦИКЛА
	LR	2,5	ЗАГРУЗИТЬ ПОЛУЧЕННОЕ ЧАСТНОЕ В Р2
	L	3,DR05	ЗАГРУЗИТЬ ОЖИДАЕМОЕ ЧАСТНОЕ В Р3
	CR	2,3	Q — ПОЛУЧЕННОЕ И ОЖИДАЕМОЕ ЧАСТНЫЕ РАВНЫ
*	BC	7,DR01	НЕТ — ВЕТВИТЬСЯ НА ПЕЧАТЬ СООБЩЕНИЯ
	LR	2,4	ЗАГРУЗИТЬ ПОЛУЧЕННЫЙ ОСТАТОК В Р2
	L	3,DR04	ЗАГРУЗИТЬ ОЖИДАЕМЫЙ ОСТАТОК В Р3
	CR	2,3	Q — ПОЛУЧЕННЫЙ И ОЖИДАЕМЫЙ ОСТАТКИ РАВНЫ
*	BC	8,DR02	ДА — ВЕТВИТЬСЯ К ВЫХОДУ ИЗ ПРИМЕРА
DR01	SVC	XD1'	ПЕЧАТЬ СООБЩЕНИЯ ОБ ОШИБКЕ
	MVC	PRINT (8), OP71	ПЕРЕСЛАТЬ КОД ОПЕРАЦИИ
*	LA	9,NOP36	В ОБЛАСТЬ СООБЩЕНИЙ ЗАГРУЗИТЬ АБСОЛЮТНЫЙ АДРЕС ПЦП
	BAL	10, SUBRT	ПЕРЕЙТИ К ПОДПРОГРАММЕ ПРЕОБРАЗОВАНИЯ ЗАКОНЧИТЬ ПРИМЕР
DR02	SVC	X'D6'	
	SNOP	0,4	
DR03	DC	XL4'00000002'	ДЕЛИТЕЛЬ
DR04	DC	XL4'FFFFFFFF'	ОЖИДАЕМЫЙ ОСТАТОК
DR05	DC	XL4'00000000'	ОЖИДАЕМОЕ ЧАСТНОЕ
	CNOP	0, 4	

Рис. 6-11. Программа примера для проверки операции DR.

выполнить переход к команде с меткой NR03, если хотя бы один из битов операнда результата выполнения операции логического умножения NR не равен нулю. Если же все биты операнда результата равны нулю, то печатается сообщение об ошибке;

4) по команде

CR 2,3

выполняется сравнение содержимого в регистрах P2 и P3, т. е. полученного и ожидаемого результатов, а затем по операции условного перехода

BC 8,NR02

осуществляется переход к команде с меткой NR02 (конец программы), если числа в регистрах P2 и P3 равны, и печатается сообщение об ошибке в противном случае.

Текст программы приведен на рис. 6-12.

RTN17	DC	X'17'	НОМЕР ПРИМЕРА
	DC	X'00'	ПЕРЕКЛЮЧАТЕЛЬ
			ПРИМЕРА
	DC	AL2 (RTN18—	АДРЕС СЛЕДУЮЩЕГО
		PSN)	ПРИМЕРА
	L	3,NR06	ЗАГРУЗИТЬ В P3
			ОЖИДАЕМЫЙ РЕЗУЛЬТАТ
LOOP17	L	0,NR04	ЗАГРУЗИТЬ В P0 8F5F017F
	L	2,NR05	ЗАГРУЗИТЬ В P2 80AF0F87
	NR	2,0	ПРОВЕРИТЬ КОМАНДУ
			KУ РАВЕН 1
LOP17	BC	0,LOOP17	ПЕРЕКЛЮЧАТЕЛЬ ЦИКЛА
	BC	4,NR03	ДА — ВЕТВИТЬСЯ К
			ПРОВЕРКЕ ДАННЫХ
	SVC	X'D0'	ПЕЧАТЬ СООБЩЕНИЯ
			ОБ ОШИБКЕ
	DC	X'4015'	
	DC	S(CCERR1)	
	MVC	PRINT (8), OP11	ПЕРЕСЛАТЬ КОД
			ОПЕРАЦИИ В ОБЛАСТЬ
			СООБЩЕНИЯ
*	LA	9, NOP17	ЗАГРУЗИТЬ АБСОЛЮТНЫЙ
			АДРЕС ПЦП
	BAL	10, SUBRT	ПЕРЕЙТИ К
			ПОДПРОГРАММЕ
			ПРЕОБРАЗОВАНИЯ И
			ПЕЧАТИ
NR03	CR	2,3	Q — РЕЗУЛЬТАТЫ РАВНЫ
	BC	8,NR02	ДА — ВЕТВИТЬСЯ
			К ВЫХОДУ ИЗ ПРИМЕРА
	SVC	X'D1	ПЕЧАТЬ СООБЩЕНИЯ
			ОБ ОШИБКЕ
	MVC	PRINT (8), OP11	ПЕРЕСЛАТЬ КОД
			В ОБЛАСТЬ СООБЩЕНИЙ
	LA	9,NOP17	ЗАГРУЗИТЬ АБСОЛЮТНЫЙ
			АДРЕС ПЦП
	BAL	10,SUBRT	ПЕРЕЙТИ К
			ПОДПРОГРАММЕ
			ПРЕОБРАЗОВАНИЯ И
			ПЕЧАТИ
*	SVC	X'D6'	ЗАКОНЧИТЬ ПРИМЕР
NR02	CNOP	0,4	
NR04	DC	XL4'8F5F017F'	ДАнные
NR05	DC	XL4'80AF0F87'	ДАнные
NR06	DC	XL4'800F0107'	ОЖИДАЕМЫЙ РЕЗУЛЬТАТ

Рис. 6-12. Программа примера для проверки операции NR.



Таким образом, в данном примере организованы две проверки выполнения операции NR. При первой проверяется, работает ли операция в принципе, т. е. не обнуляются ли все биты в регистре после ее выполнения, а во второй проверяется точность логического умножения заданных чисел.

## 6-8. ПРОВЕРКА ОЗУ

Оперативное запоминающее устройство ЭВМ предназначено для приема и хранения информации, поступающей из других устройств ЭВМ или извне через устройства ввода, и выдачи этой информации по соответствующим приказам в другие устройства ЭВМ.

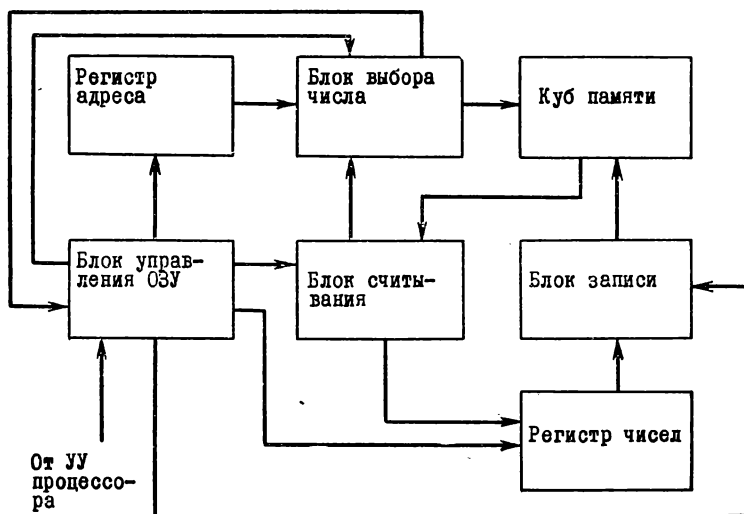


Рис. 6-13. Структурная схема ОЗУ.

Структурная схема ОЗУ представлена на рис. 6-13. Эта схема является достаточно общей и отражает принцип работы ОЗУ многих типов ЭВМ. Работает ОЗУ в соответствии со схемой рис. 6-13 следующим образом:

по сигналу, поступающему от устройства управления процессором в блок управления ОЗУ, вырабатывается сигнал, определяющий режим работы (т. е. считывание, запись, считывание-запись и т. д.), и в регистр адреса записывается код адреса числа, которое нужно считать (режим считывания), или код адреса числа и само число, записываемое в регистр чисел (режим записи);

блок управления ОЗУ вырабатывает управляющие сигналы для выполнения заданного режима работы. При считывании производится расшифровка адреса и с помощью блока считывания необходимое число из куба памяти пересылается на кодовые шины числа. При записи блок записи записывает код соответствующего числа в куб памяти. В режиме считывание-запись в кубе памяти после считывания восстанавливается только что считанное число.

Таким образом, тесты контроля ОЗУ должны включать в себя тесты контроля цепей записи и считывания чисел и адресов и тесты контроля куба памяти. В большинстве ЭВМ второго и третьего поколений куб памяти выполнен на тороидальных ферритовых сердечниках, поэтому такие ОЗУ получили название магнитных ОЗУ — МОЗУ. Эти сердечники под влиянием внешних условий, особенно температурных, могут изменять свои характеристики. Поэтому для проверки кубов памяти используются тесты, позволяющие проверить их в тяжелых режимах работы.

Для контроля цепей записи и считывания используются тесты, работающие по принципу сравнения проверочного кода с эталонным. Для этого ОЗУ разбивается на две части: неконтролируемую и контролируемую. В неконтролируемую записывается проверочный тест, а затем он переписывается в контролируемую. Проверка выполняется поразрядно при записи и считывании. Информация, записываемая в ячейки куба памяти, сравнивается с эталонным кодом (аналогично и при считывании). При несовпадении кодов сигнализация о неисправности цепи записи или считывания соответствующих разрядов выводится на пульт управления. Для проверки используются тесты, содержащие набор различных примеров и, в частности, тесты типа *бегущий нуль* и *бегущая единица*. *Бегущий нуль* — это группа кодов, в каждом из которых записаны единицы во все разряды, кроме одного, в котором записан нуль. *Бегущим нулем* этот тест называется потому, что в каждом коде местоположение нуля сдвинуто на один разряд:

```

01111 ... 11
10111 ... 11
11011 ... 11
11101 ... 11
. . . . .
11111 ... 01
11111 ... 10

```

Группа кодов *бегущая единица* представляет собой построенный по такому же принципу тест, где в каждом коде все разряды заняты нулями, а один — единицей, которая от кода к коду перемещается на один разряд:

```

10000 ... 00
01000 ... 00
00100 ... 00
. . . . .
00000 ... 01
    
```

Примеры этих тестов, включенных в систему ТЕСТ монитор, будут рассмотрены в следующем параграфе.

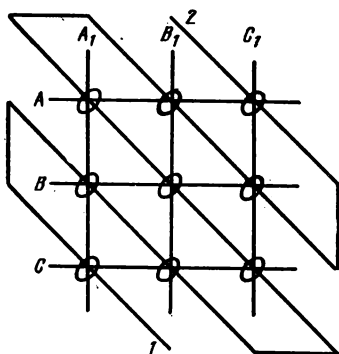


Рис. 6-14. Элемент МОЗУ матричного типа.

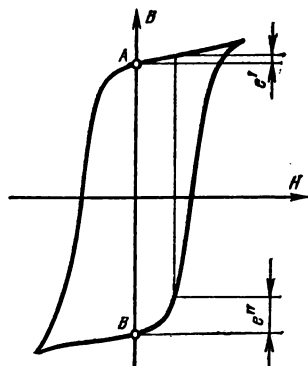


Рис. 6-15. Возникновение сигнала помехи.

Остановимся более подробно на особенностях проверки МОЗУ, связанных с использованием в качестве элементов памяти ферритовых тороидов или, как их принято называть в литературе, сердечников с ППГ (прямоугольной петлей гистерезиса). Различают две схемы построения МОЗУ — *матричную* и с *прямым выбором*, называемую иначе схемой Z.

В МОЗУ матричного типа (рис. 6-14) каждый элемент памяти хранит один разряд машинного слова. Матрица из девяти элементов, приведенная на рисунке, хранит один разряд девяти машинных слов или девяти ячеек МОЗУ. Матрица содержит координатные или адресные шины — горизонтальные и вертикальные  $A, B, C, A_1, B_1, C_1$ , позволяющие выбрать по заданному адресу нужный сердечник. Эти шины проходят через все матрицы МОЗУ. Обычно МОЗУ

содержит  $n$  таких матриц, где  $n$  — разрядность машинных слов, используемых в данной ЭВМ; в каждой матрице хранится один разряд  $m \times m$  машинных слов.

Для считывания информации используется одна на всю матрицу шина — шина 1, 2. Для считывания информации из заданного сердечника на соответствующие горизонтальную и вертикальную координатные шины подаются полуточки считывания. В момент совпадения импульсов по шине считывания проходит сигнал, если сердечник перемагничивается, т. е. если в нем была записана 1. Запись информации в ячейку также осуществляется при подаче токов по координатным шинам. Шина считывания при этом выполняет функцию запрета. Если сигнала запрет на шине считывания нет, то выбранный сердечник перемагничивается, т. е. в него записывается 1, если же запрет есть, то сердечник не перемагничивается, т. е. в нем будет записан 0.

В МОЗУ с прямым выбором каждый элемент памяти пересекается разрядной шиной, которая является общей для одного разряда всех машинных слов ОЗУ, и словарной шиной, которая является общей для всех разрядов одного машинного слова. При считывании по словарной шине подается мощный импульс считывания всех разрядов заданного числа и информация снимается с разрядных шин. Запись же новой информации в сердечники проводится в два этапа. На первом — выполняется очистка ячейки, в которую будет записано новое машинное слово. Для этого по словарной шине, соответствующей выбранному адресу, подается импульс, переводящий все сердечники данной ячейки в состояние 0.

На втором этапе запись 1 в соответствующий разряд машинного слова производится при совпадении импульсов полуточек в разрядной и словарной шинах.

Преимуществом схемы с прямым выбором является большая помехоустойчивость за счет более мощных импульсов тока, так как в этой схеме помехи значительно ниже за счет физической развязки цепей считывания каждого машинного слова. Недостатком этой схемы, ограничивающим ее повсеместное применение, являются неэкономичные затраты оборудования для адресного возбуждения словарных шин.

На работе МОЗУ матричного типа помехи сказываются значительно больше, чем в МОЗУ с Z-схемой.

Рассмотрим возникновение сигнала помехи в паре сердечников МОЗУ матричного типа.

Пусть положение сердечника при хранении в нем 0 отмечено точкой  $A$ , а при хранении 1 — точкой  $B$  (рис. 6-15). Пусть в паре взаимно компенсирующих сердечников записаны нули, тогда при подаче импульса считывания ЭДС помехи, выдаваемая этой парой сердечников,

$$e_n = e' - e' = 0,$$

где  $e'$  — ЭДС считывания 0.

Если же в сердечниках была записана 1, то ЭДС помехи при ее считывании

$$e_n = e'' - e'' = 0,$$

где  $e''$  — ЭДС считывания 1.

Однако если в одном из сердечников был записан 0, а в другом 1, то ЭДС помехи

$$e_n = e'' - e' \neq 0,$$

и эта разность будет тем большей, чем меньше прямоугольность петли гистерезиса у рассматриваемой пары сердечников. Таким образом, для проверки МОЗУ матричного типа необходимо сформировать тяжелый код, который будет записывать в каждую из пар компенсирующих сердечников соответственно 0 и 1.

Помехи возникают из-за непрямоугольности петли гистерезиса и в МОЗУ с Z-схемой, однако обычно они оказываются значительно ниже, чем в матричных МОЗУ.

Рассмотрим влияние помех на работу формирователей записи. При некоторых тяжелых кодах, записанных в МОЗУ, на обмотку записи со стороны сердечников может воздействовать значительная ЭДС самоиндукции (противо-ЭДС), снижающая импульс записи. И если параметры формирователя записи не будут в этом режиме соответствовать номинальным, то запись информации будет выполняться с ошибкой. Пусть, например, все сердечники матрицы (в МОЗУ матричного типа), заисключением того, в который необходимо произвести запись, находятся в состоянии 1, а в выбранную ячейку записывается 0. Тогда в смежных сердечниках возникают ЭДС, направленные встречно по отношению к приложенному напряжению. При этом их суммарная ЭДС может достигнуть большого значения и привести к снижению амплитуды импульса записи.

В МОЗУ, построенном по Z-схеме, тяжелая нагрузка на формирователи возникает в тех случаях, когда одноименные разряды всех ячеек содержат либо единицы, либо

нули. Тогда при записи в одну из ячеек нуля, если в остальных ячейках были записаны единицы, или единицы, если были записаны нули, возникает тяжелый режим работы. Таким образом, код, состоящий из одних единиц (нулей), создает тяжелый режим работы для формирователя тока записи при записи соответственно нуля (единицы) в одну из ячеек, так как возникающие при этом в ячейках ЭДС суммируются. Такая проверка позволяет выявить в основном неисправности в усилителях и формирователях, используемых в МОЗУ для считывания и записи информации.

Остановимся далее на проверке, позволяющей выявить ухудшение свойств самих сердечников под влиянием температурных нагревов или старения материала, из которого они изготовлены.

Одним из факторов, вызванных ухудшением свойств сердечника, является его размагничивание. Если в матричной схеме МОЗУ у одного из сердечников значительно нарушится прямоугольность петли гистерезиса (под влиянием температурного нагрева или старения материала), а в этом сердечнике была записана единица, то при многократном считывании информации из соседних сердечников уровень остаточной индукции в этом сердечнике значительно уменьшится и может оказаться на уровне помехи. В этом случае информация, записанная в сердечник, будет потеряна.

Выявление сердечников, начинающих терять свои свойства, производится в режиме профилактического контроля. Проверочные тесты создают последовательно для каждой ячейки МОЗУ матричного типа тяжелый режим. При этом в проверяемые ячейки записываются одни единицы, а в соседних ячейках выполняется многократное считывание. Если хотя бы один из сердечников проверяемой ячейки начинает терять свои свойства, то под воздействием тяжелого режима в нем произойдет стирание информации. После выполнения заданного количества циклов считывания содержимое проверяемой ячейки сравнивается с эталоном. При несовпадении кодов программа выходит на останов и сигнализирует о нем на пульте управления или на пишущей машинке. Если все в порядке, то проверяется следующая ячейка.

В современных ЭВМ для контроля правильности записи, считывания и восстановления информации всех ячеек МОЗУ очень часто используется тест *падающий дождь*. Идея этого теста заключается в следующем. В нулевую ячейку МОЗУ

записываются единицы во всех разрядах, например 11111111, а остальные ячейки очищаются, т. е. в них находятся нули по всем разрядам.

Содержимое  $i$ -й ячейки МОЗУ на  $n$ -такте пересчета определяется по формуле

$$(a_i)_n = (a_i)_{n-1} \oplus (a_{i-1})_n,$$

где  $\oplus$  — символ сложения по модулю 2, т. е.

$$(a_1)_1 = (a_1)_0 \oplus (a_0)_1,$$

или

$$(a_1)_1 = 11111111 \oplus 00000000 = 11111111.$$

Таким образом, если обозначить максимальный адрес МОЗУ через  $i = m$ , то через  $m$  тактов проверки во всех ячейках МОЗУ будут записаны единицы. Содержимое первой ячейки на  $(m + 1)$ -м такте будет составлять:

$$(a_1)_{m+1} = (a_1)_m \oplus (a_0)_{m+1};$$

$$(a_1)_{m+1} = 11111111 \oplus 11111111 = 00000000,$$

а содержимое второй ячейки

$$(a_2)_{m+1} = (a_2)_m \oplus (a_1)_{m+1};$$

$$(a_2)_{m+1} = 11111111 + 00000000 = 11111111$$

и т. д.

Таким образом, через  $2m$  пересчетов во все четные ячейки МОЗУ будут записаны единицы, а во все нечетные — нули.

Через  $m \times m$  пересчетов код, содержащий все единицы, останется только в последней,  $m$ -й ячейке МОЗУ. Если продолжить вычисления еще  $m^2$  раз, то код со всеми единицами будет в ячейках с адресами  $m$  и  $m - 1$ , еще после  $m^2$  изменений — в ячейках  $m$ ,  $m - 1$  и  $m - 2$  и т. д. Наконец, если выполнить пересчет  $m^3$  раз, то код, содержащий все единицы, окажется во всех ячейках МОЗУ, а эта картина наблюдается после  $m$  первых тактов проверки. Таким образом, через  $m^3 - m$  тактов пересчета результат повторяется. Тест *падающий дождь* получил свое название потому, что движение единиц на экране осциллографа при выполнении проверки ячеек напоминает дождь, поэтому он очень удобен для визуального наблюдения за ходом контроля. Так как в течение первых  $m^2$  тактов единица хранится в ячейке с адресом  $m$ , в течение следующих  $m^2$  тактов — в ячейке с адресом  $m - 1$  и т. д., то на экране осциллографа наблюдается движение единиц по экрану. В случае сбоя

равномерное перемещение единиц прекращается и по нарушению картинки можно определить, что не все ячейки МОЗУ работают нормально, однако локализовать место неисправности с помощью этого теста нельзя.

Кроме рассмотренного варианта реализации теста *падающий дождь*, являющегося наиболее распространенным, встречаются и другие разновидности этого теста. Например, в МОЗУ из  $m$  ячеек, содержащих по  $n$  разрядов каждая, записывается такой вариант теста: в первые  $n$  ячеек записываются коды, содержащие по одной единице в каждой ячейке, причем эти единицы смещены в ячейках на один разряд, т. е. в  $n$  ячейках расположены  $n$  кодов с единицей в каждом разряде, а в  $m - n$  ячейках ОЗУ записаны коды, состоящие из нулей. Далее алгоритм реализации теста тот же, что и в рассмотренном основном примере. Существуют и другие способы задания начальных кодов в ячейки МОЗУ, но алгоритм преобразования этих кодов с помощью операции сложения по модулю 2 везде остается одним и тем же.

#### 6-9. ТЕСТ ПРОВЕРКИ МОЗУ В СИСТЕМЕ ТЕСТ-МОНИТОР

Оперативная память в малых моделях ЕС ЭВМ состоит из трех функционально независимых видов памяти: основной (емкостью 256—1024 К), локальной (емкостью 256 байт) и мультиплексной (емкостью 768—1792 байт).

Рассмотрим тесты для проверки основной памяти, включенные в систему ТЕСТ-монитор. Для проверки правильности функционирования основной памяти МОЗУ ЕС ЭВМ применяются две тест-секции, работающие по принципу решения отдельных примеров и сравнения результатов решения с эталонными.

Первая тест-секция Е3910 включает в себя примеры, позволяющие контролировать работу блоков основной памяти в следующих режимах: *адресный код, запись и считывание нулей, запись и считывание единиц, бегающий ноль, бегающая единица, тяжелый код, шахматный код.*

Вторая тест-секция Е3920 содержит два режима контроля основной памяти: *дождь* и *случайные числа.*

Остановимся кратко на тестах контроля основной памяти. Вся работа тест-секций организуется управляющей программой, которая автоматически, если пример выполнен правильно, передает управление следующему примеру и при необходимости прерывает выполнение тест-секций при получении соответствующих команд от схем контроля.



Если при выполнении любого примера управляющая программа обнаружит ошибку, то она передает управление подпрограмме «Печать сообщения об ошибке», которая выводит следующее стандартное, одинаковое для всех примеров сообщение

АДРЕС	РЕЗУЛЬТАТ	ЭТАЛОН
AAAAAA	XX XX XX XX	YY YY YY

где AA...A — абсолютный адрес ячейки в основной памяти; XX...X — полученный результат; YY...Y — эталонный результат.

Это сообщение означает, что при проверке ячейки памяти с адресом AA...A при сравнении результата решения данного примера с эталонным ответом произошло несовпадение значений.

Все эталонные ответы записаны в виде специального блока констант, входящего в состав теста основной памяти. Константы в этом блоке формируются с помощью оператора определения констант DC.

Режим *адресный код* служит для проверки цепей записи и считывания адресной части команд. Работает этот пример следующим образом: вначале выделяется проверяемая область основной памяти, т. е. определяется ее начальная и конечная абсолютные адреса. Затем в младшие разряды ячеек этой области записываются их абсолютные адреса. В режиме считывания производится сравнение содержимого ячеек со значением абсолютных адресов этих ячеек. Если результаты не совпали, то печатается стандартное сообщение об ошибке, если совпали, то выделяется новая область в основной памяти и проверка продолжается.

В режимах записи нулей и единиц в выделенной области основной памяти записывается во все ячейки соответственно код 00000000 или FFFFFFFF. Далее производится считывание информации из этих ячеек и сравнение ее с эталонной. Если результаты не совпадают, то печатается сообщение об ошибке, если же совпадают, то программа передает управление следующему примеру.

В режимах *бегающий ноль* и *бегающая единица* проверяется работа выделенной области памяти. Проверка выполняется следующим образом. Вначале во все ячейки выделенной для проверки области памяти записывается код бегающего нуля 7FFFFFFF. Далее из каждой ячейки производится считывание записанной в ней информации и сравнение ее с эталоном, формируемым в блоке констант. Если результаты

совпадают, то считывается информация из следующей ячейки, а если нет, то печатается сообщение об ошибке. В режиме бегающей единицы работа по проверке организована аналогично, только вместо кода бегающего нуля используется код бегающей единицы — 80000000.

Следующие четыре примера этой тест-секции служат для проверки правильности работы ячеек основной памяти в режимах *тяжелый код* и *обратный (инверсный) тяжелый код*. Проверка в этих примерах построена по одинаковому принципу, поэтому рассмотрим только один пример — проверка в режиме тяжелого кода.

Эта проверка выполняется в зависимости от содержимого 4-го и 13-го разрядов заданного абсолютного адреса ячейки основной памяти. Обозначим:  $A$  — значение 40-го разряда абсолютного адреса ячейки основной памяти;  $B$  — значение 13-го разряда абсолютного адреса ячейки основной памяти. Запись информации в каждую ячейку проверяемой области памяти проводится по следующему принципу. Если  $A + B = 1$ , то записывается число 00000000, если  $A + B = 0$ , то записывается число FFFFFFFF.

Далее выполняется считывание информации из всех ячеек проверяемой области памяти и сравнение с эталонной.

Последней проверкой, которая выполняется этой тест-секцией, является проверка в режиме *шахматный код*. Этот пример работает следующим образом. Во все ячейки проверяемой области памяти первоначально записывается код AA55AA55, представляющий собой чередование единиц и нулей (10101010010101010101001010101). Далее, в первой ячейке выделенной для проверки области памяти выполняется инвертирование первых четырех байтов данных. Эта ячейка в дальнейшем будет называться *приемником*. Затем из третьей ячейки выделенной области памяти пересылается в первую записанный код AA55AA55. Ячейка, из которой пересылается информация, будет в дальнейшем называться *передатчиком*. Далее выполняется обратная пересылка данных из приемника в передатчик и производится сравнение информации, прошедшей двойную пересылку, с эталоном. Если ошибки нет, то адрес передатчика увеличивается на четыре и вся операция повторяется снова. Далее вновь увеличивается на четыре адрес передатчика и производится обмен информацией между новым передатчиком и старым приемником. Так продолжается до тех пор, пока адрес передатчика не станет равен адресу последней части проверяемого участка памяти. После этого программа

увеличивает на единицу адрес приемника и вся процедура проверки выполняется вновь, пока в качестве приемника не будут просмотрены все ячейки выделенной для проверки области памяти.

После окончания выполнения проверки во все ячейки выделенной области памяти записывается инверсный предыдущему код 55AA55AA и вся описанная выше процедура повторяется для этого нового кода.

Вторая тест-секция E3920 выполняет две проверки в режимах дождя и записи и считывания случайных чисел.

Проверка в режиме падающего дождя в данном случае выполняется следующим образом: в первую ячейку выделенной области памяти, обозначенную через  $N$ , записывается шестнадцатиричная константа  $X'FFFFFFFFFFFFFFFF'$ , т. е. двойное слово, состоящее из одних единиц. Затем выполняется сложение по модулю 2 содержимого ячейки с номером  $N$  и содержимого ячейки с номером  $N + 8$ . Результат при первом просмотре должен равняться соответственно  $FFFFFFFFFFFFFFFF$ . При следующих циклах решения результат сложения содержимого  $N$ -й и  $(N + 8)$ -й ячеек может равняться как  $FFFFFFFFFFFFFFFF$ , так и  $0000000000000000$ . После выполнения сложения для соответствующего цикла решения формируется эталонный ответ и сравнивается с полученным результатом сложения содержимого двух ячеек. Если эталонный и полученный ответы совпадают, то результат сложения записывается в ячейку с адресом  $N + 8$ , если нет, то печатается сообщение об ошибке. На следующем этапе выполняется сложение по модулю 2 содержимого  $(N + 8)$ -й и  $(N + 16)$ -й ячеек выделенной области памяти. Так продолжается до окончания просмотра всех ячеек. Затем цикл повторяется, но с использованием того содержания ячеек, которое было получено в предыдущем цикле вычислений. Всего выполняется 15 циклов решения примера.

Режим *записи и считывания случайных чисел* служит для последней (при безошибочном прохождении предыдущих примеров) и наиболее логически сложной проверки ячеек основной памяти (в настоящей книге не приводится).

## 6-10. ПРОВЕРКА ВНЕШНИХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ

В современных ЭВМ в качестве внешних запоминающих устройств (ВЗУ) наиболее часто используются накопители на магнитных лентах (НМЛ) и накопители на магнитных

дисках (НМД). Эти два вида ВЗУ входят обычно в основные комплекты машин третьего поколения, а в качестве дополнительного оборудования эти машины могут быть доукомплектованы наиболее быстродействующими среди ВЗУ накопителями на магнитных барабанах.

Основными задачами, стоящими при проверке ВЗУ, являются: проверка цепей записи и считывания информации, проверка команд управления работой ВЗУ, проверка различных режимов работы устройств. Так как в современных ЭВМ в комплект машины входит несколько накопителей, управляемых одной стойкой управления (например, в машинах ЕС-1022, ЕС-1030 одна стойка управления лентами приходится на группу от четырех до восьми НМЛ, а стойка управления дисками на два, три или четыре НМД), то в тестах предусматривается возможность проверки как индивидуальной работы каждого НМЛ и НМД со своей стойкой, так и групповой работы устройств. Проверка осуществляется по тем же принципам, что и проверка работы процессора и ОЗУ. В тесты включается группа примеров, которые записываются в заданные ячейки ОЗУ и на заданное место в ВЗУ. Затем производится считывание информации из ВЗУ в свободные ячейки ОЗУ и коды сравниваются. При несовпадении печатается информация об ошибке.

Перед контролем записи и считывания в НМЛ обычно проверяются цепи разметки зон. При этом в предварительно размеченные зоны на магнитной ленте записывается проверочный код (чаще всего это номер зоны), далее этот код считывается в свободные ячейки ОЗУ и выполняется сравнение с эталоном.

Рассмотрим тесты для контроля накопителей на магнитной ленте и на магнитных дисках, используемые в системе ТЕСТ-монитор для ЕС ЭВМ.

Функциональный тест проверки магнитной ленты в системе ТЕСТ-монитор состоит из десяти тест-секций (F5010, F5020, F5030, F5040, F5050, F5060, F5070, F5080, F50F0 и F5100), включающих 17 подпрограмм и 58 примеров.

Текст-секции разбиты на две неравные группы, в первую из них входит девять тест-секций, кроме F5100, а во вторую — F5100. Все тест-секции первой группы имеют общие подпрограммы и служат для организации проверки выполнения команд и реакции устройств ввода-вывода на сбои, а также для проведения различных проверок реакции НМЛ на маркер конца ленты. Тест-секция F5100 служит для проверки взаимозаменяемости НМЛ. Она обслуживается спе-

циальными подпрограммами, и информация об ошибках, выдаваемая ею, отличается по форме от сообщений тест-секций первой группы.

Не вдаваясь в детальное рассмотрение структуры всех тест-секций функционального теста проверки НМЛ, рассмотрим для примера состав и назначение одной из них — тест-секции F5010.

Тест-секция F5010 используется для проверки ряда команд управления работой НМЛ и состоит из десяти примеров.

В примере 1 проверяется состояние устройства после выполнения команды ПЕРЕМОТАТЬ. По команде ПЕРЕМОТАТЬ в заданном накопителе на магнитную ленту происходит перемотка ленты назад до маркера начала ленты.

Вначале для проверки включается подпрограмма инициализации, которая анализирует готовность выбранного для проверки устройства к работе. Кроме того, эта подпрограмма анализирует плотность записи на магнитную ленту, т. е. 8 строк/мм или 32 строк/мм.

После этого выполняется подпрограмма перемотки ленты и программа переходит к подпрограмме, подготавливающей сообщение об ошибке, т. е. ожидаемые начальный и конечный байты состояния и адрес команды, при выполнении которой произошла ошибка.

Далее выполняется команда УТОЧНИТЬ СОСТОЯНИЕ и проводится сравнение полученного результата с эталонным. Если они не совпадают, то управление передается подпрограмме анализа ошибок, которая подготавливает и печатает сообщение об ошибке. Команда УТОЧНИТЬ СОСТОЯНИЕ передает в канал 6 байтов уточненного состояния. Каждый разряд байта уточненного состояния (БУС) указывает состояние соответствующих цепей устройства управления внешними устройствами (УВУ) и НМЛ или ошибки, обнаруженные во время выполнения предыдущей операции. Наименование и условия формирования разрядов байтов уточненного состояния приведены в табл. 6-1.

Если после выполнения команды УТОЧНИТЬ СОСТОЯНИЕ ошибки нет, то управление передается командам последовательного сравнения 1, 2, 3, 4 и 5-го байтов уточненного состояния устройств ввода-вывода с ожидаемыми результатами. Если все проверки благополучно завершаются, то управление передается подпрограмме выхода из примера. Если же результаты не совпали, то печатается сообщение об ошибке.

В примере 2 проверяется команда СТЕРЕТЬ ПРОМЕЖУТОК. По команде СТЕРЕТЬ ПРОМЕЖУТОК выбранный накопитель передвигает ленту вперед приблизительно на 95 мм и стирает содержимое этого промежутка. Эта команда используется для пропуски поврежденного участка ленты, на котором не удается правильно записать данные. Для проверки используется тот факт, что при выполнении этой программы магнитная лента должна продвигнуться. В программе используются те же подпрограммы, что и в предыдущем примере. Поэтому, не останавливаясь на подробном описании порядка их выполнения, рассмотрим только сущность проверки.

После выполнения подпрограммы инициализации и перемотки магнитная лента устанавливается в точку начала ленты. Далее выполняется команда СТЕРЕТЬ ПРОМЕЖУТОК, а затем с помощью подпрограммы уточнения состояния проверка состояния первого БУС

Таблица 6-1

## Байты уточненного состояния НМЛ

Разряд байта	Наименование разряда	Условия формирования
<i>Нулевой байт</i>		
0	Команда не отвергнута	<p>Устанавливается в следующих случаях:</p> <p>команда <b>ЗАПИСАТЬ</b>, <b>ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН</b>, <b>СТЕРЕТЬ ПРОМЕЖУТОК</b> адресованы к накопителю, на котором запись запрещена</p> <p>получена команда, не относящаяся к данному устройству;</p> <p>канал адресует команду <b>СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ</b> тому ВУ, который не имеет оборудования для считывания при движении назад</p> <p>канал выдает команду <b>ЗАРЕЗЕРВИРОВАТЬ</b> или <b>ОСВОБОДИТЬ</b> в устройство, когда определена цепочка команд</p>
1	Требуется вмешательство	<p>Выбранное ВУ не существует или находится в состоянии неготовности, указанном первым разрядом первого байта уточненного состояния. Если ВУ становится не готовым в ходе выполнения операции, то в байте состояния устанавливаются признаки «Канал кончил», «ВУ кончило», «Сбой в устройстве»</p>
2	Ошибка на шинах канала	<p>Неправильная четность на информационных шинах канала во время передачи команды или данных</p>
3	Сбой оборудования	<p>В четвертом байте уточненного состояния установлен в единицу хотя бы один из разрядов</p>
4	Ошибка в данных	<p>В третьем байте уточненного состояния установлен хотя бы один из следующих разрядов: 0, 1, 2, 3, 4, 7 или установлен нулевой разряд в первом байте уточненного состояния</p>

Разряд байта	Наименование разряда	Условия формирования
5	Переполнение	Во время выполнения команд СЧИТАТЬ, СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ канал не может принять представленные ему данные, а во время выполнения команды ЗАПИСАТЬ не выдает требуемые данные в установленном интервале времени. При этом условия передача данных прекращается и выполнение операции заканчивается. Если возник сигнал «Сбой в информации», установка признака «Переполнение» блокируется
6	Нуль счетчика слов	Устанавливается при выполнении команды ЗАПИСАТЬ, если передача прекращается до выдачи первого байта данных. При этом условия лента НМЛ не начинает двигаться
7	—	Не используется
<i>Первый байт</i>		
0	Помеха в промежутке	Во время выполнения команд СЧИТАТЬ, ШАГ НА ЗОНУ ВПЕРЕД, ШАГ НА ГРУППУ ЗОН ВПЕРЕД указывает, что данные были обнаружены после строки ПКС, но до начала следующей зоны. Данные до строки ПКС контролируются и передаются в канал. Данные, считанные после строки ПКС, устанавливают признак «Помеха в промежутке, движение ленты продолжается, но передача данных в канал не происходит»
1	Состояние А НМЛ	Адресуемый накопитель находится в состоянии готовности
2	Состояние Б НМЛ	Адресуемый накопитель находится в состоянии неготовности. При этом устанавливается признак «Сбой в устройстве», за исключением случая, когда к неготовому ВУ адресована команда УТОЧНИТЬ СОСТОЯНИЕ. Одновременная установка первого и второго разрядов первого байта уточненного состояния означает, что НМЛ занят выполнением перемотки. В этом случае устанавливается признак «Занято»
3	—	Не используется

Разряд байта	Наименование разряда	Условия формирования
4	Начало ленты	Лента на выбранном ВУ находится на маркере НЛ
5	Состояние записи НМЛ	Выбранное ВУ находится в состоянии записи
6	Защита записи	Лента в выбранном ВУ защищена от записи, то есть в подающей катушке отсутствует кольцо
7	—	Не используется

*Второй байт*

		Этот байт содержит номер ошибочной дорожки. Если при выполнении команд СЧИТАТЬ, СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ произошел сбой в информации и была определена сбойная дорожка, то байт содержит единицу только в одном разряде, указывая номер этой дорожки. Если при выполнении вышеуказанных команд не произошел сбой в информации или сбой был, но сбойная дорожка не была найдена, то в этом байте устанавливаются в единицу 6-й и 7-й разряды
--	--	---

*Третий байт*

0	Ошибка четности в регистре записи-считывания	При выполнении команд СЧИТАТЬ, СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ определена ошибка четности в регистре записи-считывания. Этот признак не устанавливается при переполнении и после приема указания «Останов»
1	Ошибка в регистре продольного контроля	При выполнении команд ЗАПИСАТЬ, ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН, СЧИТАТЬ И СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ определена ошибка в регистре продольного контроля
2	Перекос	Чрезмерный перекося обнаружен при контрольном считывании в командах ЗАПИСАТЬ, ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН
3	Ошибка в регистре циклического контроля	Ошибка в регистре циклического контроля обнаружена при выполнении команд СЧИТАТЬ или СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ



Разряд байта	Наименование разряда	Условия формирования
4	Ошибка четности в регистре перекоса	При выполнении команд ЗАПИСАТЬ, ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН обнаружена ошибка четности в регистре перекоса
5		Не используется
6	Состояние движения назад	В выбранном накопителе лента движется в обратном направлении
7	Ошибка контроля сравнением	Четность байта данных на входе регистра информации и на его выходе не совпадает
<i>Четвертый байт</i>		
0	Эхо-ошибка	Ни один из триггеров записи в накопителе не изменил своего состояния
1	НМЛ отвергнут	Выбранный накопитель неправильно выдает ответ на сигнал из УВУ «Установить состояние записи» или «Установить состояние воспроизведения» или становится неготовым во время выполнения операции, включающей движение ленты
2	Ошибка синхронизатора считывания	Устанавливается при неверном счете синхронизатора считывания во время выполнения операции считывания или контрольного считывания при записи
3	Ошибка синхронизатора записи	Устанавливается при неверном счете синхронизатора записи во время выполнения операции записи
4	Ошибка счетчика задержек	Устанавливается при неверном счете счетчика задержек в операциях с движением ленты
5	Индикатор последовательности В	
6	Индикатор последовательности Б	Наличие одного из этих признаков сигнализирует о том, что нарушилась последовательность выполнения операции
7	Индикатор последовательности А	
<i>Пятый байт</i>		
		Не используется (УВУ передает нули во всех разрядах этого байта)

устройства ввода-вывода. После выполнения команды СТЕРЕТЬ ПРОМЕЖУТОК бит начала ленты в первом БУС должен установиться в 0, если этого не произошло, то печатается сообщение об ошибке.

Пример 3 служит для проверки команды ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН. Команда ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН вырабатывает байт маркера группы зон и строку продольного контроля (ПКС) и передает их для записи в выбранный НМЛ. Напомним, что информация записывается на магнитную ленту побайтно. Каждый байт, параллельно записываемый на девять дорожек ленты, состоит из восьми информационных битов и одного контрольного. Для записи контрольного разряда используется четвертая дорожка. Группа байтов, составляющая одно сообщение, начало и конец которого определяются в специальном командном слове канала (КСК), образует зону. Минимальная длина зоны — 18 байт, максимальная — принципиально ограничена длиной ленты (при плотной записи 32 бит/мм — 22 Мбайт). Рекомендуется максимальная длина зоны — 2048 байт. В операциях считывания одна зона определяется как одна физическая запись. Для считывания каждой зоны требуется отдельная команда канала. При плотности записи 32 бит/мм в конце зоны записываются две контрольные строки: строка циклического контроля (ЦКС) и ПКС. Контрольные программные строки формируются устройством управления во время записи информации. Для формирования ПКС ведется подсчет единиц на каждой дорожке зоны. Их общее число на любой дорожке должно быть четным. Это делается путем записи нуля или единицы в соответствующем разряде ПКС. Строка ПКС одновременно является признаком конца зоны. Зоны разделяются промежутками, необходимыми для разгона и торможения ленты. Межзонный промежуток должен быть не менее 12,7 мм. Обычно этот промежуток выбирают с некоторым запасом и устанавливают равным 15,2 мм. Группы зон могут разделяться маркером группы зон, который представляет собой однобайтовую зону с контрольной строкой ПКС. Маркер группы зон записывается на ленту в виде отдельной зоны, т. е. от других записей он отделяется межзонными промежутками.

Проверка выполняется на основе установления факта, что после выполнения данной команды магнитная лента продвинулась.

После выполнения подпрограммы инициализации и перемотки магнитная лента устанавливается в точку начала ленты. Далее выполняется команда ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН. И затем после выполнения подпрограммы уточнения состояния проверяется наличие нуля в бите начала ленты первого БУС проверяемого устройства. Если там записан не 0, то печатается сообщение об ошибке.

В примере 4 проверяется правильность выполнения команды ПЕРЕМОТАТЬ. После выполнения подпрограммы инициализации выполняется команда СТЕРЕТЬ ПРОМЕЖУТОК, а затем подпрограмма ПЕРЕМОТАТЬ с помощью счетчика циклируется и выполняется 5 раз. После каждой проверки анализируется байт состояния (БС) проверяемого устройства. Если при одной из проверок полученный и эталонный результаты не совпадают, то печатается сообщение об ошибке.

В примере 5 проверяется выполнение команды СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ над маркером группы зон. Команда СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ принимает данные из НМЛ и передает их в канал ввода-вывода, проверяет данные на четность и указывает обнаруженные ошибки. Операция выполняется до тех пор, пока будет считана информация всей зоны. При достижении межзонного промежутка лента останавливается. Особенностью выпол-

нения этой команды является то, что поскольку лента движется в обратном направлении, то контрольные строки считываются прежде, чем байты данных. Считываемые байты данных размещаются каналом в основной памяти в порядке убывания адресов.

В этом примере последовательно выполняются команды ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН, а затем СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ. Проверка правильности выполнения команды выполняется на основе сравнения ожидаемых и полученных данных обратного чтения. При выполнении программы возможен особый случай, поэтому в программе два вида сообщений об ошибке: сообщение 05. В БС устройства не установился в единицу бит особого случая. Сообщение 06 выглядит так:

```
XXXXXXXXX
YYYYYYYYY
СООБЩЕНИЕ 06
```

где ХХ...Х — эталонное значение; УУ...У — данные считывания в обратном направлении.

Сообщение 06 дает сведения об ошибке при выполнении команды СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ на зоне, содержащей маркер группы зон.

В примере 6 проверяется выполнение команд СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ и СЧИТАТЬ. Команда СЧИТАТЬ выполняется аналогично команде СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ, за исключением того, что лента движется в прямом направлении и контрольные строки будут считываться нормально, т. е. после байтов данных.

Проверка этих команд производится на основе сравнения результатов выполнения команды ЗАПИСАТЬ с данными обратного и прямого чтения.

В примере 7 проверяется правильность выполнения цепочки команд. При проверке выполняется цепочка из команд ЗАПИСАТЬ, ЗАПИСАТЬ, ЗАПИСАТЬ, затем анализируется записанный адрес. Команда ЗАПИСАТЬ осуществляет прием данных из канала и передачу их в выбранный НМЛ для записи на ленту, считывает и проверяет на четность записанные ранее байты, а также устанавливает соответствующие индикаторы при обнаружении ошибок. Передача каждого байта данных из канала производится по запросам УВУ. Устройство проверяет эти байты на четность. Во время записи УВУ формирует из байтов данных контрольные строки ЦКС (при плотности 32 бит/мм) и ПКС и записывает их на ленту после последнего байта данных. Число записываемых на ленту байтов по одной команде ЗАПИСАТЬ определяется каналом.

В примере 8 проверяется выполнение команды ШАГ НА ЗОНУ НАЗАД. При команде ШАГ НА ЗОНУ НАЗАД проверяемый НМЛ перемещает ленту назад до последнего промежутка между зонами или до промежутка между маркером начала ленты и первой зоной на ленте.

Для проверки вначале выполняется цепочка команд ЗАПИСАТЬ, затем последовательно выполняются команды ШАГ НА ЗОНУ НАЗАД и СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ. Результат считывания в обратном направлении сравнивается с записанными данными.

В примере 9 проверяется выполнение команды ШАГ НА ЗОНУ НАЗАД на зоне, содержащей маркер группы зон.

При проверке последовательно выполняются команды ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН и ШАГ НА ЗОНУ НАЗАД. Полученное

в результате выполнения команды ШАГ НА ЗОНУ НАЗАД значение байта состояния проверяемого устройства сравнивается с эталонным.

В примере 10 проверяется выполнение команды ШАГ НА ЗОНУ НАЗАД на зоне, содержащей маркер начала ленты.

Для проверки выполняются последовательно команды ПЕРЕКЛОПАТЬ и ШАГ НА ЗОНУ НАЗАД на зоне, содержащей маркер начала ленты. Полученный в результате выполнения команды ШАГ НА ЗОНУ НАЗАД байт состояния устройства сравнивается с эталонным значением.

Примеры в остальных тест-секциях теста функциональной проверки магнитной ленты построены аналогично примерам в тест-секции F5010.

Прежде чем рассматривать тесты контроля накопителей на магнитных дисках, кратко остановимся на принципах размещения информации в типовом пакете сменных дисков типа ЕС-5053.

Пакет предусматривает запись по 200 основным и 3 запасным дорожкам на каждой поверхности диска. Дорожки нумеруются от внешней к внутренней, поэтому самая первая внешняя имеет номер 000, а крайние внутренние, которые используются в качестве запасных, имеют номера 200, 201 и 202. Для контроля используется дорожка с адресом 073. На каждой дорожке может быть записано последовательно бит за битом 3625 байтов информации. Так как диаметры дорожек различны, а количество байт информации на каждой из них одинаково, то плотность записи меняется от дорожки к дорожке, от 30 бит/мм на внешней дорожке до 44 бит/мм на крайней внутренней. Десять дорожек, расположенных друг под другом на рабочих поверхностях десяти дисков, носят название цилиндров. Емкость одного цилиндра при плотной записи информации составляет 36 250 байт, а емкость всего пакета — 7,25 Мбайт. Для записи и считывания информации накопитель имеет десять магнитных головок, каждая из которых состоит из универсальной (для записи и воспроизведения) головки и головки стирания, которые размещаются в одном корпусе. Магнитные головки располагаются друг под другом и укрепляются на каретке, которая может перемещать их в радиальном направлении по отношению к дискам. Каретка может фиксироваться в одном из 203 положений, подводя головки к одному из цилиндров. Одновременно запись или считывание информации ведет только одна головка из десяти.

Для удобства поиска записанной на диске информации запись данных на каждой дорожке осуществляется вполне определенным образом. При записи вся дорожка разделя-

ется на ряд участков (рис. 6-16). В начале каждой дорожки после пробела  $\alpha_1$  записывается собственный адрес. Этот адрес содержит адрес цилиндра и адрес головки для этой дорожки. Далее идет пробел  $\alpha_2$ , а за ним помещается запись  $R_0$  — описатель дорожки. Далее через пробелы  $\alpha_3$  следуют записи данных  $R_1, R_2, \dots, R_n$ . Для проверки достоверности записи и считывания данных устройство УВУ вычисляет два байта циклического контроля и записывает их в конце каждой записи. При чтении записи контрольные биты восстанавливаются, а байты циклического контроля вновь вычисляются и сравниваются с записанными. Байты циклического контроля (ЦК) получаются путем сложения по модулю два четных (байт 1ЦК) и нечетных (байт 2ЦК) байтов записи.

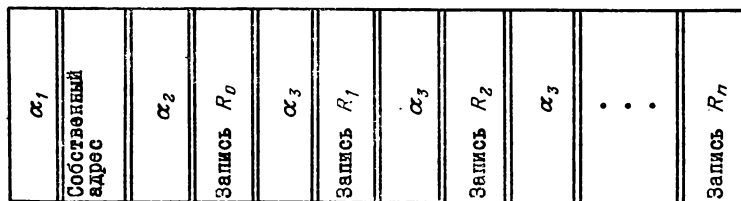


Рис. 6-16. Формат дорожки сменного диска.

Собственный адрес состоит из 7 байт, из них во 2-й и 3-й записывается адрес цилиндра, в два следующих — адрес головки, в 6-й и 7-й 1ЦК и 2ЦК. Первый байт собственного адреса носит название флагового байта. Во флаговом байте первые шесть разрядов (с 0-го по 5-й) всегда нулевые. Шестой разряд устанавливается в единичное состояние, если дорожка дефектная, а 7-й — если она запасная.

Все записи пользователя  $R_i$  ( $i = 1, 2, \dots, n$ ) имеют одинаковый формат (рис. 6-17). В каждой из них перед полем данных располагается еще поле счета и поле ключа, разделенные пробелом. В поле счета байты СС и НН указывают адрес цилиндра и головки для данной дорожки. В остальных байтах записывается:

порядковый номер записи (он меняется от 1 до 255);  
число байтов поля ключа без байтов циклического контроля;

число байтов поля данных без байтов циклического контроля (это число может изменяться от 0 до 65 535 байт).

Пять байтов в поле счета, в которых указываются адрес цилиндра, головки и порядковый номер записи на дорожке,

называются идентификатором. Он используется для поиска записи.

В поле ключа может быть записана любая информация, но обычно содержимое поля ключа используется для характеристики информации, помещенной в поле данных этой записи. Поле ключа содержит от 0 до 255 байт. Поле ключа используется для организации формата записи данных. Это позволяет быстрее отыскивать нужные данные на носителе.

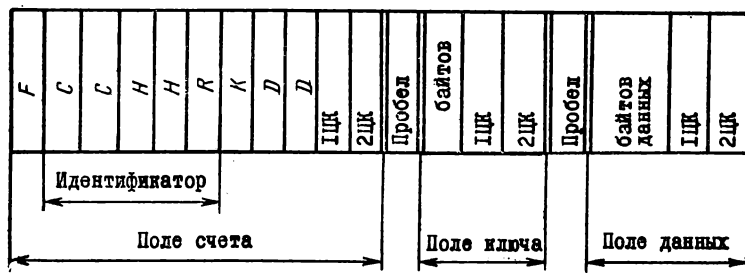


Рис. 6-17. Формат записи пользователя.

Запись  $R_0$  отличается от всех записей  $R_i$  ( $i = 1, 2, \dots, n$ ) тем, что она не имеет поля ключа, а ее поле данных состоит из 8 байт (не считая байтов циклического контроля) и содержит информацию о том, что записано на дорожке. В частности, запись  $R_0$  указывает порядковый номер последней записи на дорожке.

Пробелы  $\alpha_i$  ( $i = 1, 2, 3$ ) служат для разделения записей. Пробел  $\alpha_1$  состоит из 36 байт, пробел  $\alpha_2$  — из 18. Пробел  $\alpha_3$  имеет переменную длину, начиная от 27 байт, и отличается наличием двух сигнальных байтов, называемых адресным маркером. Адресный маркер при чтении информации сигнализирует о месте начала записи.

Тестовый контроль накопителя на сменных магнитных дисках состоит в проверке нормального функционирования различных команд, служащих для управления работой НМД. Появление сбоя при выполнении любой команды передается в устройство управления с помощью БС НМД и БУС НМД. Однако, прежде чем рассматривать конкретную информацию, которую несут разряды БС и БУС, остановимся кратко на основных командах, с помощью которых осуществляется управление записью, считыванием и обработкой информации в НМД.

Все команды управления работой НМД разделяются на пять групп: команды *установки*, команды *поиска*, команды *чтения*, команды *записи* и *управления*.

В группу команд *установки* входят следующие команды: УСТАНОВКА, УСТАНОВКА ЦИЛИНДРА, УСТАНОВКА ГОЛОВКИ и ОРИЕНТАЦИЯ. Все эти команды используются для выбора цилиндра и дорожки в накопителе. Например, по команде ОРИЕНТАЦИЯ магнитные головки НМД устанавливаются на дорожке с адресом 0 цилиндра с адресом 000.

Группа команд *поиска* используется для нахождения нужной записи на дорожке или на цилиндре. Аргументом поиска может быть идентификатор, ключ записи, ключ и данные, собственный адрес дорожки. Рассмотрим работу некоторых команд поиска.

По команде ПОИСК ПО РАВЕНСТВУ СОБСТВЕННОГО АДРЕСА устройство управления ищет индекс начала дорожки, считывая адрес цилиндра и адрес головки из собственного адреса, и сравнивает их последовательно с четырьмя байтами, получаемыми из основной памяти машины.

Команда ПОИСК ПО КЛЮЧУ предназначена для поиска записи по значению ее ключа. Если этой команде предшествовала команда ПОИСК ПО ИДЕНТИФИКАТОРУ, то устройство управления осуществляет сравнение с ключом записи, идентификатор которой считывался предыдущей командой.

Команда ПОИСК ПО КЛЮЧУ И ДАННЫМ служит для поиска записи по значению поля ключа и поля данных. С помощью этой команды реализуется так называемое сканирование набора данных, позволяющее осуществить поиск в файле записи по какой-то ее части или полю.

Группа команд *считывания* предназначена для чтения информации и передачи ее из НМД в ОЗУ.

Команда СЧИТАТЬ СОБСТВЕННЫЙ АДРЕС выполняет поиск индекса начала дорожки, чтение собственного адреса и передачу 5 байт адреса в канал.

По команде СЧИТАТЬ СЧЕТ производится чтение и передача в канал содержимого поля счета записи. По этой команде производится поиск на дорожке очередного адресного маркера, считывание поля счета и передача в канал 8 байт этого поля.

По команде СЧИТАТЬ ДАННЫЕ считывается и передается из накопителя в основную память информация из поля данных записи.

Команда СЧИТАТЬ ПЗП служит для установки цилиндра с адресом 000 и головки с адресом 0 в накопителе, к которому выполняется обращение. После установки выполняется поиск индекса и считывание содержимого поля данных записи  $R_1$ . Эта операция называется первоначальной загрузкой программы (ПЗП).

Группа команд *записи* служит для передачи информации из основной памяти в накопитель и записи ее на диски.

Команда ЗАПИСАТЬ СОБСТВЕННЫЙ АДРЕС записывает в поле собственного адреса дорожки 5 байт, полученных из основной памяти, и 2 байт циклического контроля. Выполнение этой команды начинается после индекса начала дорожки и пробела  $\alpha_1$ .

Команда ЗАПИСАТЬ ДАННЫЕ записывает данные из канала в ранее сформированное поле данных. Эта команда используется совместно с предшествующими ей командами поиска типа ПОИСК ПО РАВЕНСТВУ КЛЮЧА.

По команде ЗАПИСАТЬ СЧЕТ записывается информация в поля счета, ключа и данных записи. Перед полем счета записывается пробел  $\alpha_3$  с адресным маркером. Этой команде должны предшествовать команды поиска типа ПОИСК ПО РАВЕНСТВУ КЛЮЧА.

К группе команд *управления* относятся такие команды, как УПРАВЛЯТЬ ОБЛАСТЬЮ СЧЕТА, УСТАНОВИТЬ МАСКУ и др.

Команда УПРАВЛЯТЬ ОБЛАСТЬЮ СЧЕТА позволяет при чтении пропустить дефектную область адреса дорожки или прочитать данные из записи с дефектным полем счета.

Команда УСТАНОВИТЬ МАСКУ применяется для запрета выполнения команд типов установки и записи. Устройство управления получает по этой команде один байт, передающий информацию о запрете выполнения соответствующих команд. Нулевой и первый разряды этого байта имеют следующие значения:

00 — разрешены все команды записи, за исключением команд ЗАПИСАТЬ СОБСТВЕННЫЙ АДРЕС и ЗАПИСАТЬ  $R_0$ ;

01 — запрещены все команды записи;

10 — разрешены только команды ЗАПИСАТЬ ДАННЫЕ и ЗАПИСАТЬ КЛЮЧ и ДАННЫЕ;

11 — разрешены все команды записи;

Третий и четвертый разряды этого байта имеют следующие значения:

00 — разрешены все команды установки;



01 — разрешены команды УСТАНОВКА ЦИЛИНДРА и УСТАНОВКА ГОЛОВКИ;

11 — запрещены все команды установки.

Все остальные разряды байта, т. е. 2, 5, 6 и 7-й должны быть равны нулю.

Команду УСТАНОВИТЬ МАСКУ в каждой цепочке команд можно использовать только один раз.

Как уже отмечалось выше, контроль за выполнением всех команд осуществляется с помощью байтов состояния (БС) НМД (табл. 6-2) и байтов уточненного состояния (БУС) НМД (табл. 6-3).

Текст накопителя на сменных магнитных дисках состоит из четырех тест-секций. Тест-секции F6100 служат для организации функциональной проверки накопителя, F6110 — для проверки механизма доступа, F6120 — для проверки записи и чтения и F6140 — для проверки взаимозаменяемости накопителя.

Все тест-секции состоят из нескольких примеров и стандартных подпрограмм. Проверка начинается с обращения к стандартной подпрограмме НАЧАТЬ ВВОД-ВЫВОД, которая запускает цепочку команд и подпрограмму ОЖИДАТЬ. Подпрограмма ОЖИДАТЬ служит для выдержки времени для выполнения заданных команд. Если при обработке встретятся те или иные ошибки, то управление передается по программе ОБРАБОТКА и ВЫВОД СООБЩЕНИЯ ОБ ОШИБКАХ, которая печатает сообщение о полученной ошибке.

После выполнения команды ВВОД-ВЫВОД информация о состоянии системы ввода-вывода представляется в виде признака результата (кода условия). По признаку результата управляющая программа процессора может судить о том, как начата (прекращена) операция ввода-вывода или в каком состоянии находится канал или конкретное устройство ввода-вывода.

Рассмотрим описание одной тест-секции F6100, выполняющей функциональную проверку накопителя на сменных магнитных дисках. Эта проверка выполняется семью примерами.

В примере 1 проверяется, свободен ли канал. Если при проверке появляется ошибка, то печатается стандартное сообщение следующего типа:

```
ОШ. XX ОШИБКА BS10, СЕКЦ S, ПР. R,  
УСТР CVV АДР, АААААА
```

где XX — код ошибки в примере; S10 — команда; S — номер секции; R — номер примера; CVV — адрес устройства ввода-вывода; AA — A — адрес ошибки,

Таблица 6-2

## Байт состояния НМД

Разряд байта	Наименование разряда	Условия формирования
0	Внимание	Не используется
1	Модификатор состояния	Выполнялась команда поиска и условия поиска удовлетворены. Вместе с признаком «Занято» означает, что занято устройство управления
2	УВУ кончило	Если ранее был выдан байт состояния с признаком «Занято», а затем УВУ освободилось. Вместе с признаком «Сбой в устройстве» означает, что обнаружен сбой после выдачи байта состояния с признаком «Канал кончил»
3	Занято	Адресованный каналом накопитель занят; этот признак устанавливается, когда начинается новая цепочка команд, а механизм накопителя еще находится в движении, которое задано предшествующей командой установки, или когда накопитель зарезервирован другим каналом. Вместе с признаком «Модификатор состояния» означает, что УВУ занято. Устанавливается также, если новая цепочка команд задается каналом в то время, когда УВУ выполняет стирание дорожки после окончания выполнения команд записи формата или когда УВУ занято выполнением операции, заданной командой другого канала
4	Канал кончил	Устанавливается при окончании работы УВУ с каналом после того, как вся необходимая для выполнения операции информация получена или выдана в канал
5	ВУ кончило	Адресованный каналом накопитель не занят. После команд установки он выдается в канал, чтобы сообщить, что выполнение операции накопителем завершено. При выполнении других команд выдается устройством управления одновременно с признаком «Канал кончил»

Разряд байта	Наименование разряда	Условия формирования
6	Сбой в устройстве	В УВУ или накопителе обнаруживается ошибка или сбой при выполнении команды канала. Устанавливается одновременно с признаком «Канал кончил» и «ВУ кончило», если он выдается в канал в конечном байте состояния
7	Особый случай	Управляющая программа обнаруживает, что длина поля данных, указанная в поле счета, равна нулю. Этот признак сообщает также, что обнаружен конец массива информации при выполнении команд: СЧИТАТЬ ПЗП, СЧИТАТЬ КЛЮЧ И ДАННЫЕ, СЧИТАТЬ СЧЕТ, КЛЮЧ И ДАННЫЕ, СЧИТАТЬ ДАННЫЕ, ЗАПИСАТЬ КЛЮЧ И ДАННЫЕ или ПОИСК ПО КЛЮЧУ И ДАННЫМ

Таблица 6-3

**Байты уточненного состояния НМД**

Разряд байта	Наименование разряда	Условия формирования
<i>Нулевой байт</i>		
0	Команда отвергнута	УВУ получило недействительный код операции недействительную последовательность команд недействительный адрес установки Если нарушена маска защиты записи, этот признак устанавливается вместе с признаком «Защита накопителя»
1	Требуется вмешательство	Адресованный каналом накопитель: физически не подключен к системе физически подключен, но не может использоваться (не включен двигатель и т. д.)

Разряд байта	Наименование разряда	Условия формирования
2	Ошибка на шинах канала	При подаче информации из канала в УВУ обнаружена ошибка при контроле на нечетность в данных или коде команды
3	Сбой оборудования	В УВУ или в накопителе обнаружены необычные условия. Состояние, указанное этим разрядом, расшифровывается во втором байте БС
4	Ошибка в данных	Обнаружена ошибка в данных, считанных с дорожки накопителя
5	Переполнение	Сигнал ИНФ-К не был получен УВУ в заданное время после выдачи в канал сигнала ИНФ-А или канал поддерживает цепочку, но очередная команда поступила в УВУ слишком поздно, чтобы ее можно было выполнить
6	Контроль дорожки	Устанавливается в следующих случаях: если при обработке переходящей записи командами типа СЧИТАТЬ, ЗАПИСАТЬ или ПОИСК в накопителе выбрана дефектная дорожка если однопорожечная команда (кроме команд ПОИСК СОБСТВЕННОГО АДРЕСА, СЧИТАТЬ СОБСТВЕННЫЙ АДРЕС и СЧИТАТЬ) выполняется на дефектной дорожке; если при выполнении многодорожечной команды или команды, обрабатывающей переходящую запись, УВУ делает попытку переключиться с запасной дорожки на основную
7	Ошибка установки	Накопитель не может завершить установку адреса дорожки по следующим причинам: заданный адрес установки выходит за пределы адресов, действительных для данного типа накопителя. При этом УВУ устанавливает и признак «Команда отвергнута»; канал передал в УВУ меньше шести байтов адреса установки. При этом УВУ устанавливает и признак «Команда отвергнута».

Разряд байта	Наименование разряда	Условия формирования
		<p>возник отказ оборудования, при котором механизм доступа накопителя не может правильно установиться</p> <p>УВУ устанавливает также признак «Ошибка установки» в том случае, когда при выполнении многодорожечных операций или операций переходящей записи адрес вновь выбранной дорожки не совпадает с ее адресом, подсчитанным в устройстве управления</p>
<i>Первый байт</i>		
0	Ошибка в поле счета	Обнаружена ошибка в информации на участке счета. Одновременно с этим признаком УВУ устанавливает признак «Ошибка в данных»
1	Переполнение дорожки	Запись информации не была закончена к моменту обнаружения индекса при выполнении команд: ЗАПИСАТЬ ЗАПИСАТЬ СЧЕТ, КЛЮЧ И ДАННЫЕ, ЗАПИСАТЬ ДАННЫЕ, ЗАПИСАТЬ КЛЮЧ И ДАННЫЕ и УПРАВЛЯТЬ ОБЛАСТЬЮ СЧЕТА
2	Концы цилиндра	В накопителе обнаружен конец цилиндра, когда выполнение цепочки команд не закончено
3	Неправильная последовательность команд	Неправильная последовательность фиксируется в том случае, когда канал в одной цепочке выдает две команды УСТАНОВИТЬ МАСКУ и при попытке подключить головку без предварительной установки адреса дорожки Одновременно устанавливается признак «Команда отвергнута»
4	Запись не найдена	При выполнении цепочки команд УВУ два раза обнаружило прохождение индекса в накопителе, не выполнив команд считывания или записи в поле данных, чтения собственного адреса, чтения поля данных записи. Кроме того, этот признак устанавливается вместе с признаком «Пропуск адресного маркера», если на дорожке нет информации, т. е. не обнаруживается ни адресный маркер, ни запись собственного адреса, ни запись $R_0$

Разряд байта	Наименование разряда	Условия формирования
5	Защита записи	<p>Команды типа УСТАНОВКА, ЗАПИСАТЬ или многодорожечная команда чтения или поиска поступает с нарушением маски защиты записи. Если маска записи запрещает команду типа ЗАПИСАТЬ; заданную каналом, то кроме признака «Защита записи» устанавливается признак «Команда отвергнута»</p>
6	Пропуск адресного маркера	<p>При выполнении любой команды или цепочки команд, которые выполняют последовательные считывания на дорожке, оказывается, что две следующие друг за другом записи имеют одинаковые значения нулевого разряда флагового байта. Кроме того, этот признак устанавливается вместе с признаком «Запись не найдена», если на дорожке нет записанной информации</p>
7	Неразрешенный переход	<p>Используется как дополнительный признак переходящей записи и устанавливается в следующих случаях:</p> <ul style="list-style-type: none"> <li>когда УВУ обнаруживает, что дорожка, которая выбрана после завершения обработки предыдущего сегмента записи, отмечена как дефектная. При этом одновременно устанавливается признак «Контроль дорожки»</li> <li>когда УВУ делает попытку выполнить переход с запасной дорожки. При этом устанавливается также признак «Контроль дорожки»</li> <li>когда собственный адрес дорожки, выбранной в накопителе, после обработки предыдущего сегмента записи не совпадает с ее адресом</li> </ul> <p>В этом случае одновременно устанавливается признак «Ошибка установки»</p> <ul style="list-style-type: none"> <li>когда при выполнении перехода на новую дорожку УВУ обнаруживает сигнал «Конец цилиндра». В этом случае дополнительно устанавливается признак «Конец цилиндра»</li> </ul>

Разряд байта	Наименование разряда	Условия формирования
		когда нарушена маска защиты записи накопителя. При этом дополнительно устанавливается признак «Защита записи»

*Второй байт*

0	Сбой в накопителе	В накопителе обнаружен сбой (выбрано более одной головки, накопитель пытается писать и читать одновременно)
1	Ошибка при преобразовании	Не используется
2		УВУ обнаружило ошибку при преобразовании данных из параллельной формы в последовательную
3	Ошибка АЛУ	Не используется
4		Управляющая программа УВУ обнаружила нелогичное условие, которое свидетельствует о сбое оборудования
5	Невыбранное состояние	Есть сигнал о состоянии накопителя на шинах сопряжения с накопителями, когда ни один из них логически не присоединен к устройству управления
6		Не используется
7		Не используется

*Третий байт*

0	Готов	Накопитель работоспособен и не занят выполнением команды типа УСТАНОВКА
1	Накопитель подключен	Накопитель работоспособен и логически подключен к устройству управления
2	Неисправно	В накопителе имеются неисправности
3	Запись без сбоев	Для накопителей типа ЕС-5050 не используется; для накопителей ЕС-5051, ЕС-5033, ЕС-5035 означает, что запись прошла без сбоев
4	Накопитель подключен	Устанавливается одновременно с первым разрядом третьего байта уточненного состояния

Разряд байта	Наименование разряда	Условия формирования
5	Конец цилиндра	При выборе новой дорожки в накопителе номер дорожки превышает максимально возможное значение для данного типа накопителя
6	Ошибка доступа	Не используется
7		Выполнение команды типа УСТАНОВКА в накопителе не закончено, несмотря на то что максимальное время выполнения установки истекло

*Четвертый байт*

Не используется

*Пятый байт*

0000110	Выдается в виде одной из следующих комбинаций разрядов: Выполнялась команда типа СЧИТАТЬ
00100101	Выполнялась команда типа ЗАПИСАТЬ
00100101	Выполнялась команда ПОИСК ПО РАВЕНСТВУ КЛЮЧА И ДАННЫХ, и до момента ошибки логическое условие равенства удовлетворялось
01000101	Выполнялась команда ПОИСК ПО ПРЕОБЛАДАНИЮ КЛЮЧА И ДАННЫХ, и до момента ошибки удовлетворялось логическое условие поиска
01100101	Выполнялась команда ПОИСК ПО РАВЕНСТВУ ИЛИ ПРЕОБЛАДАНИЮ КЛЮЧА И ДАННЫХ, и до момента ошибки удовлетворялось логическое условие равенства
01010101	Выполнялась одна из трех команд поиска по ключу и данным, и до момента ошибки логическое условие заведомо не удовлетворялось
01110101	Выполнялась команда ПОИСК ПО ПРЕОБЛАДАНИЮ КЛЮЧА И ДАННЫХ или ПОИСК ПО РАВЕНСТВУ ИЛИ ПРЕОБЛАДАНИЮ КЛЮЧА И ДАННЫХ, и до момента ошибки выполнено логическое условие преобладания, т. е. уже выполнено логическое условие поиска



В примере 1 возможны следующие коды ошибок:  
0101 — получен код условия 1. В канале осталось необработано прерывание;

0102 — получен код условия 2. Канал работает;

0103 — получен код условия 3. Канал не работает.

В примере 2 проверяется готовность заданного устройства ввода-вывода (УВВ) и возможность доступа к нему. Стандартное сообщение об ошибке при выполнении второго примера может содержать следующие коды ошибок:

0201 — код условия 1, но нет бита ошибки в устройстве;

0202 — код условия 2. Канал работает;

0203 — код условия 3: УВВ не работает. Для дополнительной более глубокой проверки следует зациклить пример;

0204 — код условия 1. Установлены биты ошибки в устройстве в байте состояния и сбоя в накопителе в байте уточненного состояния;

0205 — код условия 1. Установлен бит ошибки в устройстве

в байте состояния, но не установлен бит подключения накопителя;

0206 — код условия 1. Установлен бит ошибки в устройстве

в байте состояния; в байте уточненного состояния есть бит подключения накопителя, но нет бита готовности;

0207 — код условия 1. В байте состояния канала УВВ установлен бит ошибки в устройстве. В байте уточненного состояния бит подключения накопителя и готовности установлены, бит невыбранного состояния не установлен;

0208 — код ошибки, указывающий на то, что к установлению бита ошибки в устройстве в байте состояния канала ввода-вывода привели какие-то условия, отличные от указанных в коде 0207;

0209 — код условия 1. Байт состояния канала ввода-вывода нулевой и бит ошибки в устройстве нет, УУ не занято;

0210 — в байте состояния канала УВВ установлен бит ошибки в устройстве и в байте уточненного состояния установлен бит конца цилиндра;

0211 — код условия 1. Устройство управления занято.

Пример 3 служит для проверки команды НЕТ ОПЕРАЦИИ. При этом возможны следующие коды ошибок:

0301 — ошибка в устройстве;

0302 — нет сигналов, что канал и устройство УВВ работу закончили;

0303 — неверно выработан код условия — 0,2 или 3 вместо 1.

Пример 4 служит для проверки команды УТОЧНИТЬ СОСТОЯНИЕ.

Возможные коды ошибок:

0401 — ошибка в устройстве;

0403 — нет сигналов, что канал и устройство УВВ работу закончили;

0404 — полученные байты уточненного состояния отличны от эталонных, но ошибки в устройстве нет и байт состояния канала УВВ нулевой.

Пример 5 служит для проверки отсутствия «зависания» информации при выполнении команд ЧТЕНИЕ СА (собственного адреса), УСТАНОВКА ГОЛОВКИ; УСТАНОВКА, УСТАНОВКА В НАЧАЛЬНОЕ ПОЛОЖЕНИЕ.

Возможные коды ошибок:

0501 — «зависание» после команды ЧТЕНИЕ СА;

0502 — «зависание» после команды УСТАНОВКА ГОЛОВКИ;

- 0503 — «зависание» после команды УСТАНОВКА;
- 0504 — «зависание» после команды ОРИЕНТАЦИЯ;
- 0505 — потеря готовности после команды УСТАНОВКА ГОЛОВКИ.

В примере 6 чтение собственных адресов на всех десяти дорожках от 00 до 09 нулевого (00) цилиндра.

Сообщение об ошибках в этом случае имеет вид:

```

ОШ      06XX   CVV   АААААА
НОМЕР ГОЛОВКИ 00 01 02 03 04 05 06 07 08 09
РЕЗ. ЧТЕНИЯ СА 00 01 02 0X НН ТХ ОП СН ОД ДР
ГЛ 04 СЧИТ. СА С5 С5 С5 С5 С5
С SW Y ... YY          БУС В-В В-В В-В В-В
ГЛ 05 ...
.....
ГЛ 09 ...
С SW ...

```

Расшифруем условные обозначения, используемые в стандартном сообщении об ошибках.

Кроме обычного сообщения с кодом ошибки здесь печатается карта головок, в которой перечислены номера головок сменного пакета магнитных дисков от 00 до 09 и результаты чтения собственных адресов. Собственный адрес равен при отсутствии сбоя устанавливаемому, т. е. адрес номера головки (ОХ) и результат чтения СА должны совпадать. Если же они не совпали (в примере, начиная с четвертой головки), то для всех несовпавших отмечается вид ошибки:

- НН — на дорожке запись не найдена;
- ПР — не было передачи данных;
- ОП — ошибка при преобразованиях;
- СН — сбой в накопителе;
- ОД — ошибка в данных;
- ДР — другие ошибки.

После карты для каждой из головок, в которой произошел сбой, печатается считанный СА, слово состояния УВВ (СSW YYY ... Y) и байт уточненного состояния устройства (БУС В-В В-В В-В В-В).

До начала проверки поле чтения СА предварительно заполняется символами С5, и если после окончания примера содержимое поля не изменилось (как это показано в данном случае), то, значит, передачи данных не было.

Приведем коды ошибок при выполнении примера 6.

- 0601 — на всех дорожках запись не найдена;
- 0602 — на всех дорожках ошибка в данных;
- 0603 — на всех дорожках ошибка при преобразовании;
- 0604 — на всех дорожках сбой в накопителе;
- 0605 — ошибка в данных и на всех дорожках запись не найдена;
- 0607 — сбой в накопителе на девяти дорожках;
- 0608 — несовпадение номеров головок;
- 0609 — ошибки на нескольких дорожках;
- 0610 — нет записи на одной из дорожек;

В примере 7 проверяется команда УСТАНОВКА В НАЧАЛЬНОЕ ПОЛОЖЕНИЕ.

Возможные коды ошибок:  
0701 — ошибка в устройстве при выполнении установки;  
0702 — не установлен бит ЗАНЯТО в CSW;  
0703 — «зависание» при отсутствии ошибки в устройстве;  
0704 — ошибка доступа;  
0706 — после выполнения команды ОРИЕНТАЦИЯ считанный собственный адрес отличен от 0000 при отсутствии ошибки в устройстве;  
0707 — ошибка в устройстве при чтении СА.

## ГЛАВА СЕДЬМАЯ

### ДИАГНОСТИКА НЕИСПРАВНОСТЕЙ ЭВМ

#### 7-1. ДИАГНОСТИКА НЕИСПРАВНОСТЕЙ ЭЛЕМЕНТОВ ЭВМ

Диагностические тесты предназначены для выявления и локализации неисправностей, возникающих в отдельных блоках и элементах ЭВМ.

В гл. 6 были рассмотрены контролирующие тесты, позволяющие проверить, как отдельные устройства ЭВМ (процессор, ОЗУ, ВЗУ и УВВ) выполняют свои функции, т. е. выполняют арифметические и логические операции, осуществляют переход по командам передачи управления, осуществляют обработку и хранение программы и числовых данных и т. д. Задача диагностических тестов сложнее. Получив указание от оператора или непосредственно от контролирующего теста, диагностический тест должен локализовать и выявить конкретную неисправность элемента. Разберем на примере работы простой комбинационной схемы, состоящей из одного логического элемента И, как усложняется тест при переходе от функции обнаружения неисправности, т. е. контроля, к ее локализации, т. е. диагностике.

**Пример 7-1.** На рис. 7-1 приведены схемы логического элемента И на два входа для следующих случаев:

- 1) схема исправна (рис. 7-1, а);
- 2) первый вход оборван (рис. 7-1, б). В этом случае работа схемы практически не зависит от того, какой сигнал приходит на вход  $X_1$ , т. е. для работы схемы обрыв первого входа эквивалентен постоянной подаче на этот вход сигнала  $X_1 \equiv 1$ ;
- 3) короткое замыкание сопротивления  $R$ . В этом случае, какие бы сигналы ни приходили на входы  $X_1$  и  $X_2$ , на выходе  $Z$  постоянно будет держаться напряжение источника питания, т. е.  $Z \equiv 1$  при любых  $X_1$  и  $X_2$  (рис. 7-1, в).

На рис. 7-2 приведены таблицы истинности работы схемы для трех указанных случаев соответственно. На рис. 7-2, б и в значения сигналов  $X_1$  и  $Z$ , отличные от истинных, обведены кружками.

Сравнение вторых строк всех трех таблиц истинности показывает, что наличие неисправности обнаруживает простой тест  $X_1 = 0$ ,  $X_2 = 1$ . Однако тип неисправности этот тест не указывает, так как при этом тесте  $Z = 1$  при неисправности  $X_1 \equiv 1$  и при неисправности  $Z \equiv 1$ . Для выявления конкретного места неисправности необходимо на входы  $X_1$  и  $X_2$  подать тест  $X_1 = 0$  и  $X_2 = 0$ . Тогда, если  $Z = 0$ ,

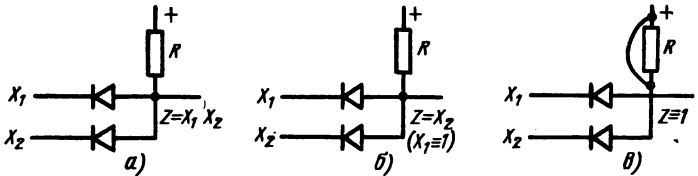


Рис. 7-1. Схема элемента И при отсутствии и наличии неисправностей.

неисправен первый вход схемы, т. е.  $X_1 \equiv 1$ ; если же  $Z = 1$ , то неисправен выход схемы, т. е.  $Z \equiv 1$ . Таким образом, для проверки работоспособности схемы достаточно одного теста 0,1. Для выявления же отказавшего элемента нужно последовательно использовать два теста, вначале 0,1, а после этого 0,0.

Входы		Выход
$X_1$	$X_2$	$Z$
0	0	0
0	1	0
1	0	0
1	1	1

а)

Входы		Выход
$X_1$	$X_2$	$Z$
1	0	0
1	1	1
1	0	0
1	1	1

б)

Входы		Выход
$X_1$	$X_2$	$Z$
0	0	1
0	1	1
1	0	1
1	1	1

в)

Рис. 7-2. Таблицы истинности элемента И.

Приведенный пример касался проверки неисправностей в одном логическом элементе при условии, что входы и выход этого элемента контролируются. В ЭВМ содержатся сотни тысяч элементов, соединенных в схемы, у которых нет возможности контролировать все входы и выходы. ЭВМ третьего поколения строятся на базе интегральных схем, каждая из которых содержит десятки и сотни комбинационных элементов. С конструктивной точки зрения де-

лать в этих схемах много контрольных выводов невыгодно — это невыполнимо технологически и усложнило бы общую монтажную схему блоков ЭВМ. В данной главе будет рассматриваться путь поиска неисправностей, который позволяет по заданной комбинации входных сигналов и измеренным выходным определить отдельный дискретный неисправный элемент внутри схемы. Заметим, что этот путь применим для поиска не только отдельного отказавшего элемента сложной схемы, но и отказавшей интегральной схемы, рассматриваемой как отдельный функциональный элемент в сложном блоке, состоящем из многих схем.

В настоящее время различают тесты, использующиеся для диагностики неисправностей в комбинационных и последовательностных схемах. Для комбинационных схем методы диагностики неисправностей более просты, чем для последовательностных. В связи с этим описание методов построения диагностических тестов начнем с тестов для комбинационных схем.

### 7-2. МЕТОД АКТИВИЗАЦИИ ОДНОМЕРНОГО ПУТИ

Рассмотрим один из методов построения диагностических тестов для комбинационных схем ЭВМ — метод активизации одномерного пути.

Идею метода активизации одномерного пути легко понять из следующего примера.

**Пример 7-2.** Пусть задана простая комбинационная схема, состоящая из двух последовательно соединенных элементов ИЛИ и И на три входа каждый (рис. 7-3). Необходимо подобрать такую последовательность входных сигналов ( $X_1, X_2, X_3, X_4, X_5$ ), которая в сочетании с выходным сигналом  $Z$  позволила бы обнаружить любую неисправность элемента ИЛИ, выход которого непосредственно не измеряется. Неисправности элемента ИЛИ могут иметь следующий вид: обрыв цепи одного из входов (неисправность типа  $X_i \equiv 0$ , где  $i$  — номер входа, в цепи которого имеется обрыв); обрыв в цепи нагрузки (неисправность типа  $Y \equiv 0$ ) и короткое замыкание (неисправность типа  $Y \equiv 1$ ).

Легко заметить, что при работе одной схемы ИЛИ обрыв в цепи любого входа обнаруживает один из трех тестов: 1)  $X_1 = 1, X_2 = 0, X_3 = 0$ ; 2)  $X_1 = 0, X_2 = 1, X_3 = 0$ ; 3)  $X_1 = 0, X_2 = 0, X_3 = 1$ . А для проверки выхода нужно вслед за первыми тестами подать тест 000, который обнаружит неисправность типа  $Y \equiv 1$ . Неис-

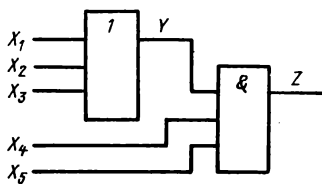


Рис. 7-3. Комбинационная схема, состоящая из элементов ИЛИ и И.

правность типа  $Y \equiv 0$  проверяется теми же тестами, что и неисправность входов. Например, если после подачи теста 100 выходной сигнал  $Y = 0$ , то подается тест 010. Если на выходе  $Y = 0$ , то в схеме неисправность типа  $Y \equiv 0$ ; если же  $Y = 1$ , то неисправность  $X_1 \equiv 0$  (при проведении анализа предполагалось, что в схеме только одна неисправность, если же их одновременно две или три, то их можно выявлять последовательно, используя все четыре теста).

Однако описанные выше результаты можно использовать, если  $Y$  измеряется. В нашей же схеме измеряется  $Z$ , а  $Y$  измерить нельзя. Идея метода активизации одномерного пути состоит в том, чтобы подать на доступные входы элемента И  $X_4, X_5$  такие сигналы, которые позволили бы по выходу  $Z$  судить о неисправностях в элементе ИЛИ. Если задать  $X_4 = 1, X_5 = 1$ , то  $Z = 1$ , если  $Y = 1$  и  $Z = 0$ , то  $Y = 0$ .

Таким образом, нам удалось создать для внешнего элемента такие условия, что контролируемый выход  $Z$  позволяет однозначно (если перед этим проверен элемент И) судить о неисправностях внутреннего элемента, выход которого не контролируется.

Метод активизации одномерного пути применим к комбинационной схеме, состоящей из любого числа последовательно соединенных элементов. Для сложных схем выделяют прямую и обратную фазы движения. Путь в прямой фазе выбирается следующим образом. Пусть имеется  $n$  последовательно соединенных логических элементов. Предположим, для простоты, что неисправность имеется в первом элементе. Тогда сигналы на входы всех элементов от второго до  $n$ -го должны быть подобраны таким образом, чтобы выход каждого  $i$ -го элемента ( $i = 2, 3, 4, \dots, n$ ) определялся только тем его входом, сигнал на который поступает с  $(i-1)$ -го элемента. Весь путь прохождения сигнала организуется так, что контролируемый выход  $n$ -го элемента однозначно определяет неконтролируемый выход первого элемента. Тогда, задав последовательность тестовых сигналов для проверки первого элемента, можно по выходу  $n$ -го найти в первом неисправность. В обратной фазе устанавливается значение сигналов на всех контролируемых входах схемы, позволяющих провести активизацию одномерного пути. Рассмотрим пример построения прямой и обратной фазы движения для комбинационной схемы, состоящей из восьми элементов ИЛИ и И.

**Пример 7-3.** Пусть задана схема, состоящая из пяти ячеек ИЛИ и четырех ячеек И, соединенных, как это показано на рис. 7-4. Каждая из ячеек имеет два входа. И пусть в схеме имеется следующая неисправность: выход ячейки  $\mathcal{Y}2$   $Y_2 \equiv 1$ . Необходимо выбрать сигналы  $X_1, X_2, \dots, X_8$  так, чтобы по контролируемым выходам схемы  $Z_1$  и  $Z_2$  можно было обнаружить эту неисправность. Для активизации пути организуем фазу движения вперед. Очевидно, что движение возможно по одному из трех путей: с выхода  $\mathcal{Y}2$  на вход  $\mathcal{Y}4$  и далее на  $\mathcal{Y}7$  и выход  $Z_1$ ; с выхода  $\mathcal{Y}2$  на вход  $\mathcal{Y}5$  и далее через  $\mathcal{Y}6$  на  $\mathcal{Y}7$  и выход  $Z_1$ ;

и, наконец, с выхода  $У2$  через  $У5$ ,  $У6$  и  $У8$  на выход  $Z_2$ . Так как в задаче нет дополнительных условий, которые позволили бы выбрать наилучший в каком-то смысле путь, то выберем для примера путь через элементы  $У2$ ,  $У5$ ,  $У6$ ,  $У8$  и на выход  $Z_2$ .

Для активизации пути через элемент  $У5$ , а он представляет собой схему И, необходимо, чтобы на второй его вход подавался сигнал, равный единице ( $X_5 = 1$ ). Тогда выход  $У5$  будет соответствовать выходу

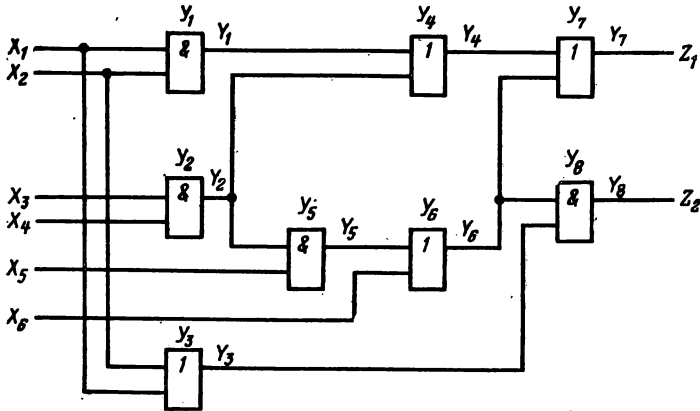


Рис. 7-4. Пример комбинационной схемы.

исследуемого элемента  $У2$ , т. е.  $Y_5 = Y_2$ . Элемент  $У6$  представляет собой схему ИЛИ. Для активизации пути через этот элемент необходимо задать  $X_6 = 0$ . Тогда  $Y_6 = Y_5 = Y_2$ . И, наконец, для активизации пути через  $У8$  (схему И) необходимо, чтобы ее второй вход  $Y_3 = 1$ . В этом случае

$$Z_2 = Y_6 = Y_5 = Y_2.$$

Таким образом, задав на входы  $X_3$  и  $X_4$  тест для проверки элемента  $У2$  по выходу  $Z_2$ , может найти неисправность в нем, если  $X_5$ ,  $X_6$  и  $Y_3$  будут принимать указанные выше значения.

На этом фаза движения вперед заканчивается. Теперь остается вернуться к началу схемы (обратная фаза движения) и получить первоначальные значения сигналов на всех входах схемы. Так как значения  $X_5$  и  $X_6$  были определены в прямой фазе движения, то в обратной фазе нужно найти  $X_1$ ,  $X_2$ ,  $X_3$  и  $X_4$ . При этом  $X_1$  и  $X_2$  легко находятся из условия  $Y_3 = 1$ . Так как элемент  $У3$  представляет собой схему ИЛИ, то для того, чтобы  $Y_3 = 1$ , можно задать входные сигналы либо  $X_1 = 1$ ,  $X_2 = 0$ , либо  $X_1 = 0$ ,  $X_2 = 1$ . Выберем для определенности первый вариант. Далее выбираем сигналы  $X_3$  и  $X_4$ . В примере 7-1 было показано, что неисправность типа  $Y_2 = 1$  может возникнуть либо в результате обрыва по цепи одного из входов ( $X_3 \equiv 1$  или  $X_4 \equiv 1$ ), либо непосредственно из-за неисправности выходной цепи ( $Y_2 \equiv 1$ ). Для локализации места неисправности нужно последовательно подать два теста: 0, 1 или 1, 0, а затем 0, 0. Таким образом, общий тест

проверки элемента  $У2$  в схеме рис. 7-4 будет иметь вид: 1, 0, 0, 1, 1, 0 для проверки первого входа и выхода или 1,0,1,0,1,0 для проверки второго входа и выхода.

Если после подачи таких тестов на входы  $X_1, X_2, X_3, X_4, X_5, X_6$  соответственно  $Z_2 = 0$ , то схема  $У2$  исправна, если же  $Z_2 = 1$ , то в схеме есть неисправность и нужно провести дополнительное исследование и подать тест 1, 0, 0, 0, 1, 0.

Если после подачи второго теста  $Z_2 = 0$ , то неисправность во входной цепи; если же  $Z_2 = 1$ , то неисправность в выходной цепи элемента.

Метод активизации одномерного пути достаточно прост, но может быть применен не для всех схем. Это связано с тем, что в этом методе используется не вся информация, которую можно получить от схемы. В самом деле, мы активизируем лишь один путь и следим только за ним, не обращая внимания на то, что делается в других ветвях схемы, хотя наши сигналы поступают и на них. Вторым недостатком метода является сложность его алгоритмизации.

От указанных недостатков в значительной мере свободен  $d$  — алгоритм Рота. Подробное изложение этого метода требует весьма громоздких объяснений и здесь опускается. Не излагая сам метод, очень кратко остановимся на его идее.

Идея метода состоит в одновременной активизации всех возможных путей от места неисправности ко всем выходам схемы. Для этого вначале выбирается тест для некоторой заданной неисправности и генерируются все возможные пути от места неисправности ко всем выходам одновременно. На каждом шаге проверяются ограничения, задаваемые алгоритмом в определенной форме.

Этот шаг алгоритма является общим для всех активизированных входов схемы метода активизации одномерного пути. После прохода к выходу схемы организуется движение в обратную сторону аналогично обратной фазе одномерного метода.

Остановимся очень коротко на методах построения тестов для последовательных схем. Эта задача оказывается более сложной, чем для комбинационных схем. Дело в том, что последовательные схемы включают элементы памяти, и поэтому проверки нужно строить таким образом, чтобы они охватывали все возможные состояния схемы, а это удваивает число проверок для каждого входящего в схему элемента памяти (так как, например, каждый триггер имеет два устойчивых состояния). Другая сложность связана с неоднозначностью первоначального состояния



схемы. В самом деле, если в схему включено несколько триггеров, то до начала проверки схему нужно установить в некоторое начальное заранее заданное состояние. Это может быть выполнено, если в схеме имеются специальные цепи сброса всех запоминающих элементов в начальное состояние. Но при этом возникает проблема предварительной проверки цепей сброса. Перечисленные и другие трудности привели к тому, что в настоящее время практически не существует методов алгоритмического построения диагностических тестов для последовательностных схем, подобных *d*-алгоритму Рота для комбинационных схем.

Основными методами, которыми пользуются сейчас для последовательностных схем, являются эвристические методы. Рассмотрим идею такого метода, называющегося *лучший следующий* или *возврат к хорошему*.

Для того чтобы построить тест, подающийся на все входы проверяющейся схемы, на некотором *i*-м такте проверки строят ряд тестов, отличающихся от теста, полученного на предыдущем такте, на 1 бит. Для каждого из полученных таким образом тестов выполняется полный просчет схемы и оценка полученного результата по некоторому заданному критерию качества работы (например, времени выполнения или числу шагов от входов до выходов схемы или количеству неисправностей, определяемого этим тестом и т. д.). Все тесты, критерий качества работы для которых оказывается ниже некоторого *порогового* значения, отбрасываются. Из оставшихся выбирается один, наилучший с точки зрения заданного критерия, а остальные запоминаются. Далее для наилучшего из выбранных тестов выполняется следующий такт. Если из построенных на новом такте тестов ни один не удовлетворяет заданному критерию, то нужно вернуться к одному из запомненных на предыдущем шаге тестов и повторить все сначала. Очевидно, что работа этого метода во многом зависит от успешного выбора критерия, порогового значения критерия, первоначального теста, т. е. в достаточной мере зависит от субъективных причин. Легко заметить также, что при большом количестве входов схемы этот метод становится чрезвычайно громоздким. К сожалению, как этот метод, так и другие методы, использующиеся при построении тестов для последовательностных схем, требуют почти полного перебора всех возможных вариантов на каждом такте вычислений и не гарантируют, что после *n* тактов мы не окажемся в положении, когда не удастся найти приемлемого теста для  $n + 1$

такта и для построения полного диагностического теста для данной схемы придется изменить первоначальный тест или перейти к другому методу и начать все сначала.

### 7-3. ТАБЛИЧНЫЕ МЕТОДЫ СИНТЕЗА МИНИМАЛЬНОГО НАБОРА ТЕСТОВ

Рассмотренные методы позволяют синтезировать тесты, обнаруживающие один заданный вид неисправности в схеме. Однако на практике обычно приходится решать задачу синтеза тестов, обнаруживающих все или подавляющее большинство неисправностей в целом функциональном устройстве. Используя описанные выше методы, теоретически можно для такого устройства, как процессор, получить количество тестов более чем  $10^6$ . Естественно, что запомнить и тем более эффективно работать с таким набором тестов нельзя. Реально поступают проще. Во-первых, каждый тест позволяет обычно найти не одну, а несколько неисправностей. Во-вторых, в современных ЭВМ все шире используются средние интегральные схемы и большие интегральные схемы, включающие десятки и сотни элементов. Для таких схем неисправность ищется не с точностью до элемента, а с точностью до схемы. И, в-третьих, имеется целый ряд методов, например ТЕСТ-ДЕТЕКТ Рота, метод эквивалентной нормальной формы и табличные методы, позволяющие резко сократить набор тестов, необходимый для обнаружения неисправностей в устройствах ЭВМ. Остановимся подробнее на наиболее простых (с точки зрения описания, но не машинной реализации) методах отыскания минимального набора тестов, основанных на преобразовании таблиц неисправностей.

Таблица неисправностей представляет собой двумерный массив, где неисправности расположены по строкам, а тесты — по столбцам. Если имеется таблица на  $n = 6$  неисправностей и  $m = 5$  тестов (табл. 7-1), то возможность отыскания  $i$ -й неисправности  $j$ -м тестом отмечается в ней как

$$X_{ij} = 1 \quad (i = 1, 2, \dots, n; j = 1, 2, \dots, m).$$

В противном случае

$$X_{ij} = 0.$$

Строки таблицы, показывающие, какие тесты могут обнаружить данную неисправность, называются отображе-

ниями неисправностей, т. е. отображение  $i$ -й неисправности может быть записано как

$$X_i = (X_{i1}, X_{i2}, X_{i3}, \dots, X_{im}).$$

Пусть имеются две неисправности, например в  $j$ -й и  $k$ -й строках, тогда можно сказать, что они различимы, если  $X_{jp} \neq X_{kp}$  хотя бы для одного  $p$ .

Таблица 7-1

Таблица неисправностей

Неисправность	Тесты				
	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>
X <sub>1</sub>	1	1	0	1	0
X <sub>2</sub>	1	1	1	0	1
X <sub>3</sub>	0	1	1	1	1
X <sub>4</sub>	0	1	0	0	1
X <sub>5</sub>	0	0	0	1	1
X <sub>6</sub>	1	0	1	0	0

При составлении таблиц используются как методы синтеза тестовых воздействий для определения заданных неисправностей методом активизации одномерного пути и  $d$ -алгоритма Рота, так и данные статистики о количестве и видах неисправностей, полученные в процессе наладки узлов и устройств ЭВМ, а также результаты физического моделирования работы машины при внесении в ее устройство различных неисправностей. При этом в выбранное устройство искусственно вносится заданная неисправность, затем через него пропускается контрольный пример и определяется реакция устройства на этот пример. В диагностическую таблицу включаются только те неисправности, которые приводят к стабильному функциональному изменению работы данного устройства. При определении перечня возможных неисправностей для диагностической таблицы необходимо иметь в виду, что имеется ряд неисправностей, которые на неисправной машине не могут быть найдены с помощью диагностических тестов. К ним относятся неисправности, приводящие к полному нарушению нормальной работы машины, а именно неисправности в цепях питания, дешифраторе, коде операций, генераторе тактовых импульсов и др.

Задача отыскания минимального набора тестов для обнаружения всех неисправностей по имеющейся полной таблице неисправностей состоит в нахождении такого мини-

мального набора столбцов таблицы, чтобы каждая строка соответствующей подтаблицы имела по крайней мере одну единицу. Наиболее простым теоретически, но практически самым громоздким методом перехода от полной таблицы к сокращенной до минимального набора столбцов является метод полного перебора. Машинная реализация этого метода позволяет успешно использовать его для сравнительно небольших схем. Для устройств с большим числом элементов этот метод получается чрезвычайно громоздким и в этих случаях используют специальные подходы, например метод *пересечения тестовых наборов* и *метод критерия различимости*.

Идея метода *пересечения наборов* тестов состоит в таком преобразовании полной таблицы неисправностей в сокращенную, при котором выполняется переход от определения одной неисправности несколькими тестами к группе неисправностей, определяемой одним или несколькими тестами.

Рассмотрим реализацию метода пересечения наборов тестов. Из таблицы неисправностей берутся вначале первая и вторая строки и над соответствующими элементами строк  $X_{11}$  и  $X_{21}$ ,  $X_{12}$  и  $X_{22}$ ,  $X_{13}$  и  $X_{23}$ , ...,  $X_{1m}$  и  $X_{2m}$  выполняется операция поразрядного логического умножения. Рассмотрим пример.

Выполним операцию поразрядного логического умножения над первыми двумя строками табл. 7-1. В результате таблица примет вид табл. 7-2.

Таблица 7-2

Первое сокращение таблицы неисправностей

Неисправность	Тесты				
	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$
$(X_1, X_2)$	1	1	0	0	0
$X_3$	0	1	1	1	1
$X_4$	0	1	0	0	1
$X_5$	0	0	0	1	1
$X_6$	1	0	1	0	0

Здесь  $(X_1, X_2)$  — новая строка таблицы, определяющая неисправности  $X_1$  и  $X_2$ . Заметим, что если бы в результате умножения строка  $(X_1, X_2)$  содержала только нулевые элементы, то согласно идее метода нужно было бы искать пересечение (т. е. выполнить операцию поразрядного логи-

ческого умножения) 1-й и 3-й строки. Если же строка  $(X_1, X_3)$  также окажется нулевой, то ищется пересечение 2-й и 3-й строк или 1-й и 4-й и так продолжается до первой ненулевой строки. Вернемся к рассмотрению нашего примера. Выполним операцию поразрядного логического умножения над 1-й и 2-й строками табл. 7-2. Новая строка будет иметь вид:

$$(X_1, X_2, X_3) \quad 0 \ 1 \ 0 \ 0 \ 0.$$

Перемножив рассмотренным способом вновь полученную строку на следующую строку таблицы, находим:

$$(X_1, X_2, X_3, X_4) \quad 0 \ 1 \ 0 \ 0 \ 0.$$

Если мы продолжим умножение дальше, то следующий этап дает

$$(X_1, X_2, X_3, X_4) \quad 0 \ 0 \ 0 \ 0 \ 0.$$

Заменив 5-ю строку на 6-ю, получим:

$$(X_1, X_2, X_3, X_4, X_5, X_6) \quad 0 \ 0 \ 0 \ 0 \ 0.$$

Две последние полученные строки в таблице пустые, т. е. нет ни одного из тестов  $T_1, T_2, \dots, T_5$ , который позволил бы обнаружить комплексы неисправностей

$$(X_1, X_2, X_3, X_4, X_5) \text{ или } (X_1, X_2, X_3, X_4, X_6).$$

Окончательно сокращенная таблица неисправностей для рассмотренного примера дана в табл. 7-3.

Таблица 7-3

Сокращенная таблица неисправностей

Неисправность	Тесты				
	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$
$(X_1, X_2, X_3, X_4)$	0	1	0	0	0
$X_5$	0	0	0	1	1
$X_6$	1	0	1	0	0

Согласно этой таблице тест  $T_2$  позволяет обнаружить неисправности вида  $X_1, X_2, X_3$  и  $X_4$ .

Тесты  $T_4$  и  $T_5$  обнаруживают неисправность  $X_5$ , а тесты  $T_1$  и  $T_3$  — неисправность  $X_6$ .

Однако данное решение является не единственным и даже не наилучшим. Если вернуться к исходной табл. 7-1

и искать пересечение строк со 2-й по 5-ю, то получим новую строку в виде

$$(X_2, X_3, X_4, X_5) \quad 0 \ 0 \ 0 \ 0 \ 1$$

А пересечение 1-й и 6-й строк дало бы

$$(X_1, X_6) \quad 1 \ 0 \ 0 \ 0 \ 0$$

Таким образом, новая (полученная при другом переборе тех же строк исходной таблицы) сокращенная таблица состоит всего из двух строк и позволяет найти с помощью теста  $T_5$  неисправности  $X_2, X_3, X_4$  и  $X_5$ , а с помощью теста  $T_1$  неисправности  $X_1$  и  $X_6$ .

Полученные результаты показывают, что метод пересечения наборов тестов достаточно прост и эффективен, но не всегда с первого раза приводит к оптимальному тестовому набору.

Рассмотрим вкратце второй метод сокращения полной таблицы неисправностей — метод *критерия различимости*. Идея метода состоит в преобразовании полной таблицы неисправностей в сокращенную на основе отбора более эффективных тестов по заданным критериям эффективности. В качестве критерия в методе используется *вес* теста  $W_j$ . Под весом теста понимают число пар неисправностей, которые этот тест различает. Вес теста  $W_j$  вычисляется по формуле

$$W_j = (n_0^j) \cdot (n_1^j), \quad (7-1)$$

где  $n_0^j$  — количество нулей;  $n_1^j$  — количество единиц в  $j$ -м столбце таблицы неисправностей.

Работа метода заключается в организации ряда последовательных итераций. По заданной полной таблице неисправностей находится тест с наибольшим весом. Этот тест делит все неисправности в таблице на два блока — с обнаруживаемыми и необнаруживаемыми неисправностями.

Далее таблица перестраивается соответствующим образом, из отбора исключается первый найденный тест, по новой таблице считаются новые веса, для теста с наибольшим полученным весом проводится перестройка таблицы и разбиение ее на новые блоки и т. д. Процесс продолжается до тех пор, пока разделение неисправностей перестает приносить дальнейшее улучшение, т. е. до тех пор, пока веса всех использованных тестов не станут равны нулю.

## 7-4. СЛОВАРИ НЕИСПРАВНОСТЕЙ

Таблицы неисправностей создаются на стадии разработки, создания и испытания ЭВМ. При эксплуатации машин применяют словари неисправностей. Наиболее простым типом словаря неисправностей является словарь точного соответствия. Пользуются этим словарем следующим образом: вначале пропускаются диагностические тесты и записываются результаты испытаний, а затем эти результаты сравниваются с входными элементами словаря. Полное совпадение позволяет точно определить обнаруженные неисправности. Для удобства пользования словарю точного соответствия обычно строятся следующим образом: в одной колонке перечисляются неисправности, в другой — обнаруживающие их тесты. В табл. 7-4 приведен пример словаря точного соответствия (табл. 7-4, б), полученного на основе таблицы неисправностей (табл. 7-4, а) и позволяющего обнаружить восемь неисправностей с помощью четырех тестов.

Таблица 7-4

Неисправности	Тесты			
	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>
X <sub>1</sub>	1	1	1	1
X <sub>2</sub>	1	1	1	0
X <sub>3</sub>	1	1	0	0
X <sub>4</sub>	1	0	0	1
X <sub>5</sub>	0	1	1	0
X <sub>6</sub>	0	1	0	0
X <sub>7</sub>	0	0	1	0
X <sub>8</sub>	0	0	0	1

а) Таблица неисправностей

Неисправности	Тесты
X <sub>1</sub>	T <sub>1</sub> , T <sub>2</sub> , T <sub>3</sub> , T <sub>4</sub>
X <sub>2</sub>	T <sub>1</sub> , T <sub>2</sub> , T <sub>3</sub>
X <sub>3</sub>	T <sub>1</sub> , T <sub>2</sub>
X <sub>4</sub>	T <sub>1</sub> , T <sub>4</sub>
X <sub>5</sub>	T <sub>2</sub> , T <sub>3</sub>
X <sub>6</sub>	T <sub>2</sub>
X <sub>7</sub>	T <sub>3</sub>
X <sub>8</sub>	T <sub>4</sub>

б) Словарь точного соответствия

Однако на практике достаточно часто оказывается, что результаты прохождения тестов не совпадают со словарем точного представления. Это вызвано тем, что в машине возникли некоторые новые, не исследованные заранее неисправности или произошло наложение ряда неисправностей, более того, два последовательных выполнения тестовой процедуры в присутствии одиночной устойчивой неисправности могут дать несовпадающие результаты. Поэтому, кроме основного словаря точного соответствия, обычно

используют вспомогательные словари. Одним из них является гнездовой словарь.

Идея гнездового словаря состоит в использовании понятия *близости* между результатами прохождения тестов и предварительно построенной таблицы неисправностей. Для этого вся таблица разбивается на равное число частей (сегментов), в каждом из которых все разряды одинаковы. Далее составляется гнездовой словарь, в котором записаны гнездовые идентификаторы и соответствующие им неисправности. Гнездовой идентификатор получается следующим образом. Пусть таблица неисправностей получена с помощью девяти тестов и одна из строк ее для неисправности  $X_i$  имеет вид 000001011.

Разобьем строку на три равные части и каждому сегменту припишем ту из цифр (0 или 1), которых в каждой части будет больше. Для неисправности  $X_i$  имеем «гнездо» с гнездовым идентификатором 001.

В табл. 7-5 приведены таблица неисправностей (а) и гнездовой словарь для нее (б).

Таблица 7-5

Неисправности	Тесты								
	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>7</sub>	T <sub>8</sub>	T <sub>9</sub>
X <sub>1</sub>	0	0	0	0	0	1	0	1	1
X <sub>2</sub>	0	0	1	1	0	0	1	1	1
X <sub>3</sub>	0	1	1	0	0	0	1	0	1
X <sub>4</sub>	0	0	1	0	0	0	1	1	1
X <sub>5</sub>	0	1	1	1	0	0	1	1	1

а) Таблица неисправностей

Гнездовой идентификатор	Неисправности
001	X <sub>1</sub> , X <sub>2</sub> , X <sub>4</sub>
101	X <sub>3</sub> , X <sub>5</sub>

б) Гнездовой словарь

Если в результате прохождения девяти тестов мы получили некоторую неисправность  $X_j$  с кодом 100010110, то, разбив код неисправности на три части и применив указанное выше правило, получим гнездовой идентификатор неисправности 001.



Согласно гнездовому словарю, приведенному в табл. 7-5, б, полученная неисправность близка к  $X_1$ ,  $X_2$  и  $X_4$ .

Гнездовые словари, так же как и другие формы вспомогательных словарей, теряют большое количество информации, так как с целью ускорения их работы производится сжатие информации (в нашем примере из каждых трех значащих разрядов оставался только один), поэтому на практике обычно используют сочетание вспомогательных словарей гнездового типа для первичного поиска группы неисправностей и словаря точного представления для уточнения.

#### **7-5. ДИАГНОСТИЧЕСКИЙ ТЕСТ МАГНИТНОЙ ЛЕНТЫ В СИСТЕМЕ ТЕСТ-МОНИТОР**

Диагностический контроль всех устройств и блоков ЭВМ ЕС-1022 выполняется аппаратурным и аппаратурно-программным путем. При аппаратурно-программном методе контроля используются микропрограммы.

Тестовый метод используется при диагностическом контроле для проверки правильности функционирования контрольных цепей и цепей обнаружения ошибок.

Рассмотрим один из таких тестов, включенных в состав ТЕСТ-монитора, — тест диагностики магнитной ленты.

Тест состоит из трех функционально независимых тест-секций, предназначенных для проверки цепей корректировки ошибок, обнаружения ошибок и правильности функционирования регистра циклического контроля.

Каждая тест-секция, выполняющая одну из указанных выше проверок, состоит из подпрограмм этой тест-секции и примеров. Кроме того, все тест-секции используют ряд общих подпрограмм.

В число общих подпрограмм входят:

1) подпрограмма выхода по ошибке; эта подпрограмма включается, если при просмотре примера была обнаружена ошибка, и готовит сообщение об ошибке для печати;

2) подпрограммы выполнения команд управления для выполнения команд ШАГ НА ГРУППУ ЗОН ВПЕРЕД, ШАГ НА ГРУППУ ЗОН НАЗАД, ЗАПИСАТЬ МАРКЕР ЗОН, ПЕРЕМОТАТЬ. После выполнения каждой из этих подпрограмм в БС (табл. 7-6) проверяемого УВВ бит ошибки в УВВ должен равняться 0. Если же он равен 1, то в соответствующей подпрограмме имеется ошибка и управление передается подпрограмме печати ошибок;

Таблица 7-6

## Байт состояния ЗУ на магнитной ленте

Разряд байта	Наименование разряда	Условия формирования
0	Внимание	Не используется
1	Модификатор	<p>Устанавливается в следующих случаях:</p> <p>в конце операции, во время которой выполнялась последовательность сигналов УВУ по окончании команды управления, если устанавливается признак «Сбой в устройстве» или «Особый случай», а признак «Канал кончил» уже принят по окончании работы с каналом А (Б), если перед этим выполнялась последовательность сигналов занятого УВУ для канала Б (А); байт состояния посылается в канал Б (А)</p>
3	Занято	<p>Без признака «Модификатор состояния» означает:</p> <p>адресуемый накопитель занят (например, выполняет команду ПЕРЕМОТАТЬ) состояние адресуемого накопителя было заполнено устройством управления от предыдущей операции</p> <p>адресуемый накопитель подсоединен к другому каналу</p> <p>Вместе с признаком «Модификатор состояния» означает:</p> <p>УВУ находится в состоянии хранения прерывания</p> <p>УВУ выполняет операцию</p> <p>УВУ подсоединено к другому каналу.</p>
4	Канал кончил	Заданная операция выполнена на уровне канала. В случае команд режима и вспомогательных команд операция начата УВУ или уже закончена им и канал может быть свободен
5	ВУ кончило	Накопитель закончил операцию или состояние накопителя изменилось с «Не готов» на «Готов», или закончено выполнение команды ПЕРЕМОТАТЬ и РАЗГРУЗИТЬ на уровне УВУ

Разряд байта	Наименование разряда	Условия формирования
6	Сбой в устройстве	Означает: установлен в единицу любой разряд нулевого БУС адресуемый накопитель начинает выполнение команд СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ, ШАГ НА ЗОНУ НАЗАД, ШАГ НА ГРУППУ ЗОН НАЗАД в то время, как лента в нем находится на маркере НЛ или он обнаружен в процессе выполнения этих команд команда ПЕРЕМОТАТЬ И РАЗГРУЗИТЬ закончена на уровне УВУ
7	Особый случай	Устанавливается в том случае, когда команды ЗАПИСАТЬ, ЗАПИСАТЬ МАРКЕР ГРУППЫ ЗОН, СТЕРЕТЬ ПРОМЕЖУТОК выполняются накопителем, в котором лента находится за маркером КЛ; маркер группы зон был считан при выполнении команд СЧИТАТЬ, СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ, ШАГ НА ЗОНУ НАЗАД, ШАГ НА ЗОНУ ВПЕРЕД

3) подпрограмма установки диагностического режима для выполнения команды ДИАГНОСТИКА; после выполнения этой команды в байте состояния проверяемого устройства бит ошибки в УВВ должен равняться 0, если же он равен 1, то произошла ошибка;

4) подпрограмма записи для выполнения команды ЗАПИСЬ; контроль за выполнением этой команды вновь выполняет бит ошибки в УВВ в байте состояния устройства; однако правильное выполнение команды ЗАПИСАТЬ характеризуется установкой этого бита в 1;

5) подпрограммы печати ошибок, вывод на печатающее устройство ЭВМ сообщения об ошибке.

Всего в тесте диагностики магнитной ленты используются двенадцать общих подпрограмм.

Остановимся далее на специальных подпрограммах и примерах, используемых в одной из тест-секций теста тест-секции F51A, предназначенной для проверки цепей обнаружения ошибок в устройстве управления накопителями на магнитной ленте.

В этой тест-секции, кроме общих подпрограмм, используются три специальные:

1) подпрограмма считывания и считывания в обратном направлении для проверки команд СЧИТЫВАНИЕ и СЧИТЫВАНИЕ В ОБРАТНОМ НАПРАВЛЕНИИ; контроль за выполнением команды осуществляется по установлению бита ошибок в УВВ в байте состояния в 1;

2) подпрограмма проверки ошибок, которая проверяет наличие 0 или 1 в битах байтов уточненного состояния: в регистре записи-считывания, в регистре продольного контроля, в регистре циклического контроля, помехи, ошибки в данных;

3) подпрограмма окончания, которая печатает сообщение для каждого проверенного УВВ об окончании всех проверок.

В тест-секции F51A для непосредственного выполнения проверок используются четыре примера.

В первом примере выполняется проверка байтов уточненного состояния контролируемого УВВ после выполнения команд ЗАПИСАТЬ, УТОЧНИТЬ СОСТОЯНИЕ, ПРОВЕРИТЬ ВВОД-ВЫВОД.

Проверка выполняется в три этапа.

На первом этапе устанавливается диагностический режим и после выполнения подпрограммы уточнения состояния сравниваются действительная информация, записанная в БУС, и ожидаемая.

На втором этапе вновь подается команда УТОЧНИТЬ СОСТОЯНИЕ и сравнивается с данными, полученными на предыдущем этапе. Если регистр уточненного состояния при подаче команды не изменил информацию, содержащуюся в нем, то команда работает правильно. В противном случае печатается сообщение об ошибке.

На третьем этапе проверяется команда ПРОВЕРИТЬ ВВОД-ВЫВОД. Контроль правильности выполнения команды осуществляется так: выполняется команда с мнемоническим кодом на языке АССЕМБЛЕР T10 и проверяется, произошло ли изменение содержания регистра уточненного состояния. Если нет, то все нормально, в противном случае печатается сообщение об ошибке.

На каждом этапе выполнения примера при наличии ошибок печатаются свои сообщения об ошибках.

На первом этапе

XXXXXXXXXXXX — ОЖИДАЕМЫЕ БУС

XXXXXXXXXXXX — ПОЛУЧЕННЫЕ БУС

ПОЛУЧЕННЫЕ И ОЖИДАЕМЫЕ БУС НЕ РАВНЫ

Здесь БУС — байты уточненного состояния.

На втором этапе

```
XXXXXXXXXXXX — ОЖИДАЕМЫЕ БУС  
XXXXXXXXXXXX — ПОЛУЧЕННЫЕ БУС  
БУС СБРАСЫВАЮТСЯ СЛЕДУЮЩЕЙ КОМАНДОЙ.  
УТОЧНИТЬ СОСТОЯНИЕ
```

На третьем этапе

```
XXXXXXXXXXXX — ОЖИДАЕМЫЕ БУС  
XXXXXXXXXXXX — ПОЛУЧЕННЫЕ БУС  
БУС СБРАСЫВАЕТСЯ КОМАНДОЙ T10
```

Во втором примере проверяется, устанавливаются ли в 1 в байтах уточненного состояния при заданной ошибке в данных соответствующие биты при ошибках в регистрах записи-считывания (РЗС), продольного контроля (ПКР) и циклического контроля (ЦКР). В программе примера проверки трех регистров выполняются последовательно.

Вначале в регистр R0 записывается номер зоны, зона пересылается в поле записи диагностического режима и выполняется команда на запись и считывание зоны. После считывания в третьем БУС бит ошибки в регистре записи-считывания должен равняться 1. Если же он равен 0, то программа печатает сообщение об ошибке.

Для проверки работы регистра продольного контроля вновь в регистр R0 записывается номер зоны, зона пересылается в поле записи диагностического режима и выполняется команда на запись и считывание зоны. Однако при записи в строке продольного контроля содержится информация с заранее заданной ошибкой. Поэтому после выполнения команды УТОЧНИТЬ СОСТОЯНИЕ третий байт регистра уточненного состояния должен содержать в бите ошибки в ПКР 1. Если там записан 0, то программа должна напечатать сообщение об ошибке.

При проверке работы ЦКР в исходные данные включена информация с ошибкой в строке циклического контроля. После выполнения этой команды в третьем байте регистра уточненного состояния в бите ошибки в ЦКР должна быть установлена 1.

Так как весь пример служит для проверки правильности реакции регистров на появление ошибок в исходной информации, то при всех трех проверках в нулевом БУС бит ошибки в данных должен равняться 1.

Третий пример служит для проверки работы регистров записи-считывания и продольного контроля.

Проверка работы регистра продольного контроля выполняется следующим образом.

В регистр P0 записывается номер зоны, а в регистр P1 — количество проверяемых зон. Проверка осуществляется на записи с диагностикой восьми зон. После установления диагностического режима выполняется запись каждой зоны в поле записи с использованием ПКР. После каждой записи осуществляется считывание зоны, уточнение ее состояния, считывание в обратном направлении с уточнением состояния. При этом в байтах уточненного состояния проверяемого устройства биты ошибки в данных, ошибки в ПКР и ошибки в ЦКР должны быть установлены в 1.

Для проверки регистра записи-считывания вновь в P0 записывается номер зоны, а в P1 — количество проверяемых зон. В этом случае оно равно 16. Далее устанавливается диагностический режим, пересылается в поле записи первая зона, выполняется ее запись, затем выполняются команды ШАГ НА ГРУППУ ЗОН НАЗАД и ШАГ НА ГРУППУ ЗОН ВПЕРЕД, далее выполняется чтение и анализ уточненного состояния. После этого проводится анализ БУС устройства и проверяется, установлены ли биты ошибки в данных, ошибки в РЭС и ошибки в ЦКР в состоянии 1. Если нет, то печатается сообщение об ошибке, если да, то вводится вторая зона и весь цикл повторяется.

Четвертый пример служит для проверки способности блока управления накопителями на магнитной ленте обнаружить помехи в межзонном промежутке.

Делается это следующим образом. Вводится диагностический режим и поле записи очищается. Далее в поле записи пересылается зона, записывается один байт информации и на расстоянии около одного сантиметра от строки продольного контроля записывается помеха. Затем выполняется считывание и анализ УС. Если в результате анализа окажется, что в байтах уточненного состояния биты ошибки в регистре записи-считывания, ошибки в регистре продольного контроля, ошибки в регистре циклического контроля, ошибки в данных и помехи в промежутках не равны 1, то печатается сообщение об ошибке.

Если же все эти биты равны 1, то управление передается подпрограмме окончания, которая отключает диагностический режим и печатает сообщение

**УВВ ХХХХ ЗАВЕРШИЛО РАБОТУ**

где УВВ — устройство ввода-вывода; ХХХ — номер УВВ.

## 7-6. ПОСТРОЕНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОЙ ДИАГНОСТИКИ ЭВМ МЕТОДОМ РАСКРУТКИ

В предыдущих параграфах главы были рассмотрены элементы построения диагностических тестов и работа одного из таких тестов, включенного в систему ТЕСТ-монитор. Было отмечено, что полная система диагностики ЭВМ единой системы строится на основе комбинированного принципа с использованием аппаратного и программного контроля.

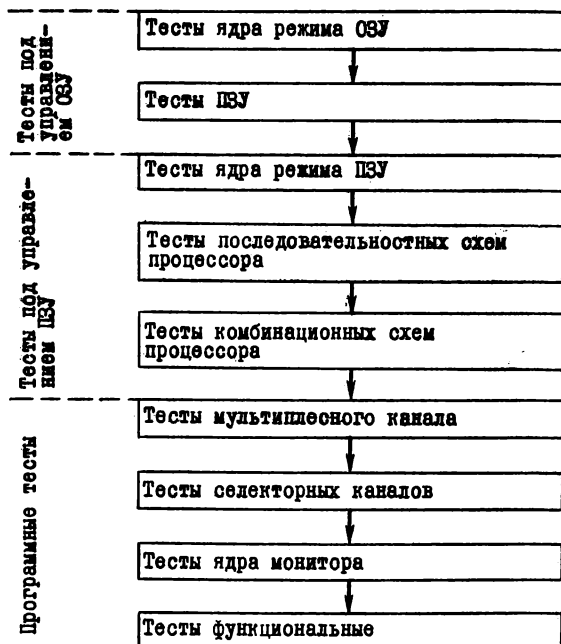


Рис. 7-5. Схема реализации метода раскрутки.

В данном разделе будет показан метод *раскрутки*, положенный в основу комбинированной системы автоматической диагностики ЭВМ ЕС-1030.

Сущность метода *раскрутки* состоит в последовательном расширении работоспособной части ЭВМ. Вначале выбирается некоторая, проверенная заранее часть оборудования, называемая диагностическим ядром ЭВМ. С его помощью проверяется другая выделенная часть оборудования, которая после окончания проверки присоединяется к ядру. Проверенная таким образом расширенная часть аппаратуры

используется для диагностической проверки новой зоны. И до полной проверки всех устройств. На рис. 7-5 приведена схема, поясняющая этапы работы метода раскрутки для диагностики ЭВМ ЕС-1030.

Диагностические тесты хранятся на магнитной ленте и при запуске системы загружаются через селекторный канал в выделенную область памяти ОЗУ объемом 4 К. Эта выделенная область памяти предварительно проверяется оператором вручную с пульта машины.

Проверка выделенной области памяти ОЗУ проводится в режимах записи и чтения нулей и единиц и записи и чтения тяжелого и обратного тяжелого кодов. Эта проверка позволяет локализовать неисправность, если она есть, с точностью до адреса и разряда ОЗУ. В селекторном канале, по которому производится загрузка тестов, предусмотрено автоматическое восьмикратное повторение загрузки. Если все восемь попыток оказались неудачными, то загрузка тестов прекращается и канал проверяется вручную. (Если выбранный селекторный канал оказался неисправным, то загрузка тестов выполняется по одному из других каналов.)

После проверки диагностического ядра ЭВМ диагностика машины выполняется автоматически. Записанные в ядро тесты проверяют исправность тех схем и цепей процессора и ОЗУ, используемые в дальнейшем для проверки постоянного запоминающего устройства (ПЗУ), в котором записаны микропрограммы. Проверяемые схемы и программы образуют ядро режима ОЗУ, в которое входят: регистр адреса ОЗУ с дешифраторами, регистр адреса ПЗУ, регистр коммутатора процессора, цепи передачи информации из регистра информации ПЗУ в регистр информации ОЗУ и т. д.

На следующем этапе осуществляется проверка ПЗУ. Используя тесты, взятые из системы, проверенная часть оборудования приступает к проверке правильности работы схем дешифрации адреса ПЗУ, схем считывания информации и содержимого всех ячеек ПЗУ. При этом управление диагностическими процедурами осуществляется с помощью информации, поступающей из ОЗУ. Неисправность определяется с точностью до разряда и адреса слова ПЗУ.

Дальнейшая проверка осуществляется с использованием диагностических микропрограмм, хранящихся в ПЗУ. С помощью этих программ проверяется ядро режима ПЗУ, в которое входят следующие схемы: специальные триггеры



управления, регистр счета последовательностей и ряд микропрограмм.

После проверки ядра режима ПЗУ начинается проверка процессора с использованием тестов локализации неисправностей (ТЛН). Система предусматривает отдельную проверку последовательностных, включающих элементы памяти, и комбинационных схем. Вначале выполняется проверка последовательностных схем, состоящих из отдельных триггеров и регистров. Для этого под управлением ТЛН последовательно выполняются микропрограммы установки запоминающих элементов в схемах процессора в заданное состояние. Затем с помощью микропрограмм выполняется считывание информации о состоянии запоминающих элементов и сравнение этих состояний с эталонным. При этом последовательно проверяется установка каждого элемента в 0, затем в 1 и потом обратно в 0.

Проверка триггеров и регистров процессора осуществляется с использованием специальных дополнительных линий опроса состояния элементов, которые используются только при диагностике. Особенность этих линий заключается в том, что они позволяют записывать состояние проверяемого регистра без изменения состояния других регистров. Результат проверки индицируется с помощью номера теста. По номеру теста, используя документацию тестов, находят конкретную неисправность. В табл. 7-7 показан пример из таблицы документации тестов проверки последовательностных схем процессора.

Т а б л и ц а 7-7

Пример таблицы документации тестов

Номер теста	Состояние	Конструктивный адрес	Адрес триггера на функциональной схеме	Обозначение
001	0	01A—2D07 (23)	15G, БАЛ-03	р5
002	1	01A—2D07 (23)		
003	0	01A—2D07 (23)		

В первой графе указан номер теста, во второй — состояние триггера под воздействием соответствующего теста, в третьей — показан конструктивный адрес контактов, позволяющий наблюдать за состоянием испытываемого триггера (в данном случае, так как речь идет о триггере, адрес один и тот же), в четвертой — адрес триггера на функцио-

нальной схеме устройства, в пятой — обозначение триггера на функциональной схеме.

Далее выполняются тесты диагностики комбинационных схем процессора, расположенных между запоминающими элементами процессора, проверенными ранее. Эти тесты позволяют локализовать неисправность с точностью от одного до трех ТЭЗ и монтажных связей между ними. Тесты комбинационных схем выполняются в следующей последовательности. Вначале с помощью микропрограммы установки производится установка регистра процессора, расположенного на входе проверяемой комбинационной схемы, в состояние, определяемое заданным кодом. Затем выполняется микрооперация приема выходного сигнала комбинационной схемы на регистр, расположенный на ее выходе. Далее производится опрос этого регистра и его состояние сравнивается с эталонным. При обнаружении неисправности индицируется номер теста, локализирующего неисправность. По номеру теста в документации определяется конкретная неисправность. Тесты диагностики комбинационных схем построены таким образом, что номер каждого следующего теста в последовательности зависит от исхода предыдущего.

После диагностики процессора проверяются каналы. Тесты каналов выполнены программными с использованием специальной команды ДИАГНОСТИКА. Эта команда позволяет связывать программный и микропрограммный режимы управления работой ЭВМ. С ее помощью можно вызвать любую микропрограмму процессора, а затем передать управление программе, опросить состояния регистров после выполнения команды, организовать проверку выполнения команды на любом такте. Вначале с помощью команды программных тестов и команды ДИАГНОСТИКА проверяется мультиплексный канал, а затем проверяются селекторные каналы. При проверке каналов интерфейса ЭВМ логически отключен. Имитация сигналов интерфейса выполняется программой ДИАГНОСТИКА и регистром имитации интерфейса.

На следующем этапе проверяется ядро монитора, в которое входят такие команды ЭВМ, как переключение программ, загрузка слова, загрузка адреса и др.

И, наконец, на последнем этапе осуществляется программная диагностика с помощью функциональных тестов, проверяющих работу всех ячеек оперативной памяти, системы прерывания, интерфейса и периферийных устройств.

**ПРОГРАММНО-АППАРАТУРНЫЕ МЕТОДЫ  
КОНТРОЛЯ И ДИАГНОСТИКИ ЭВМ**

**8-1. СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА АППАРАТУРНЫХ  
И ПРОГРАММНЫХ МЕТОДОВ КОНТРОЛЯ  
И ДИАГНОСТИКИ ЭВМ**

Рассмотренные в предыдущих главах два основных вида средств автоматического контроля — программные и аппаратные — имеют свои достоинства и недостатки.

Программные методы контроля экономичны по затратам оборудования (в простейшем случае при применении метода двойного просчета вообще не требуется дополнительного оборудования), но в то же время они обладают рядом существенных недостатков:

1. Уменьшение фактического быстродействия ЭВМ в результате увеличения числа выполняемых команд.

2. Неполнота контроля, поскольку неисправности, приводящие к останову ЭВМ, вызывают нарушение выполнения и контрольных программ.

3. Отсутствие единой методики программного контроля для различных программ, зависимость применяемых приемов от специфических особенностей конкретной программы.

4. Сложность программирования, обусловленная тем, что эффективность контроля полностью зависит от программиста.

5. Неполнота информации о месте возникновения ошибки, что связано с определенным количеством рабочих операций, выполняемых ЭВМ за промежуток времени между контрольными точками программы. При этом информация, искаженная при ошибке, может быть использована до того, как ошибка будет обнаружена.

Аппаратурный контроль может выполнить все те функции, что и программный, и еще ряд функций, которые невозможно или затруднительно осуществить программным методом.

В первую очередь, аппаратурный контроль позволяет обнаружить ошибку в момент ее возникновения, предоставляя при этом оператору или системе автоматической диагностики более точную информацию об искажениях, вызванных неисправностью. Кроме того, аппаратурный контроль может обнаружить перемежающиеся ошибки, недоступные программным методам.

## 8-2. ОСНОВНЫЕ ПОНЯТИЯ ПРОГРАММНО-АППАРАТУРНЫХ МЕТОДОВ КОНТРОЛЯ И ДИАГНОСТИКИ ЭВМ

Программно-аппаратурный (комбинированный) метод обнаружения и исправления ошибок является наиболее совершенным с точки зрения обеспечения достоверности вычислений в целом. Сущность этого метода заключается в том, что обнаружение ошибок осуществляется методами аппаратного контроля с помощью контрольных схем, встроенных во все основные устройства ЭВМ и тракты передачи информации, а исправление обнаруженных ошибок осуществляется специальной исправляющей программой, работа которой инициируется сигналами ошибок с контрольных схем.

Комбинированный метод контроля обладает достоинствами как аппаратного, так и программного методов контроля. Он, с одной стороны, требует меньших аппаратных затрат по сравнению с аппаратным методом, решающим те же задачи, и, с другой стороны, по сравнению с программным методом в меньшей степени уменьшает производительность ЭВМ, так как исправляющая программа включается только после обнаружения ошибки, т. е. достаточно редко.

Применение программно-аппаратурного метода контроля позволяет полностью автоматизировать процесс устранения влияния случайных ошибок на результаты решения задач без существенного снижения производительности ЭВМ и при незначительном увеличении объема аппаратуры.

Контрольные схемы будут обнаруживать также и систематические ошибки, появляющиеся вследствие отказов в элементах ЭВМ. Для сокращения времени поиска отказавших элементов и их замены целесообразно в ЗУ ЭВМ хранить также и диагностическую программу в виде нескольких микропрограмм или микротестов.

Собственно диагностическая микропрограмма будет в этом случае значительно проще, чем программа, составленная для ЭВМ без аппаратного контроля. Контрольная схема, обнаружившая систематическую ошибку, уже указывает на устройство, в котором произошел отказ. Таким образом, задача диагностической микропрограммы в этом случае сводится лишь к обнаружению места отказа с точностью до одного ТЭЗ.

Работу ЭВМ при наличии исправляющей и диагностических программ можно представить себе, например, следующим образом. В случае появления обнаруженной ошибки,

возникшей при хранении информации в ЗУ, при передаче ее или обработке, соответствующая контрольная схема выдает сигнал ошибки в устройство приоритетного прерывания (имеется в виду ЭВМ с многопрограммной обработкой данных). По этому сигналу происходит прерывание рабочей программы, и управление передается исправляющей программе, имеющей наивысший приоритет. Выполнение этой программы обеспечивает сохранение состояния всех регистров АУ и УУ в момент прерывания, а затем производится попытка исправить ошибку путем повторения прерванной операции. Если это удастся, ошибка квалифицируется как случайная и управление возвращается к рабочей программе для ее продолжения. Если же при 2—3 повторениях операции снова возникает сигнал ошибки в той же контрольной схеме, то контролируется наличие отказа в аппаратуре либо разрушение исходной информации при ее записи или хранении. В этом случае делается попытка определить ошибочный разряд и исправить ошибку как в данной операции, так и в последующих, если она будет появляться. По окончании цикла вычислений, а иногда и сразу после обнаружения систематической ошибки (без попытки исправления) управление передается той или иной диагностической программе для локализации и устранения отказа.

Может быть предусмотрена возможность перехода на другую рабочую программу, при которой неисправный блок (например, один из шкафов НМЛ) не используется. При этом оператору выдается сигнал отказа, а диагностика и устранение этого отказа откладываются до более благоприятного времени. Данные обо всех ошибках, состоянии регистров в момент прерывания и сообщении о мерах по устранению ошибок и диагностике отказов, принятых исправляющей и диагностическими программами, автоматически выдаются на печать. Анализ такой информации чрезвычайно полезен для организации эксплуатации ЭВМ, в частности для планирования профилактики.

На исправляющую программу возлагаются функции анализа ошибок и их исправления.

Каждая контрольная схема ЭВМ при обнаружении ошибки выдает сигнал, поступающий на определенный разряд специального регистра ошибок. Одновременно с этим сигнал ошибки поступает на систему прерывания, автоматически останавливая процесс вычислений с запоминанием необходимой информации в запоминающем устройстве.

Исправляющая программа обычно хранится в ПЗУ. Для ее обслуживания выделяется необходимое число ячеек оперативной памяти. Исправляющая программа состоит из общей части и отдельных подпрограмм, каждая из которых рассчитана на определенный тип ошибок. Каждая подпрограмма имеет выходы в диагностическую и управляющую программы.

Назначением общей части исправляющей программы является определение характера ошибки (случайная или систематическая) и исправление случайных ошибок. Общая часть исправляющей программы включается аппаратным путем при любом состоянии регистра ошибок, отличном от нулевого, т. е. при появлении одной (или нескольких одновременно) ошибки.

Обе функции, возложенные на общую часть программы, выполняются путем повторения прерванной операции. Для всех операций по вводу, выводу и внутренним передачам информации это не связано с затруднениями, так как чтение информации в современных ЗУ и тем более при вводе с перфокарт и перфолент происходит без ее разрушения. Во всяком случае она сохраняется до конца передачи всего блока. Что касается арифметических и логических операций, то они часто выполняются с использованием результата предыдущей операции. В одноадресных машинах подобная ситуация вполне естественна. Чтобы иметь возможность повторять такую операцию, операнд должен храниться до конца контролируемой операции в особом регистре, который непосредственно не участвует в работе АУ. При отсутствии в составе АУ такого регистра должно быть организовано повторение всего цикла вычислений или блока рабочей программы. Так как блок программы при работе над ним может быть изменен, его необходимо заново ввести в ОЗУ из внешнего накопителя. Для осуществления такого повторения в одной из рабочих ячеек ОЗУ, отведенных для обслуживания исправляющей программы, предусматривается хранение входа в выполняемый в данный момент цикл вычислений. По указанным выше соображениям общая часть программы должна состоять по крайней мере из двух ветвей, соответствующих двум видам повторений. Переход к той или иной ветви определяется состоянием регистра ошибок. Операция повторяется с задержкой на время, равное нескольким десяткам секунды. Необходимость задержки обусловлена тем, что причина, вызвавшая сбой (флуктуация питающих напряжений, нарушение контакта из-за

вибраций и т. д.), может существовать в течение некоторого времени, достаточного для выполнения сотен или даже тысяч операций. При повторении без задержки ошибка, случайная по своему характеру, могла бы квалифицироваться программой как систематическая.

Перед каждым повторением операции на печать или панель индикации выдается справка о состоянии регистра ошибок с отметкой текущего времени (если это предусмотрено конструкцией ЭВМ).

Эти сведения необходимы для набора статистических данных по случайным сбоям с целью правильной организации необходимых профилактических работ, а также для облегчения поиска и устранения неисправности.

### **8-3. ОБНАРУЖЕНИЕ ОШИБОК ЦЕНТРАЛЬНОГО ПРОЦЕССОРА ЭВМ МИКРОПРОГРАММНЫМИ МЕТОДАМИ**

Как известно, работоспособность сложных систем, какими являются современные ЭВМ, ухудшается не только с ростом интенсивности отказов, но и с увеличением продолжительности восстановления их готовности к работе. В связи с этим большое значение приобретает задача улучшения характеристик восстанавливаемости: времени поиска отказавшего элемента и времени устранения отказа. Опыт эксплуатации ЭВМ подсказывает, что наиболее продолжительным этапом ремонта ЭВМ является процесс поиска отказавшего элемента. Поэтому в современных ЭВМ третьего поколения предусмотрены специальные меры, позволяющие ускорить поиск неисправного элемента (узла, блока). К таким мерам относятся диагностические средства ЕС ЭВМ, предназначенные для обнаружения и локализации возможных неисправностей в оборудовании ЭВМ с точностью до одного или нескольких ТЭЭ.

Методы аппаратурного и программного контроля ЭВМ (например, с помощью программы ТЕСТ-монитора) были подробно рассмотрены в предыдущих главах. Целью настоящей главы является описание средств микропрограммной диагностики ЕС ЭВМ на примере ЕС-1022.

Микропрограммные средства предназначены для выполнения диагностических процедур на уровне микропрограмм и состоят из нескольких диагностических микропрограммных тестов, каждый из которых с помощью специальных дополнительных аппаратурных средств выполняет проверку работоспособности основных блоков машины: центрального

процессора, запоминающих устройств, каналов ввода-вывода. Все микропрограммные тесты выполняются в диагностическом режиме, который задается с помощью установки переключателей ДИАГНОСТИКА на панели управления машины в положение, соответствующее диагностическому блоку. При этом переключатели РЕЖИМ РАБОТЫ, КОНТРОЛЬ, СРАВНЕНИЕ АДРЕСОВ должны быть установлены в положение АВТОМАТ (рис. 8-1). Положение переключателя ТИП ПАМЯТИ безразлично.

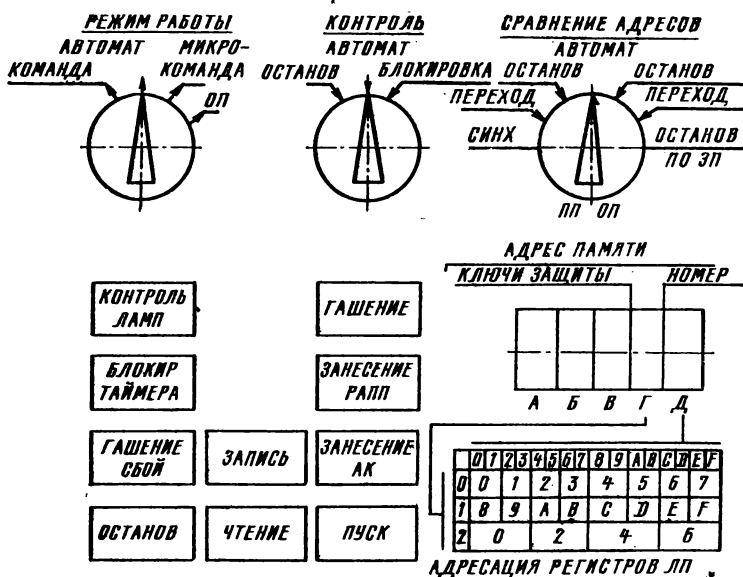


Рис. 8-1. Панель уп

Переключатель ДИАГНОСТИКА имеет следующие положения при выполнении диагностических микропрограмм:

*КМ* — канала мультиплексного; *КС1* — первого селекторного канала; *КС2* — второго селекторного канала; *ЦП* — центрального процессора; *ОП* — оперативной памяти; *ЛП* — локальной памяти.

В положении АВТОМАТ этого переключателя диагностические микропрограммы не выполняются.

Кроме этих переключателей на панели управления имеются еще переключатели набора информации. Набор АДРЕС

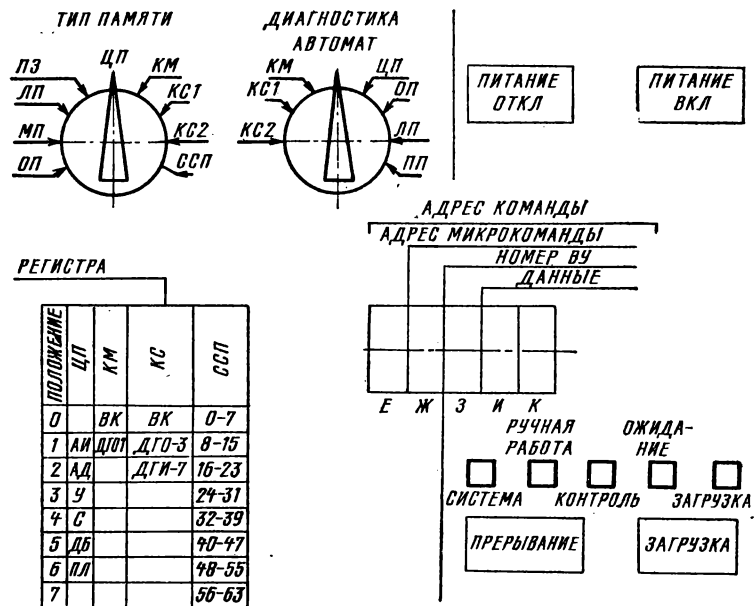


ПАМЯТИ состоит из пяти переключателей (А, Б, В, Г, Д) на 16 положений каждый и предназначен для задания:

адреса оперативной памяти при чтении или записи информации;

адреса для сравнения с адресом ОП (на переключателях А—Д) или ПЗУ (на переключателях Б—Д);

номера регистра канала или центрального процессора при выводе их содержимого в регистр индикации (на переключателе Д).



руления ЕС-1022.

Набор АДРЕС КОМАНДЫ также имеет пять переключателей (Е, Ж, З, И, К) и предназначен для задания:

адреса текущей команды (на переключателях Е—К);

адреса ПЗУ (на переключателях Ж—К);

адреса (номера) устройства ввода-вывода при вводе начальной программы (на переключателях З—К);

данных (на переключателях И—К).

Пульт управления ЕС ЭВМ помимо панели управления, показанной на рис. 8-1, имеет также панель индикации. На этой панели расположен ряд лампочек, с помощью которых индицируется содержимое регистров ПЗУ, регистров

мультиплексного и селекторных каналов, регистра индикации, а также состояние отдельных управляющих триггеров и регистров центрального процессора (регистра данных основной памяти РД, входных ГРА и ГРБ и выходного ГРВ регистров арифметического блока, регистра ошибок и т. д.).

При нажатии кнопки ПУСК происходит запуск выбранного диагностического теста. При неверном выполнении теста происходит микропрограммный останов и на панели индикации основной памяти и процессора загорается лампочка индикатора, т. е. тяжелый останов, при верном — текст зацикливается. Для останова любого микропрограммного теста необходимо переключатель РЕЖИМ РАБОТЫ установить в положение МИКРОКОМАНДА или нажать кнопку ГАШЕНИЕ.

Для создания непрерывной цепочки контроля работы центрального процессора имеется ряд контрольных точек контролируемых регистров или блоков:

- 1) информационный регистр памяти ключей защиты (ГРКЛ);
- 2) регистр ГРА АЛУ;
- 3) регистр ГРБ АЛУ;
- 4) регистр адреса оперативной памяти (ГРАОП);
- 5) регистр данных оперативной памяти (ГРД);
- 6) регистр микрокоманды (ГРМК);
- 7) регистр адреса постоянного запоминающего устройства (ГРАПП);
- 8) арифметическо-логическое устройство (АЛУ);
- 9) регистр адреса локальной памяти (ГРР);
- 10) регистр данных локальной памяти (ГРЛ).

Появление сбоя в какой-либо контролируемой точке приводит к установке соответствующего разряда в регистре сбоев (ГРСБ). Для контроля адреса в каждой микрокоманде предусмотрен разряд ГРМК, значение которого определяется как дополнение до нечетного числа единиц в ГРАПП, по которому выбирается эта микрокоманда.

Рассмотрим подробнее работу с микропрограммным базисным тестом процессора. Он предназначен для функциональной проверки работоспособности оборудования центрального процессора. Тест состоит из тринадцати блоков проверок, которые позволяют проверить выполнение отдельных функций центрального процессора.

Первый блок проверяет формирование адреса перехода или занесения естественного адреса (адреса пульта) в зависимости от положения переключателей на пульте управления.

## Словарь неисправностей базисного микротеста (ДГЦП)

ГРАПП	Индикация пульты управления				Предполагаемая причина неисправности	Примечание
	ГРА	ГРБ	ГВ	ГРД		
0 03	00E7	ГРСБ	≠0	—	Схемы контроля машины Регистр сбоев ГРСБ	В ГРСБ должно быть E7
0 07	ГРД	ГРС	—	FFFF	Микрооперация РБС-0 Выход коммутатора ЛП	—
0F0A	00AA	ГРД2	—	—	Регистр ГРД Регистр ГРСЗ	В ГРД2 должно быть AA
0F0D	ГРС2,3	0001	≠0	—	Цепи, выполняющие СДВИГ ВЛЕ- ВО	
0F0E	ГРС2,3	0001	≠0	—	Микрооперация 1ПН поля УСЛО Операция сложения по модулю 2 Сложение по модулю 2 не во всех разрядах дает 1	—
0F0F	ГРС2,3	0000	FFFF	—	Неверно работают цепи сложения FFFF + 1	—
0F13	ГРУ2,3	ГРД2,3	≠0	—	Нет перехода по L = FF	—
0F16	0000	ГРС2,3	≠0	—	Неверно выполняется деформация X Цепи выполнения операции ИЛИ, ИЛИ-НЕ	В ГРС2 должно быть FFFF

ГРАПП	Индикация пульта управления					Предполагаемая причина неисправности	Примечание
	ГРА	ГРБ	ГВ	ГРД	ГРД		
0F18	ГРС2,3	ГРУ2,3	≠0	—	—	Цели выполнения операции НЕ-И	В ГРС2,3 и ГРУ2,3 должно быть 5555
0F1A	ГРС2,3	ГРД2,3		AAAA		Микрооперация А транзит	
0F1C	ГРС2,3	ГРУ2,3				Микрооперация В транзит	
0F21	00FF	ГРД	≠0	Константа счетчика		Цели анализа счетчика ГС3 (L = FF)	В ГРС2,3 и ГРУ2,3 должно быть 5555
0F28	ГРС3	ГРД2	≠0			Цели вычитания в старшем байте Микрооперация в старшем байте	
0F2C	00ED	ГРС3				Не модифицируется счетчик ГРС3 по микрооперации МАВ	
0F2E	00FB	ГРС3				Не модифицируется счетчик ГРС3 по микрооперации МВВ	
0F31	00FB	ГРС3				Счетчик ГРС3 модифицируется при ОС27	
0F32	ГРУ2,3	ГРД2,3		00A5		Неверно выполняется деформация «накрест»	В ГРУ2,3 должно быть 005A
0F3B	ГРАИ2,3	0001	≠0			Регистр ГРАИ2,3 Дешифратор поля В микрокоманды Дешифратор поля А микрокоманды	В ГРАИ должно быть FFFF

Второй блок проверяет работу схем контроля: с помощью специальной диагностической микрооперации имитируются сбои оборудования, информация заносится в регистр ошибок и сравнивается там с контрольной константой E7.

Третий блок проверяет работу арифметическо-логического устройства (АЛУ): занесение единиц через АЛУ, узла переноса, сложение по модулю 2; наличие всех управляющих сигналов и выполнение всех логических микроопераций АЛУ; работу арифметического устройства при вычитании и сложении двоичных чисел и т. д.

Четвертый и пятый блоки проверяют формирование адресов перехода и хранения информации в регистрах центрального процессора, а также работу входных и выходных дешифраторов АЛУ. Проверка организована как цикл, во время которого все регистры, кроме одного, содержат нули. В проверяемый регистр засылаются единицы. При неверной работе дешифраторов возможна засылка их в другой регистр. Затем сигналы с регистра, содержащего единицы, подаются на входы АЛУ, т. е. проверяется и регистр, и работа входных дешифраторов. В следующем цикле регистр, содержащий единицы, обнуляется, а единицы засылаются в следующий регистр и т. д.

Шестой и седьмой блоки проверяют работу входных и выходных дешифраторов локальной памяти (ЛП). Проверка аналогична предыдущей.

Восьмой блок проверяет работу счетчиков. Проверяются все возможные установки счетчиков. Результаты сравниваются с контрольной константой.

Девятый блок проверяет приоритет занесения информации в регистр ГРД из оперативной памяти и с выходов АЛУ и ЛП в различных режимах.

Десятый, одиннадцатый и двенадцатый блоки предназначены для дальнейшей проверки работы АЛУ: десятичного сложения, формирования адресов перехода полей АЛУ, функционирования элементов, хранящих результат обработки данных в АЛУ, и т. д.

Тринадцатый, последний блок осуществляет проверку адресации локальной памяти.

В случае наличия неисправности при работе блоков 1—13 происходит диагностический останов микротеста.

Предполагаемое место неисправности указывается в специальном словаре неисправностей, связывающем индикацию пульты управления в момент останова микротеста с узлом или блоком, который может быть неисправен. Словарь неисправностей входит в состав инструкции по эксплуатации ЕС ЭВМ в раздел «Диагностика». Пример страниц словаря ЭВМ ЕС-1022 приведен в табл. 8-1. Наличие словаря неисправностей дает возможность обслуживающему персоналу в короткий срок локализовать место неисправности и значительно сокращает время ремонта ЭВМ.

#### 8.4. МИКРОПРОГРАММНЫЕ ТЕСТЫ ЗАПОМИНАЮЩИХ УСТРОЙСТВ И КАНАЛОВ ВВОДА-ВЫВОДА

Проверка работоспособности запоминающих устройств процессора ЕС ЭВМ производится при помощи ряда микропрограммных тестов, диагностирующих работу постоянной памяти, локальной памяти, оперативной памяти и блока защиты. К ним относятся:

- 1) микропрограмма входа в микротесты узлов ЭВМ и сканирования постоянной памяти (тест ДГВХ);
- 2) микротест локальной памяти (ДГЛП);
- 3) микротест оперативной памяти и блока защиты (тест ДГОП).

Рассмотрим работу этих тестов подробнее. Микротест ДГВХ предназначен для:

формирования входного адреса диагностической программы в зависимости от положения переключателя ДИАГНОСТИКА на пульте управления;

подготовки граничных адресов проверяемой области оперативной памяти для теста ДГОП;

контроля сбоя по защите или адресации оперативной памяти на необходимость прерывания.

Пуск теста осуществляется последовательным нажатием кнопки ГАШЕНИЕ на пульте управления, установкой переключателя ДИАГНОСТИКА в положение, соответствующее контролируемому узлу, и нажатием кнопки ПУСК.

Для обнаружения неисправностей в схемах постоянной памяти предусмотрено сканирование, т. е. последовательное считывание микрокоманд из постоянной памяти и контроль на НЕЧЕТ их содержимого. Граничные адреса сканируемой области постоянной памяти набираются на переключателях АДРЕС ПАМЯТИ (начальный адрес) и АДРЕС КОМАНДЫ (конечный адрес).

Микротест ДГЛП предназначен для диагностики локальной памяти путем ряда проверок.

Вначале в одну ячейку локальной памяти записываются нули. Затем идет чтение записанной информации и сравнение ее с эталонной. При несовпадении происходит диагностический останов теста. Затем нули переписываются в следующую ячейку и вновь сравниваются с эталоном. Этот циклический процесс продолжается до тех пор, пока не будет проверен весь объем ЛП.

Следующая проверка заключается в циклическом просмотре всех ячеек ЛП на запись единиц, причем единицы

записываются в какую-либо одну ячейку памяти, а в остальных хранятся нули. Проверяется правильность записи единиц и хранения нулевой информации.

Если проверка показала правильность работы теста, то ячейка, содержащая единицы, обнуляется. Единицы записываются в следующую ячейку. Этот процесс продолжается до тех пор, пока не будет проверен весь объем ЛП на хранение единиц.

Последняя проверка состоит в том, что по всему объему ЛП записывается информация с выхода арифметического блока при максимальном времени срабатывания АЛУ, затем считывается и сравнивается с эталоном. При несовпадении происходит диагностический останов.

Микротест ДГОП предназначен для обнаружения неисправностей в работе основной, мультиплексной памяти и блока защиты. Он состоит из четырех частей:

- 1) общей проверки основной памяти ОП и мультиплексной памяти (МП);
- 2) проверки основной памяти в тяжелых режимах;
- 3) проверки основной памяти при низкой частоте обращения;
- 4) проверки блока защиты.

В первую часть теста включены следующие проверки основной и мультиплексной памяти:

- чтение с перезаписью нулей и единиц в ОП и МП;
- чтение из ОП и МП и регенерация нулей и единиц с максимальной частотой обращения по сравнению с эталонной;
- запись в ОП, чтение и сравнение с эталоном кода, равного адресу;
- проверка дешифрации номера стойки и номера блока ОП;
- запись в МП и чтение с контролем кода, задаваемого переключателями на пульте управления.

Вторая часть микротеста позволяет провести:

- чтение с перезаписью тяжелых кодов;
- чтение и регенерацию тяжелых кодов с максимальной частотой обращения к ОП;
- проверку ОП на тяжелых кодах с многократным разрушением информации (так называемая «долбежка»);
- проверку ОП на случайных кодах.

Вход в микротест ДГОП осуществляется из микротеста ДГВХ при установке переключателя ДИАГНОСТИКА в положение ОП. В микротесте ДГВХ подготавливаются начальный и конечный адреса основной памяти для выполнения проверок в заданной области.

В микротесте ДГОП предусмотрены различные режимы зацикливания. Например, можно зациклить только первую часть теста, или только первую и вторую части, или только четвертую часть теста. Режим зацикливания выбирается положением переключателей набора АДРЕС КОМАНДЫ на пульте управления.

При неправильном выполнении микротеста происходит диагностический останов. Адреса всех остановов и соответствующая индикация на пульте управления с указанием местонахождения возможной неисправности приведены в специальном словаре неисправностей, аналогичном базисному микротесту процессора. Пример страницы словаря приведен в табл. 8-2.

Контроль работы оборудования каналов осуществляется по отдельным блокам или регистрам. Каждый блок и регистр просматриваются отдельной диагностической проверкой, которая состоит из микрокоманд, осуществляющих входное воздействие на эти блоки или регистры канала. Набор этих микрокоманд составляет диагностический микротест соответствующего канала, который осуществляет выполнение проверок в определенной последовательности.

Диагностический микротест канала:

воздействует на входы регистра или схемы канала;  
сравнивает с эталоном реакцию испытываемого участка на входное воздействие;  
выдает сообщение о результатах проверки участка.

При построении микротеста используется принцип установки и сброса каждого триггера, входящего в состав оборудования каналов. Все триггеры, входящие в регистры, проверяются после каждой установки или сброса по всем входам. Состояние проверяемого регистра, эталон и результат индицируются во входных регистрах ГРА и ГРБ и на выходе АЛУ.

Микропрограммные средства предназначены в основном для обнаружения ошибки в схемах управления каналами и интерфейсом, например:

а) на линиях интерфейса одновременно присутствуют несовместимые сигналы идентификации канала (УПРК — управление от канала и ИНФК — информация от канала) или идентификации абонента (УПРА — управление от абонента и АДРА — адрес от абонента);

б) сбой по четности в передаваемой по каналу числовой и управляющей информации;



Т а б л и ц а 8-2

## Словарь неисправностей микротестов ДГЛП и ДГОП

Адрес останова	Номер проверки	Причина	Индикация пульты управления				Местонахождение неисправности в тестах	
			ГРД	Регистр индикации	ГРАОП	ГРН		
0004	2	Прочитанная из ОП информация не сравнилась с эталоном	Прочитанная информация	—	Адрес ОП	—	ОП	—
00FF	1	Не сравнились два младших байта	Нулевая информация	Нули	—	Адрес ЛП	—	ЛП
00FC	1	Не сравнились два старших байта	»	»	—	То же	—	»
00E5	2	Не сравнивалась прочитанная информация с эталоном	»	Эталон	—	»	—	»
00FB	3	Не сравнились два младших байта	»	»	—	»	—	»
00FC или 00FF	3	Не сравнились два старших байта	»	»	—	»	—	»
0E08, 0E09	0, 2, 4	Останов по нечету при записи в ОП	Записываемая информация	Прочитанная информация	Адрес ОП	—	ОП	—
0E0B, 0E0C	1, 3	Не сравнивалась прочитанная информация с эталоном	Прочитанная информация	Эталон	То же	—	»	—

Адрес останова	Номер проверки	Причина	Индикация пульта управления					Местонахо- ждение неис- правности в тестах
			ГРД	Регистр индикации	ГРАОП	ГРН	ДГОП	
0Е12, 0Е13	6	Не сравнивалась про- читанная информация с эталоном	Прочитанная информация	Эталон	Адрес ОП	—	ОП	—
0Е19, 0Е3, 0Е5 0029	7	Сбой по нечету при за- писи констант в ОП Прочитанная из ОП ин- формация не сравнилась с эталоном. Нет сбоя по защите в режиме «защита по записи»	Записываемая информация Прочитанная информация	Не исполь- зуется То же	То же » »	—	» »	— —
0Е2А 0Е2, 0Е44 0Е5С	5 9, 11	Останов по нечету при записи в ОП Сбой по нечету при считывании констант Сбой по нечету при считывании тяжелого кода	Записываемая информация Прочитанная информация То же	Не исполь- зуется То же » »	» » » » » »	—	» » »	— — —

в) байт адреса или состояния УВВ содержит неверные контрольные разряды;

г) нарушены временные соотношения в последовательностях сигналов интерфейса.

Микропрограммные диагностические средства каналов ввода-вывода реализуются с помощью двух тестов:

микротеста мультиплексного канала ДГКМ;

микротеста селекторных каналов ДГКС.

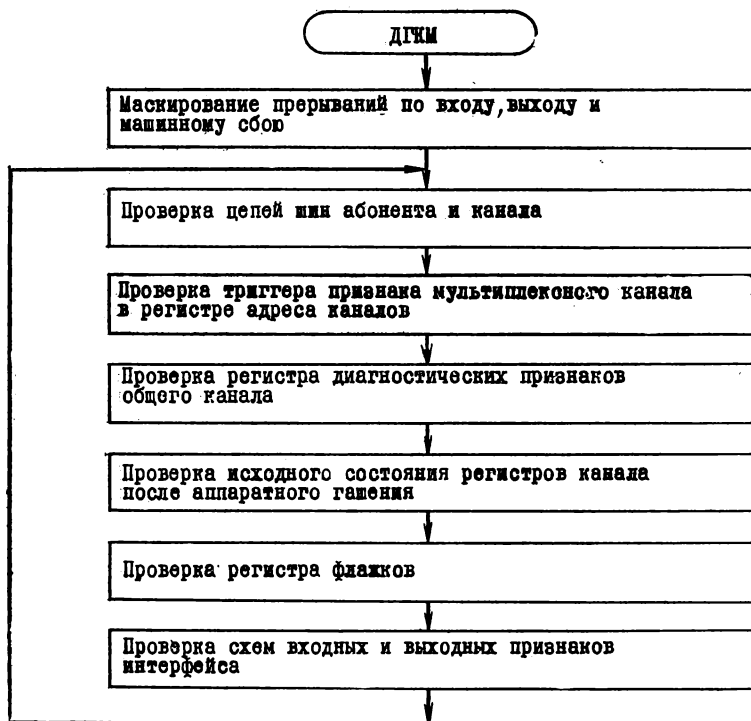


Рис. 8-2. Блок-схема микротеста мультиплексного канала.

Микротест ДГКМ занимает 213 микрокоманд, а тест ДГКС — 379 микрокоманд. Микротест для селекторных каналов является общим для двух каналов, так как диагностируемые части оборудования идентичны.

С помощью микропрограммных тестов каналов производится также проверка цепей общего канала, связанных с диагностируемым каналом. Кроме того, для выполнения

этих микротестов предусмотрено специальное диагностическое оборудование (аппаратурные средства) как в общем канале, так и в мультиплексном и селекторных каналах: в общем канале — регистр диагностических признаков, в мультиплексном — сборка входных сигналов интерфейса, в селекторных — имитаторы входных воздействий.

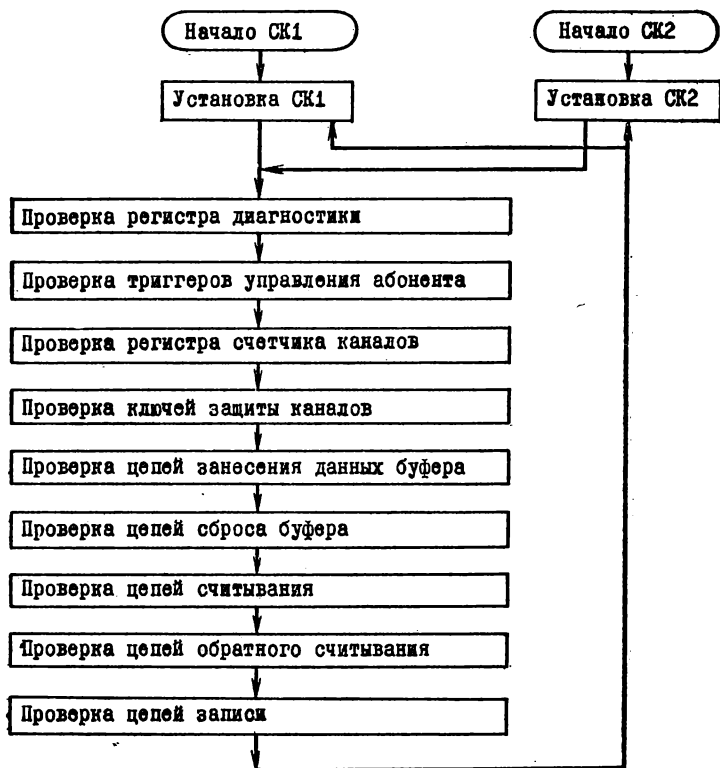


Рис. 8-3. Блок-схема микротеста селекторных каналов.

При использовании диагностических средств каналов ввода-вывода необходимо иметь исправным диагностическое ядро — оборудование центрального процессора ЭВМ. Поэтому диагностику каналов ввода-вывода необходимо проводить после диагностики и, если необходимо, ремонта устройств центрального процессора. Блок-схемы микротестов приведены на рис. 8-2 и 8-3.

## Словарь неисправностей микротестов каналов

Адрес ПЗУ	Предполагаемая неисправность	Адреса ТЭЗ с предполагаемой неисправностью		
		СК1	СК2	МК
ОВА8	Нет установки триггера ЕТУПР-К	—	—	5С33 5А31 1А33
ОВАС	Нет сброса триггеров ЕТУСЛКМ, ЕТОБМ	—	—	5А37, 5А36 5С30, 5А43 5А30
ОВАЕ	Нет установки триггера ЕТКМКОМ	—	—	5А29, 5А36 5А45, 5А33 5С11, 5С12 А137, 5А38 5А36
ОВАГ	Нет установки триггеров ЕТУСТК1, ЕТУСТК2	—	—	
ОВВ3	Нет сброса триггера ЕТБЛК-К	—	—	5С33
ОВВ6	Нет сброса триггера ЕТБЛК-К	—	—	5С33
ОВВ9	Нет сброса триггера ЕТБЛК-К	—	—	5С33
ОВС0	Нет сброса триггера ЕТУПР-А	—	—	5С33
ОВС3	Нет установки триггера ЕТИНФ-К	—	—	5С33, 5А31 1А34
ОВС6	Нет сброса триггера ЕТИНФ-А	—	—	5С33
ОВС9	Нет установки триггера ЕТВВБР-К	—	—	5С32
ОВСС	Нет сброса триггера ЕТВВБР-К	—	—	5С32, 5А43
—	Нет занесения в регистр ГРСЧ	5Е39, 5А43 5А41, 1А14 5С11, 5С12 1А07, 1А38 1А39, 1А40 1А41, 5А45	5Е39, 5А43 5А41, 1С14 5С11, 5С12 1С07, 1А38 1А39, 1А40 1А41, 5А45	—
—	Нет занесения в регистр ГРСЧ	5А43, 1А07	5А43, 1С07	—
—	Нет сброса регистра ГРСЧ по гашению	1А07, 1А14 5А27, 5А30 5С11, 5А36 5А33, 5С12	1С14, 1С07 5А27, 5А30 5С11, 5А36 5А33, 5С12	—
—	Нет сброса регистра ГРКК по гашению	5А40, 5А06 1А38, 1А39 5А27	5А40, 5С06 1А38, 1А39 5А27	—

Продолжение табл. 8-3

Адрес ПЗУ	Предполагаемая неисправность	Адреса ТЭЗ с предполагаемой неисправностью		
		СК1	СК2	МК
—	Нет занесения в регистр ГРКК	5A40, 5A06 1A38, 1A39 5A12	5A40, 5C06 1A38, 1A39 5A12	—
—	Нет занесения К-0 в регистр ГРКК	5A06	5C06	—
—	Нет сброса регистра ГРКК по гашению	5A06, 5A27	5C06, 5A27	—
—	Нет сброса буфера по гашению	1A23, 5A07 1A18, 1A14 5C12, 1A22 5A40, 5A12 1A10, 1A17 1A16, 1A19 1A20, 1A21 1A38, 1A39 1A40, 1A41	1C23, 5C07 1C18, 1C14 5C12, 1C22 5A40, 5A12 1C10, 1C17 1C16, 1C19 1C20, 1C21 1A38, 1A39 1A40, 1A41	—
—	Нет сброса управления буфером после гашения	1A23, 5A07 1A18, 1A14 5C12, 1A22 5A40, 5A12 1A10, 1A17 1A14, 5C12 5C11, 1A30 1A31, 1A32 1A33, 1A34 1A35, 1A36 1A37, 5A31	1C23, 5C07 1C18, 1C14 5C12, 1C22 5A40, 5A12 1C10, 1C17 1C14, 5C12 5C11, 1A30 1A31, 1A32 1A33, 1A34 1A35, 1A36 1A37, 5A31	—

При диагностике каналов ввода-вывода в случае несоответствия результатов входных воздействий, микротестов ДГКМ и ДГКС с эталоном происходит диагностический останов. Локализация места неисправности происходит по словарию неисправности каналов ввода-вывода, имеющемуся в инструкции по техническому обслуживанию машины. Пример страницы словаря приведен в табл. 8-3.

Для запуска канальных микротестов необходимо:

- а) привести диагностируемый канал в готовность к работе в системе;
- б) включить питание кнопкой ПИТАНИЕ ВКЛ на панели управления ЦП;

в) переключатель ДИАГНОСТИКА поставить в положение СК1, СК2 или МК в зависимости от диагностируемого канала;

г) переключатели РЕЖИМ РАБОТЫ, КОНТРОЛЬ, СРАВНЕНИЕ АДРЕСОВ установить в положение АВТОМАТ;

д) переключатель ТИП ПАМЯТИ — в положение ОП;

е) нажать последовательно кнопки ГАШЕНИЕ и ПУСК.

## ГЛАВА ДЕВЯТАЯ

### ОСНОВЫ ОРГАНИЗАЦИИ И ПЛАНИРОВАНИЯ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ ЭВМ

#### 9-1. ЦЕЛИ И ЗАДАЧИ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ ЭВМ

Техническое обслуживание ЭВМ осуществляется для поддержания работоспособности всех устройств машины в процессе эксплуатации. Все мероприятия, присущие техническому обслуживанию, можно разделить на три группы:

- 1) контроль технического состояния;
- 2) профилактические работы;
- 3) текущее техническое обслуживание.

Контроль технического состояния ЭВМ является эффективным средством повышения надежности ее работы. Методы и способы контроля как с помощью самой ЭВМ, так и при наличии специальной аппаратуры рассмотрены в предыдущих главах.

Профилактические работы направлены на поддержание работоспособности в течение определенного промежутка времени и продление ее технического ресурса.

Под текущим техническим обслуживанием понимают комплекс повседневных профилактических мероприятий и мелкий ремонт путем замены или восстановления деталей, узлов, блоков ЭВМ.

Проведение технического обслуживания ЭВМ осуществляется службой технического обеспечения вычислительного центра, производственная деятельность которой заключается в освоении, эксплуатации и модернизации всех вычислительных устройств ВЦ.

Служба технического обеспечения, как правило, состоит из следующих групп обслуживания:

- 1) ЭВМ;

- 2) внешних устройств;
- 3) средств связи и телеобработки;
- 4) периферийных устройств;
- 5) вспомогательного оборудования (кондиционеры, лифты и т. п.).

В зависимости от структуры ВЦ и объема работ на каждом производственном участке количество групп и число обслуживающего персонала в них могут меняться.

Основной задачей всех групп технического обслуживания является обеспечение бесперебойной работы закрепленных за ними технических средств в соответствии с системой плано-предупредительного ремонта. Эта система предусматривает два вида работ: повседневное техническое обслуживание (ежесменный профилактический осмотр, чистка, смазка и мелкий ремонт) и плановые регламентные работы.

При круглосуточной работе ЭВМ оперативно-техническое обслуживание ведется сменным персоналом, организованным в дежурные группы (смены), и бригадой профилактики и ремонта. Для обеспечения непрерывной работы ЭВМ удобно иметь 3—4 дежурные группы, с помощью которых можно организовать 8- или 12-часовую сменную работу.

При сменной работе следует обращать особое внимание на необходимость контроля работоспособности машин при передаче от одной дежурной смены к другой. Результаты контроля работоспособности и факт передачи дежурства следует оформлять в специальном журнале.

Обычно численность и квалификация инженерно-технического персонала, необходимого для технического обслуживания ЭВМ, указывается в эксплуатационной документации на машину.

Руководство дежурной группой (сменой) осуществляет начальник смены (инженер смены). Он является ответственным лицом за выполнение всех работ, производимых на ЭВМ во время его дежурства. Начальник смены должен организовать работу своей группы, следить за выполнением запланированных на его смену работ, производить инструктаж операторов и программистов, работающих за пультом управления ЭВМ, принимать меры по выполнению правил безопасности и устранению возникших неисправностей в работе ЭВМ.

Численный состав и классификация бригады профилактики и ремонта определяются видом выполняемых работ, ремонтпригодностью ЭВМ, сложностью функционального и конструктивного построения машины.



Решение задач бригадной профилактики и ремонта может быть обеспечено, если в ней имеются наиболее квалифицированные инженеры и механики по отдельным вычислительным устройствам, а также специалисты с большим опытом работы, знающие машину в целом, ее тестовое и функциональное обеспечение.

Перспективной формой технического обслуживания является сеть централизованного технического обслуживания (ЦТО), которая позволяет решать проблему запасных частей и дефицита технических кадров. Организация полного централизованного технического обслуживания (гарантийный ремонт, регламентные работы и т. п.) позволяет решить многие проблемы, связанные с техническим обслуживанием ЭВМ, накопить статистику отказов различных вычислительных устройств и блоков, повысить надежность ЭВМ в целом.

Выполнение работ ЦТО осуществляется на основании хозяйственных договоров. Примерная стоимость профилактического осмотра ЦТО различных ЭВМ приведена в табл. 9-1.

В дальнейшем стоимость услуг ЦТО будет уменьшаться, а их номенклатура увеличиваться.

Таблица 9-1

**Стоимость профилактического осмотра ЦТО**

Тип ЭВМ	Оптовая цена профилактического осмотра одной ЭВМ в зависимости от их количества в организации, руб.					Стоимость ремонта одной ЭВМ, руб.
	1	2	3	4	5	
«Минск-32»	5510	4660	3840	3630	3510	7730
М-220, М-222	7800	6290	5790	5550	5400	7730
«Урал-11», «Урал-14», «Урал-16»	12 100	10 100	9420	9090	8890	11 500
БЭСМ-6	18 200	15 100	14 100	—	—	22 000
ЕС-1020	13 300	10 700	9850	9400	9200	11 500
ЕС-1030	18 500	15 000	13 800	13 200	12 800	22 000
«Наири-2, С»	1360	880	710	600	580	1640

## 9-2. ПЛАНИРОВАНИЕ РАБОТ ПО ТЕХНИЧЕСКОМУ ОБСЛУЖИВАНИЮ

Профилактическое обслуживание ЭВМ построено на календарном принципе, при котором профилактические мероприятия проводятся через определенные сроки и под-

разделяются на ежедневные, еженедельные или двухнедельные, десятидневные (декадные), месячные, квартальные и годовые.

ЭВМ не подвергается капитальному ремонту ввиду того, что их моральное старение наступает скорее, чем физический износ.

Ежедневные профилактические мероприятия включают:

- 1) внешний осмотр;
- 2) чистку от бумажной пыли;
- 3) смазку механических узлов устройств ввода-вывода;
- 4) функциональный контроль машины.

При проведении еженедельных, двухнедельных, декадных и месячных профилактик выполняются следующие работы:

- 1) устранение замеченных неисправностей, которые не потребовали остановки и ремонта ЭВМ;
- 2) профилактический контроль блоков питания;
- 3) чистка, смазка, проверка и наладка внешних устройств ЭВМ;
- 4) профилактический контроль машины;
- 5) функциональный контроль ЭВМ.

В состав квартальных профилактических работ, кроме вышеуказанных, входят проверка отремонтированных блоков, проверка и регулировка схем защиты.

При годовом профилактическом обслуживании проводится полный объем мероприятий по профилактике, предусмотренный инструкцией по эксплуатации ЭВМ. Кроме того, добавляется проверка монтажа и кабельного хозяйства, чистка корпусов стоек и шкафов.

Основой правильной эксплуатации ЭВМ является обязательное соблюдение месячного графика выполнения профилактических работ, позволяющего регулировать время проведения ежедневных осмотров и сроки выполнения еженедельных профилактик, в нем же окончательно уточняются сроки квартальных (годовых) регламентных профилактических работ, намечаемых на данный месяц.

Месячный график выполнения профилактических работ составляется на основе годового плана согласно установленным инструкцией по эксплуатации нормам или в результате анализа статистических данных, полученных в процессе эксплуатации ЭВМ. Пример составления такого графика на текущий месяц приводится в табл. 9-2.

Для проведения профилактического обслуживания однотипных технических устройств, входящих в состав ЭВМ,

Таблица 9-2

Месячный график профилактических работ

Номер ЭВМ	Числа месяца																															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
	С	Н	С	С	С	—	Н	С	С	С	С	—	—	С	С	Н	С	С	—	—	С	С	Н	С	С	—	—	М	С	С		
	С	С	М	С	С	—	—	С	С	С	Н	—	—	С	С	Н	С	С	—	—	С	С	С	С	С	Н	—	—	С	С	С	
	С	С	С	Н	С	—	—	С	С	С	С	Н	—	—	С	С	С	С	—	—	Г	Г	Г	Г	Г	Г	Г	—	—	С	С	С

Начальник службы технического обслуживания \_\_\_\_\_

Диспетчерская служба \_\_\_\_\_

Примечание. Условные обозначения: С — сменная профилактика, Н — еженедельная профилактика, М — ежемесячная профилактика, Г — годовая профилактика.  
Продолжительность устанавливается в зависимости от комплектации ЭВМ и уточняется на диспетчерских совещаниях.

например магнитных дисков, составляется скользящий график профилактических работ. С целью обеспечения непрерывной работоспособности ЭВМ вместо устройств, выводимых на профилактику, подключаются резервные. В табл. 9-3 приведен пример недельного графика для четырех магнитных дисков, один из которых является резервным.

Таблица 9-3

**Недельный график профилактического обслуживания накопителей на магнитных дисках**

Устройство	Заводской номер	День недели						
		1	2	3	4	5	6	7
5056		ПР						
5056				ПР				
5056						ПР		
5056			Р		Р		Р	ПР

Примечание. ПР — профилактика, Р — резерв.

При наличии на вычислительном центре большого количества ЭВМ и других технических средств для составления графика профилактических работ часто используются методы сетевого планирования и управления.

Кроме плановых профилактических регламентных работ техническое обслуживание ЭВМ может иметь и внеплановый характер. К внеплановому профилактическому обслуживанию относятся внеочередные профилактики, назначаемые главным образом после устранения серьезных неисправностей ЭВМ. Объем профилактических мероприятий определяется характером возникшей неисправности и ее возможными последствиями. Вывод машины на внеплановую профилактику можно также производить, когда количество сбоев, возникающих за определенный установленный период времени, превышает допустимое значение.

В заключение заметим, что одним из главных путей повышения эффективности эксплуатации ЭВМ может служить оптимизация сроков проведения профилактических работ.

В процессе эксплуатации ЭВМ после уточнения конкретных эксплуатационных характеристик и определения основного режима ее работы имеется возможность провести

корректировку периодичности профилактических мероприятий. С этой целью можно воспользоваться методиками, разработанными на базе теории вероятности, статистических решений и массового обслуживания.

### 9-3. КОЛИЧЕСТВЕННЫЕ ОЦЕНКИ ПРОФИЛАКТИЧЕСКИХ МЕРОПРИЯТИЙ

Одной из основных характеристик профилактического обслуживания является длительность профилактики ЭВМ:

$$t_{\text{проф}} = \sum_{i=1}^k t_{\text{ни}} + \sum_{j=1}^n t_{\text{в}j} + t_{\text{ф.к}},$$

где  $\sum_{i=1}^k t_{\text{ни}}$  — суммарное время проведения  $k$  профилактических мероприятий, выполняемых последовательно;  $\sum_{j=1}^n t_{\text{в}j}$  — время восстановления  $n$  неисправностей за время профилактики;  $t_{\text{ф.к}}$  — время функционального контроля.

Следует отметить, что на длительность профилактики в большой мере влияет степень квалификации обслуживающего персонала. За счет повышения уровня технической подготовки, обобщения и внедрения передового опыта по техническому обслуживанию можно значительно сократить суммарное время отказов и общий расход времени на проведение профилактических работ.

В табл. 9-4 приведены значения основных показателей работы ЭВМ ЕС-1030 за 3 года эксплуатации. Как видно из таблицы, на третьем году эксплуатации за счет накопления опыта и повышения качества обслуживания удалось значительно снизить общее время профилактики при повышении всех остальных показателей работы машины. Из табл. 9-5 можно видеть, как изменялось по годам суммарное время отказов отдельных устройств этой ЭВМ. Некоторое увеличение времени отказов периферийных устройств ЕС-1030 на третьем году эксплуатации свидетельствует о быстрой изнашиваемости этих устройств и их малой надежности.

Анализ статистических данных по эксплуатации конкретной ЭВМ позволяет дать рекомендации по замене профилактик меньшей периодичности на профилактики большей периодичности (например, ежедневные — на ежене-

дельные). Это позволяет увеличить время использования ЭВМ непосредственно на вычислительные работы.

Степень влияния профилактик на надежность характеристики ЭВМ можно выразить через приращение наработки на отказ

$$\Delta T_o = T_{o. пр} - T_o,$$

где  $T_{o. пр}$  — наработка на отказ ЭВМ при профилактическом обслуживании;  $T_o$  — наработка на отказ непрофилируемой ЭВМ.

Т а б л и ц а 9-4  
Основные показатели работы ЭВМ ЕС-1030

Показатели работы ЭВМ	Значение показателей по годам эксплуатации, ч		
	первый	второй	третий
Время во включенном состоянии	3974	6865	7251
Время полезное	8220	5560	6223
Профилактика	363	602	582
Сбои и неисправности	390	500	353
Нарботка на отказ	21,8	45,7	97,3
Среднее время восстановления	2,8	1,6	1,0
Среднесуточное полезное время	8,5	15,2	17,1
Среднесуточная загрузка	11,0	18,9	19,8
Среднее время бесперебойной работы	4,8	7,8	9,7
КПД без учета профилактики (%)	92,2	92,8	94,6

Т а б л и ц а 9-5  
Отказы устройств ЭВМ ЕС-1030

Наименование устройства	Тип устройства	Время отказов по годам эксплуатации, %		
		первый	второй	третий
Процессор	ЕС-2030	11,7	8,1	6,2
Каналы	ЕС-4430	11,4	10,8	9,8
Оперативная память	ЕС-3203	7,5	2,0	2,0
Накопитель на диске	ЕС-5052	17,0	20,2	33,8
Накопитель на ленте	ЕС-5010	9,5	7,2	12,2
Устройства управления	ЕС-5551	3,1	3,0	6,1
	ЕС-5511,			
АЦПУ	ЕС-7030	9,6	8,3	5,2
Устройство ввода	ЕС-6012	10,4	11,4	7,8
Пульт оператора	ЕС-7077	13,3	17,2	6,1
Выходной перфоратор	ЕС-7010	5,5	11,3	10,5
Стойки питания	—	1,0	0,5	—

Относительное значение приращения наработки

$$\delta T_o = \frac{\Delta T_o}{T_o} = \frac{T_{o, пр}}{T_o} - 1.$$

Отношение

$$\omega = \frac{T_{o, пр}}{T_o}$$

является эффективностью профилактического обслуживания.

Другой важной количественной характеристикой является коэффициент эффективности профилактики  $k_{\text{проф}}$ , который характеризует степень повышения безотказности ЭВМ за счет предотвращения отказов в момент профилактики.

Введем обозначения:  $n_{\text{проф}}$  — количество отказов, выявленных во время профилактики;  $n_o$  — число отказов, возникших в процессе работы ЭВМ;  $n_{\text{общ}} = n_o + n_{\text{проф}}$  — общее число отказов ЭВМ за период эксплуатации.

Тогда

$$k_{\text{проф}} = \frac{n_{\text{проф}}}{n_{\text{общ}}},$$

а в предположении, что потоки отказов в ЭВМ являются простейшими,

$$\omega = \frac{n_{\text{общ}}}{n_o},$$

или

$$\omega = \frac{1}{1 - k_{\text{проф}}}.$$

Таким образом, приведенные выражения позволяют дать количественную оценку степени повышения надежности машины за счет профилактического обслуживания.

## ГЛАВА ДЕСЯТАЯ

### АВТОМАТИЗАЦИЯ ПРОФИЛАКТИЧЕСКИХ РАБОТ

#### 10-1. СПОСОБЫ И МЕТОДЫ ПРОФИЛАКТИЧЕСКОГО КОНТРОЛЯ

Основное содержание профилактического контроля ЭВМ заключается в обнаружении и замене элементов машины, параметры которых приближаются к предельно допустимым (граничным) значениям.

Изменения параметров элементов ЭВМ возникают в результате медленных физико-химических изменений в структуре деталей и блоков, их старения и изменения свойств под влиянием различных внешних воздействий. По мере приближения параметров деталей и схем к граничным значениям даже незначительные колебания внешних условий (питающего напряжения, температуры, влажности и т. п.) могут вызвать появление случайных сбоев машины. Такие элементы являются потенциальными носителями неисправностей (см. гл. 2).

Нахождение и замена их производится при профилактическом контроле ЭВМ. Обычно контроль физико-химической структуры элементов производят путем измерения какого-либо выходного параметра схемы (или блока, или устройства машины), который в любой момент времени характеризует изменение этой структуры.

Определение состояния элементов по изменениям выходного параметра можно проводить как в нормальном эксплуатационном, так и в специальном профилактическом режимах. Более широкое применение нашел метод создания специальных утяжеленных режимов в процессе профилактического контроля. Создание таких режимов может быть осуществлено изменением питающих напряжений, частоты главных синхроимпульсов машины, температурных режимов, созданием дополнительных вибраций и т. д.

Различают *статические* и *динамические* профилактические режимы. При статическом режиме контрольные значения напряжений и частоты синхроимпульсов остаются постоянными в течение всего цикла профилактического контроля, а при динамическом режиме предусматривается периодическое их изменение. Таким образом, за счет создания утяжеленных режимов работы ЭВМ можно выявить критичные по надежности элементы.

Профилактический контроль осуществляется аппаратурным и программным путями. Аппаратурный контроль проводится с помощью специальной аппаратуры, контрольно-измерительных приборов и стендов. Все элементы ЭВМ разбиваются на профилактические участки с контрольными точками на каждом из них. В контрольных точках измеряются потенциалы, наблюдаются рабочие сигналы при помощи осциллографа для сравнения с эталонными.

При проведении профилактического контроля программным путем вся машина также разбивается на автономные участки таким образом, чтобы в процессе работы специаль-



ной организующей программы можно было, включив эти участки последовательно один за другим, быстро и удобно обнаружить неисправные элементы.

Наиболее эффективным следует считать программно-аппаратурный контроль при наличии в ЭВМ развитой системы аппаратурного контроля. Именно такой контроль используется во всех машинах третьего поколения. Подробно вопросы организации профилактического контроля машин ЕС ЭВМ будут рассмотрены в следующем параграфе.

Основные типы неисправностей ЭВМ, их проявление, способы обнаружения и контроля сведены в табл. 10-1. Как видно из таблицы, большая часть неисправностей обнаруживается с помощью аппаратных и программных средств. При этом машина останавливается или начинает часто выдавать сообщения об отклонениях от правильной работы. Кроме того, неисправности можно установить по поведению операционной системы (не вызываются задачи на решение, часто выбираются задачи из решения и т. п.). Эти неисправности могут быть вызваны ошибками в программном обеспечении или ошибочными действиями инженера-оператора. Очень часто операторы-новички ошибаются при наборе режимов решения задач, в задании директив для операционной системы. Поэтому при остановке машины надо прежде всего проверить, правильно ли заданы режимы на пультах. Предыдущие действия оператора можно установить по протоколу на операторском мониторе. Пока машина остановлена, до сброса ее на нуль, необходимо собрать и зафиксировать как можно больше информации, полезной для последующего анализа. При этом не нужно торопиться делать общий сброс, так как при сбоях можно иногда произвести исправление испорченной информации и продолжить вычисления. При установлении неаппаратного характера неисправности следует принять меры по устранению программной ошибки и продолжить решение задачи. Если же характер неисправности не ясен, то необходимо провести мероприятия по защите информации на внешних запоминающих устройствах от порчи при проведении последующих проверок.

Таким образом, работы по устранению неисправностей при профилактическом контроле можно разбить на следующие этапы:

- анализ характера неисправностей по текущему состоянию ЭВМ;

- контроль параметров окружающей среды и меры по устранению их отклонений;

- локализация ошибки и определение места неисправности с помощью аппаратных и программных средств ЭВМ без осциллографа,

  - то же с осциллографом;

  - устранение неисправности;

  - возобновление решения задачи.

Основные типы неисправностей ЭВМ, способы обнаружения и контроля

№ п/п	Типы неисправностей ЭВМ	Способы обнаружения	Способы контроля
1	Монтажные ошибки, нарушающие функционирование	Аппаратурные и программные средства	Прозвоночные стенды
2	Монтажные неисправности, нарушающие функционирование	Аппаратурные и программные средства	Прозвоночные стенды
3	Неустойчивые монтажные неисправности и контактные явления, вызывающие сбои	Аппаратурные и программные средства, простукивание	Прозвоночные стенды с виброплатформой, простукивание
4	Монтажные ошибки и неисправности (неправильное согласование цепей), сужающие область устойчивой работы ЭВМ	Аппаратурные и программные средства на тяжелых профилактических режимах. Параметрический контроль сигналов в цепях ЭВМ	Функциональные стенды с тяжелыми профилактическими режимами. Параметрический контроль сигналов. Прозвоночные стенды с измерением согласующих регистров

№ п/п	Типы неисправностей ЭВМ	Способы обнаружения	Способы контроля
5	Функциональные неисправности микросхем	Аппаратурные и программные средства	Функциональные стенды
6	Параметрические неисправности микросхем (недопустимые отклонения выходных уровней и задержек сигналов, колебания чувствительности по входу), сужающие область устойчивой работы ЭВМ	Аппаратурные и программные средства на тяжелых профилактических режимах. Параметрический контроль сигналов в цепях ЭВМ	Функциональные стенды с тяжелыми профилактическими режимами. Параметрический контроль сигналов
7	Неустойчивые неисправности микросхем, вызывающие сбои	Аппаратурные и программные средства, простукивание	Функциональные стенды с виброплатформой, простукивание

Если неисправность неустойчива и проявляется редко, то необходимо предусмотреть накопление информации и локализовать неисправность с помощью аппаратурных и программных средств ЭВМ.

Для успешного проведения профилактического контроля большое значение имеет использование дополнительных контрольно-измерительных приборов (тестеров, раз-

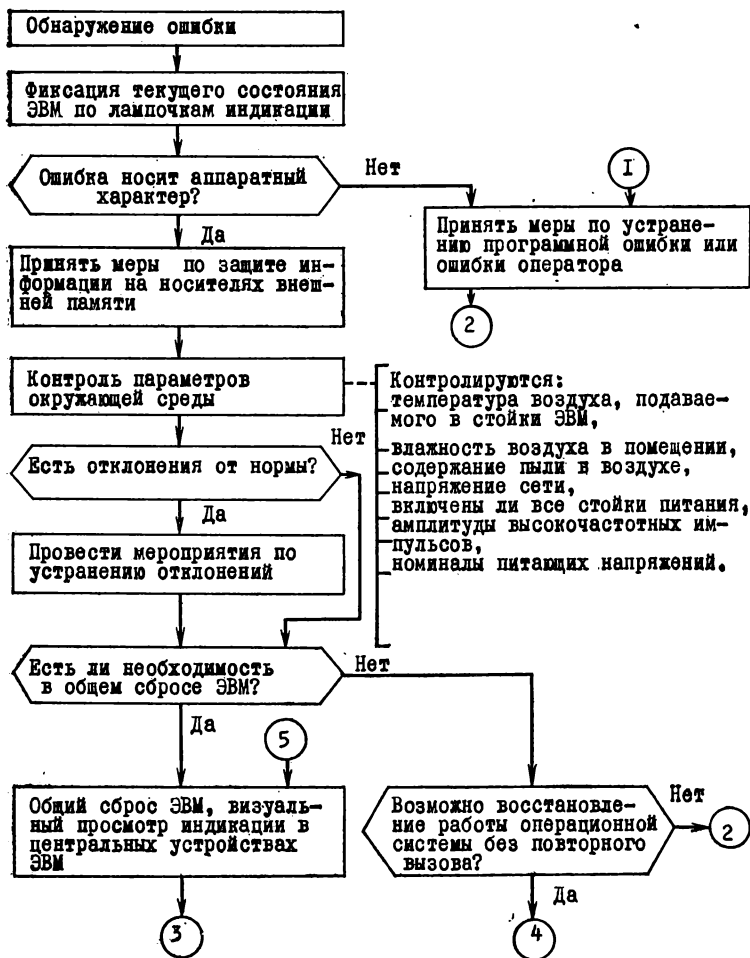


Рис. 10-1. Блок-схема реализации методики проведения профилактического контроля ЭВМ.

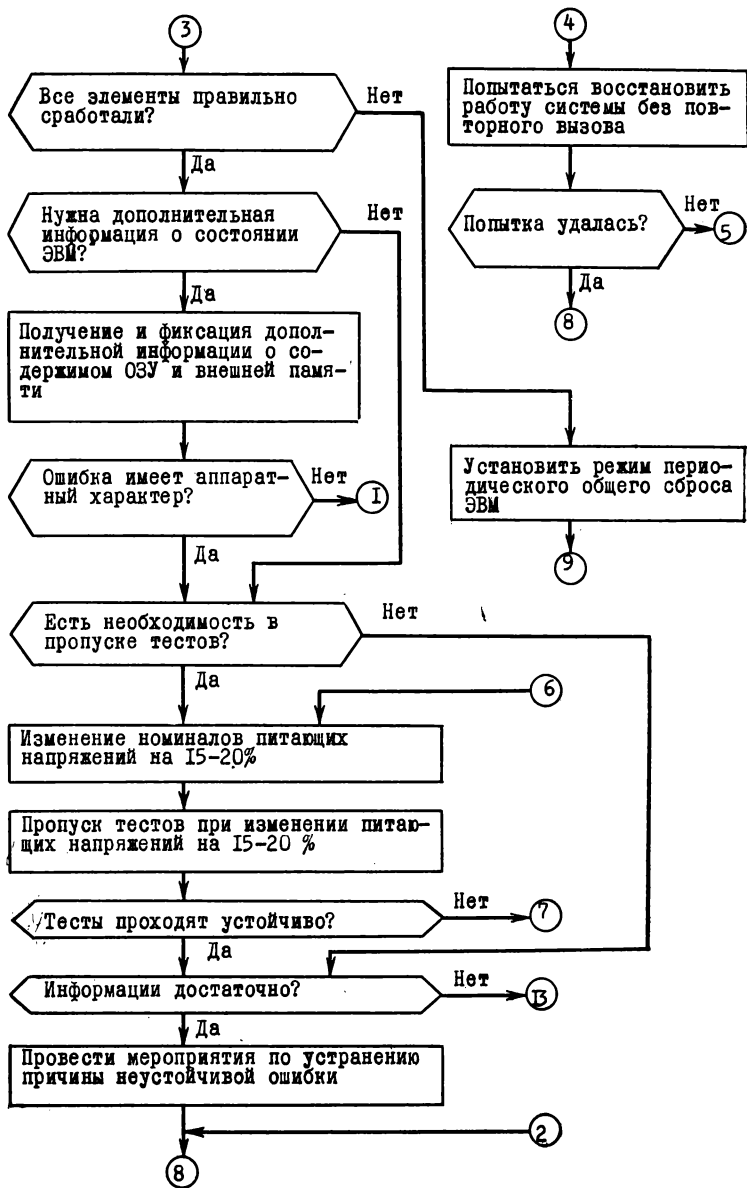


Рис. 10-1, (Продолжение).

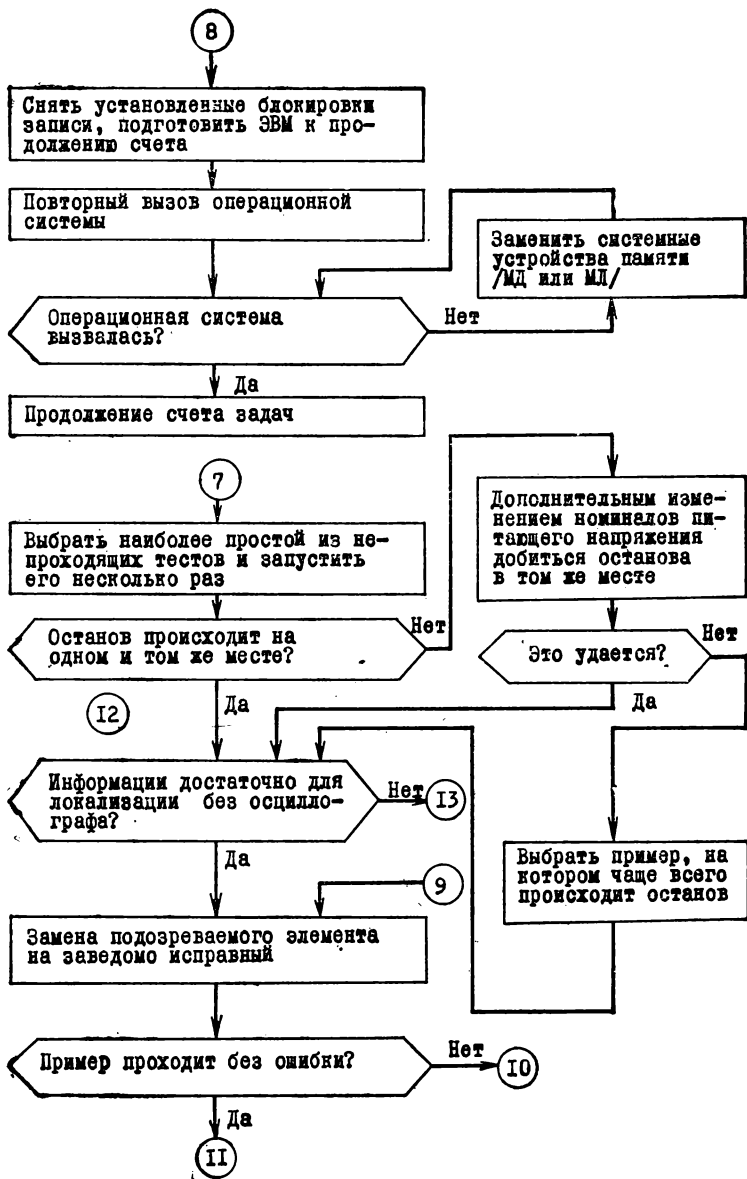


Рис. 10-1. (Продолжение).

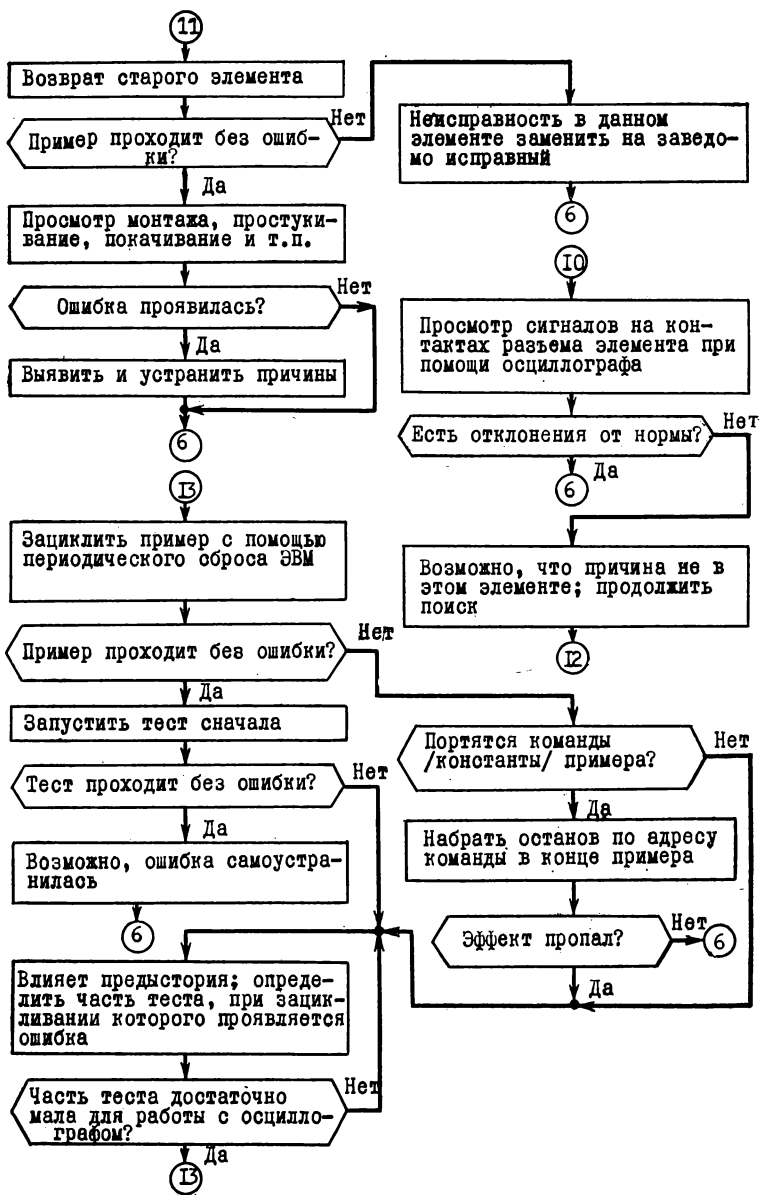


Рис. 10-1, (Продолжение).

личного рода самописцев, частотомеров, осциллографов). Осциллограф является пока главным и универсальным вспомогательным средством при проведении профилактического контроля и ремонта.

Для лучшего методического усвоения и последующей автоматизации профилактических работ целесообразно представить методику проведения профилактического контроля ЭВМ в виде блок-схемы аналогично тому, как это делается при описании вычислительных алгоритмов. Такая блок-схема дана на рис. 10-1.

## **10-2. ОРГАНИЗАЦИЯ ПРОФИЛАКТИЧЕСКОГО КОНТРОЛЯ В ЕС ЭВМ**

В вычислительных машинах третьего поколения предусмотрены специальные системы автоматического контроля и диагностики, позволяющие организовать оперативный контроль работоспособности ЭВМ и автоматический поиск (диагностику) места неисправности. Эти системы представляют собой совокупность аппаратурных и программных средств. Они позволяют контролировать как отдельные простейшие операции (прием и передача информации, сдвиг, сложение, дешифрирование и т. п.), так и наборы различных команд. Модели ЕС ЭВМ снабжены средствами аппаратурного контроля, позволяющими непосредственно и косвенно контролировать 90—95 % оборудования процессоров.

Программный контроль осуществляется с помощью специальных программ-тестов, входящих в состав комплекса программ технического обслуживания. Тесты предназначены для проверки правильности функционирования как отдельных устройств, так и машины в целом при их наладке и контроле в процессе профилактики. Они дают возможность гибко реагировать на все обнаруживаемые ошибки и сигналы прерываний от системы аппаратного контроля, управлять поведением автоматической диагностики, накапливать и анализировать информацию о состоянии машины.

В целом система обслуживания ЕС ЭВМ имеет в своем составе в качестве аппаратных средств:

- пульт управления и сигнализации;
- аппаратуру для автономной проверки устройств машины;
- систему автоматического контроля;
- систему автоматической диагностики;



аппаратуру для проведения профилактических и граничных испытаний устройств машины;  
контрольно-испытательную аппаратуру.

Автоматические системы контроля и диагностики дают возможность получить и проанализировать дополнительную, избыточную информацию о работе машины, без которой невозможен контроль ее работоспособности.

В устройствах машин ЕС используются различные способы получения избыточной информации. Она получается либо повторным запросом или преобразованием, либо дублированием аппаратуры, либо применением специальных контрольных кодов. Во всех моделях ЕС ЭВМ принят контроль по модулю 2. Он используется в процессорах, каналах, внешних запоминающих устройствах, а также в устройствах ввода и вывода.

Для сравнительного анализа избыточной информации используется схема сравнения, формирования и преобразования контрольных кодов. Их включают в различные точки устройств ЭВМ таким образом, чтобы обеспечивался контроль всех передач и преобразования информации. Например, в процессоре ЕС-2050 аппаратурный контроль охватывает:

- связи с основной и регистровой оперативными памятьми;
- цепи передачи информации между блоками и регистрами в блоках;

- блоки арифметического сумматора, цифровой десятичной арифметики и модификации адреса;

- счетчики блоков центрального управления и арифметическо-логических устройств;

- дешифраторы в схемах управления.

Это позволяет оперативно контролировать выполнение всех элементарных операций: сдвига, счета, суммирования, передачи кодов, дешифрирования и выработки управляющих сигналов.

Всего в процессоре и основной памяти имеется 66 точек контроля.

В менее мощной машине ЕС-1030 аппаратурным контролем охвачены все каналы передачи информации между блоками в процессоре, все схемы логических и арифметических операций, а также блоки управления процессором и каналами, оперативная и постоянная памяти. Всего в процессоре ЕС-2030 имеется десять точек контроля.

Подробно методы аппаратурного контроля ЭВМ рассмотрены в предыдущих главах книги.

Программные средства контроля включают в себя:

- процедуру ввода начальной программы;

- монитор контроля и диагностики;

- комплект тест-программ, не зависящих от диспетчера операционной системы:

- комплект тестов, управляемых операционной системой.

Большое значение при проведении профилактических и ремонтных работ имеет управляющая программа ТЕСТ-монитор, предназначенная для самостоятельного управления выполнением проверочных и диагностических тест-программ. Она позволяет без участия оператора произво-

дить стандартную обработку прерываний, загрузку тестов в оперативную память, прогон и заикливание отдельных тестов или примеров.

Различают три фазы работы программных средств контроля ЕС ЭВМ: рабочую, профилактическую и ремонтную.

В *рабочей фазе* под управлением операционной системы осуществляется периодическая тестовая проверка процессора, каналов и внешних устройств ЭВМ параллельно с решением задач. Здесь же предусмотрено автоматическое накопление статистики по сбоям и ее анализ.

*Фаза ремонта* разбивается на три этапа. На первом этапе производится диагностика отказавшего устройства и обнаружение места неисправности, на втором — собственно ремонт, а на третьем завершающем этапе — после-ремонтная проверка устройства проверочным тестом.

Место неисправности можно достаточно точно установить с помощью диагностических тестов и аппаратурного контроля. Для этого используется специальная команда **ДИАГНОСТИРОВАТЬ**, задающая принудительный режим проверки процессора и каналов. Информация о сбоях во внешних устройствах получается с помощью канальной команды **УТОЧНИТЬ СОСТОЯНИЕ**.

*Профилактическая фаза* заключается в плавном периодическом выполнении проверочных тестов как отдельных устройств, так и системы в целом. Такой контроль сопровождается автоматическим изменением напряжений источников питания и частоты синхронизации. При обнаружении неисправностей осуществляется ремонт отказавшего устройства.

Рассмотрим более подробно профилактическое обслуживание основных устройств машины ЕС-1022.

Инструкцией по эксплуатации предусмотрены следующие виды профилактического обслуживания: суточное, месячное и полугодовое.

При суточной профилактике процессора ЕС-2622 (на нее отводится 0,5 ч) производится тщательный осмотр, очистка и протирание внешних частей от пыли, проверка напряжений питания и проверка работоспособности процессора с помощью микропрограммных диагностических тестов.

При проведении месячной профилактики (в течение 4 ч), помимо суточного технического обслуживания, должен проводиться прогон диагностических тестов процессора и управляющей программы **ТЕСТ-монитор**, причем все тесты должны выполняться без сбоев.

Полугодовая профилактика (ее регламент 12 ч) включает в себя все проверки, проводимые при месячной профилактике, а также про-

верку основных параметров процессора и измерение мощности, потребляемой процессором.

Программные средства профилактического контроля процессора ЕС-2622 содержат следующий набор диагностических микропрограммных тестов:

проверка работоспособности ПЗУ при изменении питающего напряжения 5 В на  $\pm 5\%$ ;

проверка работоспособности оперативной памяти и памяти ключей защиты;

тест диагностики локальной памяти;

базисный тест для проверки центрального процессора;

тест диагностики правильной работы каналов.

Работа этих тестов подробно рассмотрена в предыдущих главах книги.

Для проверки правильности работы технических средств ЕС-1022 используется набор программ ТЕСТ-монитор. В его состав входит:

базовый тест;

управляющая программа;

тестовые программы.

Базовый тест необходим для проверки той части оборудования, которая необходима для работы управляющей программы. Проверяется правильность выполнения команд, используемых управляющей программой, работоспособность пишущей машинки и накопителя на магнитной ленте, на котором установлена магнитная лента с записью тестовых программ (тест-секций).

Управляющая программа служит для управления работой тест-секций и обладает следующими возможностями:

настраивается на работу в зависимости от размера памяти конкретной ЭВМ и готовности устройств ввода-вывода;

управляет последовательностью, в которой диагностируются устройства (анализируются таблицы, определяющие конфигурацию системы);

производит загрузку тест-секций;

выполняет ряд операций, общих для всех тест-секций (печать сообщений, декодировка приказов оператора, преобразование кодов и т. п.).

Тест-секции позволяют произвести проверку всех устройств, входящих в состав ЭВМ ЕС-1022: накопителей на магнитной ленте и магнитных дисках, устройств управления этими накопителями, АЦПУ, устройств ввода-вывода, основной памяти, каналов и средств защиты памяти. Кроме того, они позволяют произвести проверку взаимозаменяемости накопителей на ленте и на дисках.

Аппаратурный контроль процессора ЭВМ ЕС-1022 осуществляется при помощи десяти контрольных точек, охватывающих все основные регистры и блоки процессора. Во всех точках контроль производится путем сложения по модулю 2 содержимого регистров и сравнения результата суммирования со значениями контрольных разрядов соответствующих регистров.

Рассмотрим теперь организацию профилактического обслуживания устройств управления накопителями на магнитной ленте ЕС-5517 и на магнитных дисках ЕС-5551М.

Виды профилактик ЕС-5517:

- 1) суточная (продолжительность 25 мин);
- 2) двухнедельная (2 ч);
- 3) месячная (4 ч);
- 4) полугодовая (8 ч).

Для устройства ЕС-5551М предусмотрена лишь полугодовая профилактика в течение 2 ч.

При суточной профилактике устройства ЕС-5517 производится внешний осмотр и очистка внешних частей от пыли, а также проверка напряжений блоков электропитания.

При недельной профилактике производится проверка работы накопителя, а при месячной — всех накопителей по тест-программе ЕС-5517 при изменении напряжения источников питания  $\pm 5$  В ступенчато в пределах 4,75—5,25 В.

Кроме того, проверяется система защиты электропитания от перегрузок.

Полугодовая профилактика включает в себя все пункты месячной профилактики, а также проверку временных параметров генераторов синхронизации записи и считывания, счетчика задержек и стабильных формирователей сигналов. Измерение временных интервалов производится в заданных инструкцией по эксплуатации контрольных точках с помощью электронно-счетного частотомера типа ЧЗ-12. Эти измерения проводят при изменении напряжений источников питания на 5 %.

При проведении полугодовых контрольно-профилактических работ по устройству ЕС-5551М выполняются практически все мероприятия, предусмотренные при полугодовой профилактике ЕС-5517.

Эксплуатация машин ЕС позволила выявить наиболее характерные причины отказов основных устройств и дать рекомендации по их устранению:

*Процессор и каналы* — обрывы монтажных соединений, исчезновение контактов и разъемных типовых элементов замены, некачественный монтаж в стойках. Устранение — полная чистка монтажа, регулярная промывка спиртом разъемов ТЭЗ.

*Основная память* — монтажные ошибки, плохая отладка блоков запоминающих матриц (БЗМ), отказы ТЭЗ в мас-

совых цепях. Устранение — чистка монтажа, наладка БЗМ, стендовая проверка ТЭЗ.

*Магнитные диски* — малый срок службы магнитных головок, отказы блока питания, отказы позиционера, нарушение совместимости по записи и считыванию. Устранение — тщательное поддержание герметичности, чистоты и влажности при полной экранировке от влияния электромагнитных излучений, тщательная регулировка и подбор деталей.

*Устройство ввода перфокарт* — низкая надежность ламп подсветки в блоке считывания, высокая чувствительность к качеству перфокарт. Устранение — тщательный контроль за качеством перфокарт, обязательный вывод информации на экран дисплея для контроля.

*Устройства вывода информации* ЕС-7030, ЕС-7010, ЕС-7022 — низкое качество красящей ленты, низкая надежность перфоратора. Устранение — использовать ленту только на нейлоновой основе, заменить перфоратор на другой тип.

## ГЛАВА ОДИННАДЦАТАЯ ТЕХНИКО-ЭКОНОМИЧЕСКАЯ ОЦЕНКА МЕРОПРИЯТИЙ ПО ТЕХНИЧЕСКОМУ ОБСЛУЖИВАНИЮ ЭВМ

### 11-1. ВЫБОР И ОБОСНОВАНИЕ КРИТЕРИЕВ ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ ЭВМ

Поддержание ЭВМ в работоспособном состоянии и эффективное их использование с минимальными экономическими затратами требуют рациональной организации эксплуатации всей вычислительной системы.

Необходимость глубокого экономического анализа использования ЭВМ определяется стремительным развитием вычислительной техники, сопровождающимся огромными расходами и динамичностью сдвигов в структуре затрат (удельный вес затрат на электронные компоненты, математическое обеспечение, запоминающие и периферийные устройства, терминалы, линии связи и т. д.).

Важнейшей технико-экономической характеристикой вычислительной системы является производительность ЭВМ. Производительность вычислительной машины отражает не

только технические характеристики центральной части ЭВМ (быстродействие процессора, устройств памяти, каналов ввода-вывода), но и характеристики внешних устройств, операционных систем и даже эффективность той или иной конфигурации вычислительной системы при решении различных задач.

Производительность вычислительных систем во многом зависит от технических принципов их реализации, т.е. принципов структурной организации и функционирования технических средств и операционной системы ЭВМ.

Большое значение для повышения экономической эффективности ЭВМ имеет и улучшение производительности труда программиста. Производительность труда программиста зависит не только от его квалификации и сложности задачи, но и от режима эксплуатации ЭВМ.

Режим эксплуатации в данном случае определяет принцип взаимодействия программиста с ЭВМ в процессе отладки программы.

Количественное выражение показателя производительности ЭВМ вызывает определенные трудности. Это обусловлено большим числом факторов, влияющих на общую производительность вычислительной системы, и их сложной взаимосвязью. При оценке производительности ЭВМ обычно производится качественный анализ влияния различных факторов на общую производительность системы.

В связи с трудностью оценки абсолютного показателя производительности удобнее пользоваться показателем относительной производительности данной  $i$ -й ЭВМ по отношению к базовой  $j$ -й модели:

$$I_{ij} = \frac{P_i}{P_j},$$

где  $P$  — производительность ЭВМ.

Различают номинальную, комплексную и совокупную производительность машины.

Под *номинальной производительностью*  $P_n$  понимают производительность процессора, измеряемую количеством операций в единицу времени для типовой смеси процедур (см. гл. 2).

*Комплексная производительность*  $P_k$  — производительность ЭВМ с учетом работы устройств ввода-вывода, запоминающих устройств и другого дополнительного оборудования.

Показатели совокупной производительности определяются следующим образом:

$$\begin{aligned} P_{с.н} &= P_n k_{т.н}; \\ P_{с.к} &= P_k k_{т.к}; \end{aligned}$$

где  $k_{т.н}$  — коэффициент технического использования ЭВМ, определяемый по формуле из гл. 2.

При расчете номинальной производительности остается без внимания часть параметров (объем оперативной памяти и внешних запоминающих устройств, скорость работы периферийных устройств и т. д.), которые оказывают существенное влияние на работу ЭВМ.

Номинальная производительность рассчитывается при идеальных условиях работы машины. В действительности такой производительности достигнуть не удастся, так как в однопрограммном режиме теряется время на простои при вводе-выводе информации, а в мультипрограммном режиме и режиме разделения времени имеются системные потери времени.

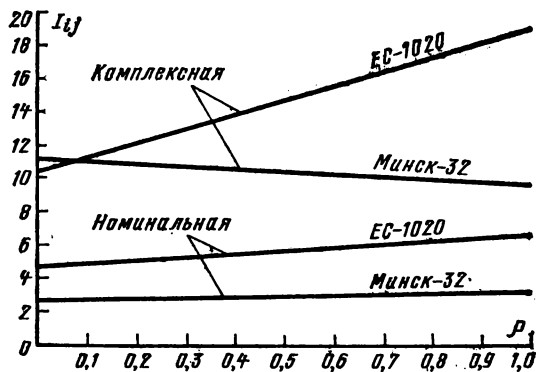


Рис. 11-1. К расчету индексов производительности ЭВМ.

На рис. 11-1 показано изменение индексов номинальной и комплексной производительности ЭВМ ЕС-1020 и «Минск-32» по отношению к ЭВМ «Минск-22» в зависимости от  $\rho$  — удельного веса экономических задач в общей загрузке машины. Из рисунка видно, что эффективность машины ЕС-1020 значительно выше машины второго поколения «Минск-32».

Сравнительные оценки индексов номинальной производительности для этих машин при решении научных и экономических задач приведены в табл. 11-1.

Таблица 11-1

Индекс номинальной производительности	Тип ЭВМ		
	«Минск-22»	«Минск-32»	ЕС-1020
При решении научных задач	1	2,6	4,6
При решении экономических задач	1	3,7	11,3
Усредненный	1	3,1	6,1

Подсчет комплексной производительности довольно сложен, так как многие параметры вычислительной машины и их влияние друг на друга не поддаются аналитическому описанию. В основном в  $P_k$  учитываются влияние объема памяти и параметры каналов. В табл. 11-2 приведены сравнительные оценки индексов комплексной производительности аналогично табл. 11-1.

Таблица 11-2

Индекс комплексной производительности	Тип ЭВМ		
	«Минск-22»	«Минск-32»	ЕС-1020
При решении научных задач	1	11,2	10,7
При решении экономических задач	1	9,9	19,0
Усредненный	1	10,1	15,9

При подсчете индексов в табл. 11-2 учитывались различия в типовом наборе внешних запоминающих устройств ЭВМ ЕС-1020 для решения научных и экономических задач. В частности, уменьшение индекса  $P_k$  машины ЕС-1020 при решении научных задач по сравнению с ЭВМ «Минск-32» вызвано малым количеством накопителей на магнитных дисках в типовой структуре ЭВМ ЕС-1020.

Сами показатели производительности ЭВМ не могут служить критериями ее эффективности. Экономически более эффективны те ЭВМ, у которых было бы минимальным отношение стоимость/производительность. В табл. 11-3 приведены сравнительные оценки стоимости единицы производительности различных ЭВМ.

Повышение стоимости единицы  $P_k$  машины ЕС-1020 по сравнению с машиной «Минск-32» вызвано высокой стоимостью внешних устройств машин ЕС. Для более полного учета эффективности каждого устройства, входящего в состав ЭВМ, следует использовать коэффициенты «стоимость/загрузка» и дебаланса, которые рассмотрены ниже.

Таблица 11-3

Стоимость единицы производительности	Тип ЭВМ		
	«Минск-22»	«Минск-32»	ЕС-1020
Номинальная	1	0,44	0,35
Комплексная	1	0,13	0,2

Для повышения производительности ЭВМ большое значение имеет рациональное решение задачи соответствия операций ввода и вывода информации и непосредственных вычислений по заданным алгоритмам.

Обычно для ЭВМ второго поколения при решении задач время, затрачиваемое на ввод и вывод информации, состав-



ляет 60—80 % общего времени решения одной задачи. При этом задачи других пользователей в течение этого времени не выполняются. Такая ситуация в процессе решения обусловлена характеристиками работы периферийных устройств и принципами реализации операций ввода и вывода для этих машин.

Для ЭВМ третьего поколения указанное несоответствие компенсируется путем совмещения во времени выполнения операций ввода и вывода для многих периферийных устройств с работой центрального процессора.

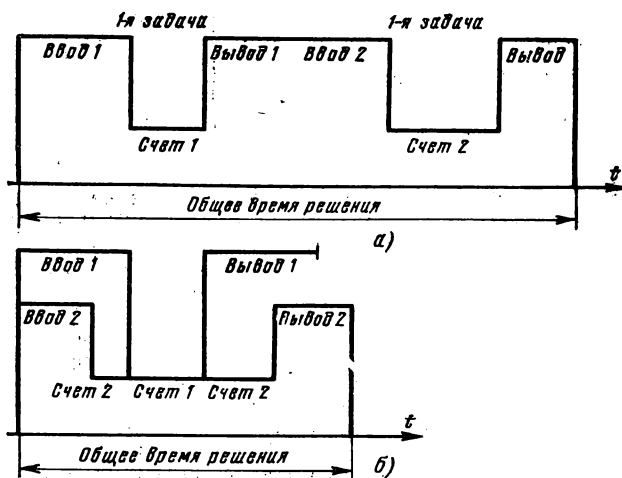


Рис. 11-2. Временные диаграммы работы ЭВМ.

Кроме того, для повышения производительности ЕС ЭВМ используется принцип мультипрограммирования, когда в оперативной памяти ЭВМ одновременно находится несколько программ и процессор переключается в ходе работы с выполнения одной программы на выполнение другой. При этом достигается определенная независимость центральной части ЭВМ от периферийных устройств, что обеспечивает гораздо более высокую загруженность процессора задачами пользователей. На рис. 11-2, а, б, показаны соответственно временные диаграммы работы ЭВМ в однопрограммном и мультипрограммном режимах при выполнении двух программ. Как видно из рисунка, режим мультипрограммирования дает значительное сокращение общего времени решения обеих задач.

Вообще говоря, этот режим характеризуется тем, что правила перехода от программы к программе устанавливаются так, чтобы избежать простоя наиболее дорогостоящих устройств ЭВМ, в частности процессора. Производительность системы зависит во многом от состава группы (пакета) программ, подлежащих совместному исполнению. Следовательно, встает задача подбора программ в пакете таким образом, чтобы оптимизировать степень занятости наиболее дорогих элементов ЭВМ. При этом качество обслуживания пользователя в этом режиме непосредственно не предусматривается. Пакеты задач вводятся в работу в определенные моменты времени. Довольно типичным является вариант с тремя пакетами в смену. Для получения результата счета пользователь должен ожидать завершения всего пакета. Поэтому наименее эффективным этот режим оказывается при отладке программ. В этом случае для внесения малейшего изменения в программу необходимо ожидать ввода в машину следующего пакета, что резко снижает производительность труда программиста.

Для решения этой задачи машины ЕС ЭВМ могут работать в других режимах, а именно в режиме параллельной обработки программ и с разделением времени.

Разделение времени представляет собой наиболее развитую форму многопрограммной работы. Это программный режим, при котором пользователь имеет возможность непосредственного доступа к машине, а время ответа по возможности близко к тому, которое было бы при единичном использовании ЭВМ.

В этом режиме, как и при параллельной обработке, имеющиеся в пакете программы выполняются поочередно, в течение некоторого отрезка времени, отводимого каждой программе. Длительность этого отрезка, как правило, меняется в зависимости от конкретной программы, что достигается с помощью специальной планирующей программы, входящей в состав операционной системы ЭВМ. Эта программа использует систему приоритетов, разрабатываемую таким образом, чтобы почти мгновенно реагировать на программы, связанные с коротким временем обслуживания, не слишком задерживая при этом выполнение длинных программ и обеспечивая тем самым оптимизацию времени ответа. Отметим, что при этом режим с разделением времени не является оптимальным с позиций производительности вычислительной машины.

Для количественной оценки эффективности вычислительной машины используются коэффициенты анализа загрузки оборудования:

1. Коэффициент машинного времени оборудования

$$q_m = \frac{t_m}{t_c - t_{п.р.}}$$

где  $t_m$  — машинное время работы оборудования за смену, ч;  $t_c$  — продолжительность смены, ч;  $t_{п.р.}$  — время простоев, включая неплановые ремонты, ч.

Здесь

$$t_m = t_c - t_{п.п.р.}$$

где  $t_{п.п.р.}$  — время плановых профилактических ремонтов за смену.

2. Коэффициент сменности оборудования

$$q_c = \frac{t_{с.ч.}}{t_{м.с.ч.}}$$

где  $t_{с.ч.}$  — фактическое число отработанных машинных часов в сутки;  $t_{м.с.ч.}$  — максимально возможное число машинных часов при работе в одну смену.

3. Стоимостный коэффициент сменности

$$q = \frac{\sum_{i=1}^n C_i q_{ci}}{\sum_{i=1}^n C_i}$$

где  $q_{ci}$  — коэффициент сменности  $i$ -й группы оборудования;  $C_i$  — стоимость  $i$ -й группы оборудования;  $n$  — количество групп оборудования.

4. Коэффициент целосменного использования оборудования

$$q_{ц} = \frac{z_1 + z_2 + z_3}{3z}$$

где  $z_1, z_2, z_3$  — единицы оборудования, работавшего в 1-й, 2-й и 3-й сменах (при трехсменной работе ЭВМ);  $z$  — общее количество единиц оборудования.

5. Коэффициент продолжительности ремонтного обслуживания

$$K_p = 1 - \frac{\sum_{i=1}^n D_{п.р.i}}{\sum_{i=1}^n D_{ф.р.i}}$$

где  $D_{п.р.i}$ ,  $D_{ф.р.i}$  — плановая и фактическая продолжительность ремонта  $i$ -й группы оборудования;  $n$  — количество групп оборудования.

Чем больше  $K_p$  приближается к нулю, тем лучше организована работа по профилактическому обслуживанию ЭВМ.

Рассмотрение этих коэффициентов в комплексе позволяет обоснованно решить наиболее сложную задачу загрузки оборудования ЭВМ — установления оптимальных коэффициентов сменности.

#### 6. Коэффициент «стоимость-загрузка»

$$F = \sum_{i=1}^n p_i v_i,$$

где  $p_i$  — относительная стоимость  $i$ -й группы оборудования;  $v_i$  — коэффициент использования во времени этой группы;  $n$  — количество групп оборудования.

Этот коэффициент может изменяться в пределах от нуля при полностью неиспользуемой ЭВМ до единицы при полной ее загрузке.

#### 7. Коэффициент дебаланса

$$B = 2 \sqrt{\sum_{i=1}^n (F - v_i)^2 p_i},$$

где  $(F - v_i)^2$  — мера отклонения загрузки  $i$ -й группы оборудования от средневзвешенной загрузки  $F$ .

Нормирующий множитель 2 обеспечивает изменение  $B$  от нуля для полностью сбалансированной системы до единицы в случае максимального разбаланса.

Для полностью сбалансированной системы  $v_i = F$  для всех  $i$ .

Максимально разбалансированная вычислительная система, в которой

$$v_i = 1; \quad \sum_{i=1}^m p_i = 0,5; \quad 1 \leq i \leq m$$

и

$$v_i = 0; \quad \sum_{i=m+1}^n p_i = 0,5; \quad m+1 \leq i \leq n.$$

В этом случае

$$F = \sum_{i=1}^m v_i p_i + \sum_{i=m+1}^n v_i p_i = 0,5;$$

$$B = 2 \left[ (0,5 - 1)^2 \sum_{i=1}^m p_i + (0,5 - 0)^2 \sum_{i=m+1}^n p_i \right]^{1/2} = 1.$$

На рис. 11-3 приведена графическая интерпретация расчета коэффициента «стоимость-загрузка». Здесь 1 — процессор,  $p_1 = 41\%$ ; 2 — ОЗУ (500 К),  $p_2 = 23\%$ ; 3 — четыре дисководы,  $p_3 = 13\%$ ; 4 — восемь накопителей,  $p_4 = 12\%$ ; 5 — считыватель перфокарт,  $p_5 = 2\%$ ; 6 — АЦПУ,  $p_6 = 9\%$ .

Коэффициент  $F$  представляет собой площадь гистограммы. Для нашего примера  $F = 0,903$  и  $B = 0,0602$ , т. е. ЭВМ эффективно используется и достаточно хорошо сбалансирована.

Следует иметь в виду, что методы анализа вычислительной мощности ЭВМ, изложенные выше, позволяют рас-

считывать в основном производительность ядра ЭВМ, состоящего из процессора, оперативной памяти и каналов. В связи с этим возникает необходимость использования более обобщенных критериев, которые смогли бы давать интегральную характеристику способности ЭВМ обрабатывать информацию. Таким критерием может служить *системная производительность* (иногда используется другой термин — пропускная способность системы), оцениваемая по быстродействию ЭВМ при решении определенного класса задач. Таким образом, под системной производительностью понимается количество типовых задач, решаемых ЭВМ в единицу времени (при этом промежуток времени, на котором подсчитывается число решенных ЭВМ задач, должен выбираться достаточно большим, чтобы исключить его влияние на системную производительность из-за возможного незавершения отдельных задач в выбранный промежуток).

Необходимо отметить, что системная производительность существенно зависит от класса задач, решаемых на ЭВМ, и целью дальнейших исследований должно быть определение классов типовых задач и выработка определенного набора количественных оценок этих типовых задач для использования их в дальнейшем при оценке эффективности ЭВМ.

С другой стороны, при заданном классе решаемых задач системная производительность определяется конфигурацией системы, а не только производительностью центрального процессора. Такая ситуация имела место уже для ЭВМ второго поколения, в которых существенное влияние на скорость решения задач оказывали также такие параметры машины, как объем ОЗУ, скорость обмена с внешней памятью, скорость ввода и вывода. Но в наиболее полной мере влияние конфигурации ЭВМ на ее системную производительность проявляется для ЭВМ третьего поколения, в основе которых лежит модульный принцип построения и мультипрограммный режим работы. Естественно, что это влияние проявляется по-разному для

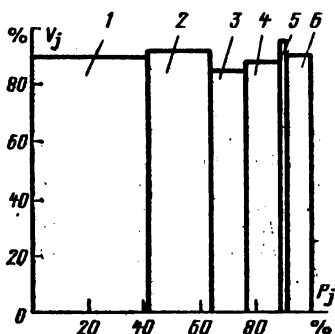


Рис. 11-3. К расчету коэффициентов «стоимость — загрузка».

задач разных классов, при этом определяющее значение для системной производительности ЭВМ на данном наборе задач может иметь та или иная группа технических средств.

Эффективность использования ЭВМ в значительной мере определяется номенклатурой ВУ и их параметрами. Производительность ВЗУ, УВВ существенно влияет на системную производительность ЭВМ при решении как информационно-логических, так и научно-технических задач.

Эффект от увеличения комплектации ЭВМ внешними устройствами проявляется в двух основных направлениях:

1) улучшении функционирования самой ЭВМ, т. е. более полная ее загрузка (увеличение количества типовых задач, решаемых ЭВМ в единицу времени) за счет сбалансирования пропускной способности отдельных технических подсистем, повышения их надежности и т. п.;

2) более полном удовлетворении требований пользователей относительно структуры ЭВМ (например, для реализации дистанционной обработки необходимы абонентские пункты и аппаратура передачи данных, для реализации в ЭВМ развитой системы банков данных требуется значительное увеличение состава внешних запоминающих устройств), характера решаемых задач (например, для автоматизации проектирования необходимы графопостроители), характеристик их решения (например, для уменьшения времени ответа на запрос в информационно-поисковых системах требуется увеличение количества АЦПУ и дисплеев), удобства общения человека с ЭВМ и т. д.

Пусть  $P_{\text{сист}}$  — количество типовых задач, решаемых ЭВМ в единицу времени (системная производительность), а  $C_{\text{сист}}$  — стоимость ЭВМ, состоящая из затрат на изготовление и эксплуатацию средств ЕС ЭВМ в данной комплектации.

При отвлечении от специфических требований пользователей для оценки комплектации ЭВМ наиболее целесообразным, как уже отмечалось выше, представляется критерий  $C_{\text{сист}}/P_{\text{сист}}$ , т. е. стоимость решения типовой задачи в ЭВМ.

Мультипрограммный режим работы ЭВМ предполагает наличие в оперативной памяти ЭВМ некоторого числа программ, готовых к обслуживанию процессором. Количество этих программ существенно влияет на эффективное использование процессора. Это связано с тем, что при решении задачи программы в большинстве случаев требуют ввода-вывода дополнительной информации с внешних носителей (НМД, НМЛ). В течение этого времени процессор может решать другую задачу, обслуживание которой в свою очередь может быть прервано необходимостью ввода-вывода дополнительной информации и т. д. Если готовой к обслуживанию задачи в оперативной памяти нет, процессор простаивает; очевидно, что с увеличением числа одновременно решаемых задач, т. е. с увеличением степени мультипрограммирования,

вероятность простоя процессора уменьшается (т. е. увеличивается его загруженность и, следовательно,  $P_{\text{сист}}$ ).

Обозначим:  $T_{\text{ЦП}}$  — загрузка центрального процессора (ЦП);  $n_1$  и  $n_d$  — количество соответственно НМЛ и НМД в рассматриваемой ЭВМ;  $m$  — степень мультипрограммирования (количество одновременно решаемых в ЭВМ задач); степень мультипрограммирования определяется ресурсами как оперативной, так и внешней памяти.

Пусть ЭВМ имеет объем  $64N$  Кбайт, ядро операционной системы вместе с областями для системного ввода-вывода занимает  $64p$  Кбайт и в мультипрограммном режиме решаются задачи, требующие в среднем  $64r$  Кбайт оперативной памяти. Тогда максимальный уровень мультипрограммирования, определяемый ресурсами ОП, равен:

$$\frac{64N - 64p}{64r} = \frac{N - p}{r}.$$

В первом приближении можно также считать, что для решения одной задачи требуется  $z$  НМД (емкостью 7,25 Мбайт) или НМЛ с учетом рабочих массивов, исходных данных, временных наборов данных и т. д. (предполагаем, что задачи решаются в основном на базе стандартных пакетов прикладных программ, размещенных на магнитных дисках или лентах); операционная система занимает два НМД. Тогда максимально возможная степень мультипрограммирования, определяемая ресурсами ВЗУ, будет равна:

$$\frac{n_d + n_n - 2}{z}.$$

Таким образом, можно считать, что

$$m = \min \left\{ \frac{N - p}{r}, \frac{n_d + n_n - 2}{z} \right\}.$$

В [5] получена зависимость  $T_{\text{ЦП}}$  от  $n_d$  и  $n_n$  в предположении, что весь процесс решения в ЭВМ  $m$  задач представим в виде замкнутой системы массового обслуживания.

Результаты расчета загрузки центрального процессора при  $N = 4$ ,  $p = r = z = 1$  и стандартных параметрах НМД и НМЛ в зависимости от комплектации ЭВМ накопителями на магнитных дисках ( $n_d$ ) и накопителями на магнитных лентах ( $n_n$ ) сведены в табл. 11-4.

Т а б л и ц а 11-4

Загрузка центрального процессора  $T_{\text{ЦП}}$  ( $n_d, n_n$ )

$n_d$	$n_n$			
	0	1	2	3
2	0	0,21	0,32	0,39
3	0,29	0,42	0,54	0,54
4	0,53	0,64	0,64	0,64
5	0,72	0,72	0,72	0,72

Анализ результатов показывает, что при одной и той же степени мультипрограммирования системная производительность ЭВМ существенно зависит от типа используемого ВЗУ. Так, при максимально возможной степени мультипрограммирования (исходя из ресурсов ОП  $m_{max} = 3$ ) загрузка ЦП при установке трех НМЛ и двух дисководов равна 0,39, а при установке пяти дисководов — 0,72.

## 11-2. ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ СИСТЕМЫ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ ЭВМ

Рассмотренные в предыдущих главах основные показатели эксплуатационных свойств ЭВМ, такие как коэффициент использования машины, коэффициент эффективности профилактики, показатели безотказности работы и восстанавливаемости, имеют существенные недостатки при использовании их для оценки экономической эффективности технического обслуживания ЭВМ, так как они не учитывают значений соответствующих им технико-экономических затрат.

Стоимость эксплуатации  $C_э$  оборудования ЭВМ складывается из нескольких статей затрат:

1.  $C_{з.п}$  — заработная плата основного производственного персонала (начальники машин, группы оборудования, смены, инженеры, механики, техники).

2.  $C_{д.з.п}$  — дополнительная заработная плата (обычно 8—10 % основной).

3.  $C_{с.с}$  — отчисления на социальное страхование (устанавливаются в размере 6,6 % к сумме основной и дополнительной затрат).

4.  $C_{ам}$  — амортизация основных фондов (вычисляется в размере 10 % стоимости вычислительных средств, периферийных и дополнительных устройств; 10,7 % стоимости вспомогательного оборудования; 18 % стоимости прочего оборудования и инвентаря). В стоимость оборудования иногда включают транспортировку, установку и наладку (10 % стоимости оборудования). Амортизация здания вычисляется как 0,25 % стоимости здания (иногда при определении амортизации учитывается средняя плата за помещение).

5.  $C_{э.э}$  — потребление электроэнергии (вычисляется по показателям потребляемой мощности каждого устройства, режима работы оборудования и отпускной цене электроэнергии).

6.  $C_{д.р}$  — расходы на ремонтные и запасные части (5 % стоимости оборудования).

7.  $C_{р.м}$  — расходы на эксплуатационные материалы (перфоленты, перфокарты, бумага для АЦПУ и т. д.).

8.  $C_{а.п}$  — содержание помещений и сооружений (водоснабжение, канализация, освещение, отопление и т. д.).

9.  $C_{д}$  — дополнительные расходы.



Целесообразно в качестве критерия технического обслуживания ЭВМ взять техническую стоимость часа вычислительного функционирования, полученную за установившийся период эксплуатации (год, квартал, месяц), причем этот период должен быть выбран много большим среднего времени между отказами ЭВМ.

Тогда стоимость часа вычислительного функционирования

$$C_{в. \phi} = \frac{C_{\phi} (1 + K_{п. \phi. п})}{T_{\phi} K_{н}}$$

где  $C_{\phi}$  — стоимость эксплуатации ЭВМ;  $T_{\phi}$  — установившийся период эксплуатации;  $K_{н}$  — коэффициент использования ЭВМ;  $K_{п. \phi. п}$  — коэффициент планово-экономического прироста.

Техническая стоимость часа запишется так

$$C_{т. ч} = \frac{C_{\phi} - (C_{з. п} + C_{д. з. п}) K_{н. р}}{T_{\phi} K_{н}}$$

где  $K_{н. р}$  — коэффициент накладных расходов.

Стоимость эксплуатационных расходов ЭВМ состоит из двух частей

$$C_{\phi} = C_{н. р} + C_{р. н}$$

где  $C_{н. р}$  — эксплуатационные расходы при исправной ЭВМ;  $C_{р. н}$  — эксплуатационные расходы при профилактике и ремонте.

При этом каждая часть эксплуатационных расходов состоит в свою очередь из двух составляющих — постоянной ( $C'_{н. р}$  и  $C_{р. н}$ ), не зависящей от функционального состояния машины, и переменной ( $C''_{н. р}$  и  $C'_{р. н}$ ), зависящей от ее работоспособности.

Используя введенные обозначения, получаем:

$$C'_{н. р} = C'_{р. н} = C_{з. м} + C_{з. в} + C_{з. п} + C_{р. м} + C_{с. с};$$

$$C''_{н. р} = (C_{з. п} + C_{д. з. п}) (1 + K_{н. р});$$

$$C'_{р. н} = (C_{з. п} + C_{д. з. п}) (1 + K_{н. р}) + C_{д} + C_{д. р}.$$

Введем дополнительные коэффициенты:

1. Коэффициент стоимости постоянных расходов за единицу времени

$$\alpha = \frac{C'_{н.р.}}{T_{\phi}}$$

2. Коэффициент затрат на зарплату в единицу времени

$$\gamma = \frac{C''_{н.р.}}{T_{\phi}}$$

3. Коэффициент стоимости запасных деталей и дополнительных расходов в единицу времени

$$\beta = \frac{C_d + C_{д.р.}}{T_{\phi}},$$

тогда  $C_3 = [\alpha + \gamma(1 + K_{н.р.}) + \beta] T_{\phi}$

и стоимость часа эксплуатации ЭВМ

$$C_4 = \frac{\alpha + \gamma(1 + K_{н.р.}) + \beta}{K_n},$$

а техническая стоимость часа

$$C_{т.ч} = \frac{\alpha + \beta + \gamma}{K_{и.}}$$

Использование двух последних выражений для оценки экономической эффективности эксплуатации вычислительной машины дает возможность учесть комплектность ЭВМ, затраты на зарплату, а также стоимость израсходованных ЗИП и дополнительных расходов.

П Р И Л О Ж Е Н И Е

Т а б л и ц а П-1

Перечень тест-секций ТЕСТ-монитора ЕС ЭВМ

Наименование тест-секции	Код	Понсковый номер
<i>Тесты процессора</i>		
Тест стандартного набора команд	F1700	170
	F1710	171
	F1720	172
	F1730	173
	F1740	174
	F1750	175
	F1760	176
	F1770	177
	Тест арифметики с плавающей запятой	F1800
F1810		181
F1820		182
Тест десятичной арифметики	F1900	190
Тест прерываний	E2000	200
	E2010	201
	E2020	202
	E2030	203
	E2040	204
	E2050	205
	E2060	206
Тест основной памяти	E3910	391
	E3920	392
Тест средств защиты памяти	F3900	390
Тест каналов	24300	430
	24310	431
	24320	432
	24330	433
	24340	434
	24350	435
	24360	436
	24370	437
	24380	438
	243A0	43A
	243B0	43B
243C0	43C	
Тест совместной работы каналов	E4C10	4C1
	E4C20	4C2
	E4C30	4C3
Тест прямого управления	F4E10	4E1
	E4E30	4E3

Наименование тест-секции	Код	Поисковый номер
<i>Тесты внешних запоминающих устройств</i>		
<b>Функциональный тест магнитной ленты</b>	F5010	501
	F5020	502
	F5030	503
	F5040	504
	F5050	505
	F5060	506
	F5070	507
	F5080	508
	F50F0	50F
	<b>Диагностика магнитной ленты</b>	F5190
F51A0		51A
F51B0		51B
<b>Двухканальный режим накопителей на магнитной ленте (НМЛ)</b>	F5290	529
	F52A0	52A
	F52B0	52B
	F52C0	52C
	F52D0	52D
<b>Тест устройства управления накопителем на магнитных дисках (НМД)</b>	F6000	600
	F6010	601
	F6020	602
	F6030	603
	F6040	604
	F6050	605
	F6060	606
	F6070	607
	F6080	608
	F60A0	60A
	F60B0	60B
	F60C0	60C
	F60D0	60D
	F60E0	60E
	F60F0	60F
<b>Тест накопителя на сменных магнитных дисках</b>	F6100	610
	F6110	611
	F6120	612
	F6140	614
<i>Тесты устройств ввода-вывода</i>		
<b>Тест устройства алфавитно-цифровой печати (АЦПУ)</b>	F8300	830
	F8370	837
	F8380	838
<b>Тест устройства ввода с перфокарт (УСК)</b>	F8C00	8C0
	F8C10	8C1
	F8C90	8C9

Продолжение табл. П-1

Наименование тест-секции	Код	Поисковый номер
Тест устройства вывода на перфокарты (УВК)	F8D40	8D4
	F8D50	8D5
	F8D90	8D9
Тест пишущей машинки (ПМ)	F9000	900
	F9010	901
	F9020	902
	F9090	909
	F9290	929
Тест устройства ввода с перфоленты (УСЛ)	F92A0	92A
	93A0	93A
Тест устройства вывода на перфоленту (УВЛ)	93B0	93B

Таблица П-2

Операторы языка АССЕМБЛЕР

Команда	Код операции (мнемонический)	Формат стандарта	Формат операндов (явный адрес)
СЛОЖЕНИЕ	AR	RR	R1, R2
СЛОЖЕНИЕ	A	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
СЛОЖЕНИЕ ПОЛУСЛОВА	AN	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
СЛОЖЕНИЕ ДЕСЯТИЧНОЕ	AP	SS	D1 (L1, B1), D2 (L2, B2) или D1 (, B1), D2 (, B2)
СЛОЖЕНИЕ КОДОВ	ALR	RR	R1, R2
СЛОЖЕНИЕ КОДОВ	AL	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ)	ADR	RR	R1, R2
СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ)	AD	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ)	AER	RR	R1, R2

Команда	Код операции (мнемонический)	Формат стандарта	Формат операндов (явный адрес)
СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ)	AE	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ)	AWR	RR	R1, R2
СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ)	AW	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ)	AUR	RR	R1, R2
СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ)	AU	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
И	NR	RR	R1, R2
И	N	RXX	R1, D2 (X2, B2) или R1, D2 (, B2)
И	NI	SI	D1 (B1), I2
И	NC	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (B2)
ПЕРЕХОД С ВОЗВРАТОМ	BALR	RR	R1, R2
ДЕЛЕНИЕ (ДЛИННОЕ)	DD	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ДЕЛЕНИЕ (КОРОТКОЕ)	DER	RR	R1, R2
ДЕЛЕНИЕ (КОРОТКОЕ)	DE	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ОТРЕДАКТИРОВАТЬ	ED	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (, B2)
ОТРЕДАКТИРОВАТЬ И ОТМЕТИТЬ	EDMK	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (B2)
ВЫПОЛНИТЬ	EX	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ИСКЛЮЧАЮЩЕЕ ИЛИ	XR	RR	R1, R2
ИСКЛЮЧАЮЩЕЕ ИЛИ	X	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ИСКЛЮЧАЮЩЕЕ ИЛИ	XI	SI	D1 (B1), I2

Команда	Код операции (мнемонический)	Формат стандарта	Формат операндов (явный адрес)
ИСКЛЮЧАЮЩЕЕ ИЛИ	XC	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (, B2)
ПОПОЛАМ (ДЛИННОЕ)	HDR	RR	R1, R2
ПОПОЛАМ (КОРОТКОЕ)	HER	RR	R1, R2
ОСТАНОВИТЬ ВВОД- ВЫВОД	HIO	SI	D1 (B1)
ПРОЧИТАТЬ СИМВОЛ	IC	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ПРОЧИТАТЬ КЛЮЧ ПАМЯТИ	ISK	RR	R1, R2
ЗАГРУЗКА	LR	RR	R1, R2
ЗАГРУЗКА	L	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ЗАГРУЗКА АДРЕСА	LA	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ЗАГРУЗКА ПОЛУ- СЛОВА	LH	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ЗАГРУЗКА (ДЛИННАЯ)	LDR	RR	R1, R2
ЗАГРУЗКА (КОРОТ- КАЯ)	LER	RR	R1, R2
ЗАГРУЗКА (ДЛИННАЯ)	LD	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ЗАГРУЗКА (КОРОТ- КАЯ)	LE	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ЗАГРУЗКА ГРУППО- ВАЯ	LM	RS	R1, R3, D2 (B2)
ЗАГРУЗКА И ПРО- ВЕРКА	LTR	RR	R1, R2
ЗАГРУЗКА И ПРО- ВЕРКА (ДЛИННАЯ)	LTDR	RR	R1, R2
ЗАГРУЗКА И ПРО- ВЕРКА (КОРОТКАЯ)	LTER	RR	R1, R2
ЗАГРУЗКА ДОПОЛНЕ- НИЯ	LCR	RR	R1, R2
ЗАГРУЗКА ДОПОЛНЕ- НИЯ (ДЛИННАЯ)	LCDR	RR	R1, R2
ЗАГРУЗКА ДОПОЛНЕ- НИЯ (КОРОТКАЯ)	LCER	RR	R1, R2

Команда	Код операции (мнемонический)	Формат стандарта	Формат операндов (явный адрес)
ЗАГРУЗКА ПОЛОЖИТЕЛЬНАЯ	LPR	RR	R1, R2
ЗАГРУЗКА ПОЛОЖИТЕЛЬНАЯ (ДЛИННАЯ)	LPDR	RR	R1, R2
ЗАГРУЗКА ПОЛОЖИТЕЛЬНАЯ (КОРОТКАЯ)	LPER	RR	R1, R2
ЗАГРУЗКА ОТРИЦАТЕЛЬНАЯ	LNR	RR	R1, R2
ЗАГРУЗКА ОТРИЦАТЕЛЬНАЯ (ДЛИННАЯ)	LNDR	RR	R1, R2
ЗАГРУЗКА ОТРИЦАТЕЛЬНАЯ (КОРОТКАЯ)	LNER	RR	R1, R2
ЗАГРУЗКА PSW	LPSW	SI	D1 (B1)
ПЕРЕСЫЛКА СИМВОЛА	MVI	SI	D1 (B1), I2
ПЕРЕСЫЛКА	MVC	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (B2)
ПЕРЕСЫЛКА ЦИФР	MVN	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (B2)
ПЕРЕСЫЛКА ЗОН	MVZ	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (B2)
ПЕРЕСЫЛКА СО СДВИГОМ	MVO	SS	D1 (L1, B1), D2 (L2, B2) или D1 (, B1), D2 (, B2)
УМНОЖЕНИЕ	MR	RR	R1, R2
УМНОЖЕНИЕ	M	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
УМНОЖЕНИЕ ДЕСЯТИЧНОЕ	MP	SS	D1 (L1, B1), D2 (L2, B2) или D1 (, B1), D2 (, B2)
УМНОЖЕНИЕ (ДЛИННОЕ)	MDR	RR	R1, R2
УМНОЖЕНИЕ (ДЛИННОЕ)	MD	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
УМНОЖЕНИЕ (КОРОТКОЕ)	MER	RR	R1, R2



Команда	Код операции (мнемонический)	Формат стан-дарт	Формат операндов (явный адрес)
УМНОЖЕНИЕ (КОРОТКОЕ)	ME	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ИЛИ	OR	RR	R1, R2
ИЛИ	O	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ИЛИ	OI	SI	D1 (B1), I2
ИЛИ	OC	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (, B2)
УПАКОВАТЬ	PACK	SS	D1 (L1, B1), D2 (L2, B2) или D1 (, B1), D2 (, B2)
ПРЯМОЕ ЧТЕНИЕ	RDD	SI	D1 (B1), I2
УСТАНОВИТЬ МАСКУ ПРОГРАММЫ	SPM	RR	R1
УСТАНОВИТЬ МАСКУ СИСТЕМЫ	SSM	SI	D1 (B1)
УСТАНОВИТЬ КЛЮЧ ПАМЯТИ	SSK	RR	R1, R2
НАЧАТЬ ВВОД-ВЫВОД	SIO	SI	D1 (B1)
СДВИГ ВЛЕВО	SLA	RS	R1, D2 (B2)
СДВИГ ВЛЕВО ДВОЙНОЙ	SLDA	RS	R1, D2 (B2)
СДВИГ ВЛЕВО КОДОВ	SLL	RS	R1, D2 (B2)
СДВИГ ВЛЕВО ДВОЙНОЙ КОДОВ	SLDL	RS	R1, D2 (B2)
СДВИГ ВПРАВО	SRA	RS	R1, D2 (B2)
СДВИГ ВПРАВО ДВОЙНОЙ	SRDA	RS	R1, D2 (B2)
БЕЗУСЛОВНЫЙ ПЕРЕХОД	BR	RR	R2
БЕЗУСЛОВНЫЙ ПЕРЕХОД	B	RX	D2 (X2, B2) или D2 (, B2)
НЕТ ОПЕРАЦИИ	NOPR	RR	R2
НЕТ ОПЕРАЦИИ	NOP	RX	D2 (X2, B2) или D2 (, B2)
СРАВНЕНИЕ	CR	RR	R1, R2
СРАВНЕНИЕ	C	RX	R1, D2 (X2, B2) или R1, D2 (, B2)

Команда	Код операции (мнемонический)	Формат стандарта	Формат операндов (явный адрес)
СРАВНЕНИЕ ПОЛУСЛОВА	CH	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
СРАВНЕНИЕ ДЕСЯТИЧНОЕ	CP	SS	D1 (L1, B1), D2 (L2, B2) или D1 (, B1), D2 (, B2)
СРАВНЕНИЕ КОДОВ	CLR	RR	R1, R2
СРАВНЕНИЕ КОДОВ	CL	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
СРАВНЕНИЕ КОДОВ	CLI	SI	D1 (B1) I2
СРАВНЕНИЕ КОДОВ	CLC	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (B2)
СРАВНЕНИЕ (ДЛИННОЕ)	CDR	RR	R1, R2
СРАВНЕНИЕ (ДЛИННОЕ)	CD	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
СРАВНЕНИЕ (КОРОТКОЕ)	CER	RR	R1, R2
СРАВНЕНИЕ (КОРОТКОЕ)	CE	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ПРЕОБРАЗОВАНИЕ В ДВОИЧНУЮ	CVB	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ПРЕОБРАЗОВАНИЕ В ДЕСЯТИЧНУЮ	CVD	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ДЕЛЕНИЕ	DR	RR	R1, R2
ДЕЛЕНИЕ	D	RX	R1, D2 (X2, B2) или R1, D2 (B2)
ДЕЛЕНИЕ ДЕСЯТИЧНОЕ	DP	SS	D1 (L1, B1), D2 (L2, B2) или D1 (, B1), D2 (, B2)
ДЕЛЕНИЕ (ДЛИННОЕ)	DDR	RR	R1, R2
ПЕРЕХОД С ВОЗВРАТОМ	BAL	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
УСЛОВНЫЙ ПЕРЕХОД	BCR	RR	R1, R2
УСЛОВНЫЙ ПЕРЕХОД	BC	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ПЕРЕХОД ПО СЧЕТЧИКУ	BCTR	RR	R1, R2
ПЕРЕХОД ПО СЧЕТЧИКУ	BCT	RX	R1, D2 (X2, B2) или R1, D2 (, B2)

Команда	Код операции (мнемонический)	Формат стандарта	Формат операндов (явный адрес)
ПЕРЕХОД ПО ИНДЕКСУ БОЛЬШЕ	VXH	RS	R1, R3, D2 (B2)
ПЕРЕХОД ПО ИНДЕКСУ МЕНЬШЕ ИЛИ РАВНО	VXLE	RS	R1, R3, D2 (B2)
ПЕРЕХОД ПО «РАВНО»	VE	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО «БОЛЬШЕ»	VH	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО «МЕНЬШЕ»	VL	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО «НЕ РАВНО»	VNE	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО «НЕ БОЛЬШЕ»	VNH	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО «НЕ МЕНЬШЕ»	VNL	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО ЗНАКУ «-»	VM	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО ЗНАКУ «+»	VP	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО «НУЛЮ»	VZ	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО «НЕ -»	VNM	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО «НЕ +»	VNP	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО «НЕ 0»	VNZ	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД ПО ПЕРЕПОЛНЕНИЮ	VO	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД, ЕСЛИ ВСЕ 1	VO	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД, ЕСЛИ НЕ ВСЕ 1	VNO	RX	D2 (X2, B2) или D2 (, B2)
ПЕРЕХОД, ЕСЛИ ЕСТЬ 1 И 0	VM	RX	D2 (X2, B2) или D2 (, B2)

Команда	Код операции (мнемонический)	Формат стандарта	Формат операндов (явный адрес)
ПЕРЕХОД, ЕСЛИ ВСЕ 0	BZ	RX	D2 (X2, B2) или D2 (, B2)
СДВИГ ВПРАВО КОДОВ	SRA	RS	R1, D2 (B2)
СДВИГ ВПРАВО ДВОЙНОЙ КОДОВ	SRDL	RS	R1, D2 (B2)
ЗАПИСЬ В ПАМЯТЬ	ST	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ЗАПИСЬ В ПАМЯТЬ ПОЛУСЛОВА	STH	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ЗАПИСЬ В ПАМЯТЬ СИМВОЛА	STC	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ЗАПИСЬ В ПАМЯТЬ ГРУППОВАЯ	STM	RS	R1, R3, D2 (B2)
ЗАПИСЬ В ПАМЯТЬ (ДЛИННАЯ)	STD	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ЗАПИСЬ В ПАМЯТЬ (КОРОТКАЯ)	STE	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ВЫЧИТАНИЕ	SR	RR	R1, R2
ВЫЧИТАНИЕ	S	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ВЫЧИТАНИЕ ПОЛУСЛОВА	SH	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ВЫЧИТАНИЕ ДЕСЯТИЧНОЕ	SP	SS	D1 (L1, B1), D2 (L2, B2) или D1 (, B1), D2 (B2)
ВЫЧИТАНИЕ КОДОВ	SLR	RR	R1, R2
ВЫЧИТАНИЕ КОДОВ	SL	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ВЫЧИТАНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ)	SDR	RR	R1, R2
ВЫЧИТАНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ)	SD	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ВЫЧИТАНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ)	SER	RR	R1, R2

Продолжение табл. П-2

Команда	Код операции (мнемонический)	Формат стандарта	Формат операндов (явный адрес)
ВЫЧИТАНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ)	SE	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ)	SWR	RR	R1, R2
ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ)	SW	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ)	SUR	RR	R1, R2
ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ)	SU	RX	R1, D2 (X2, B2) или R1, D2 (, B2)
ОБРАЩЕНИЕ К СУПЕРВИЗОРУ	SVC	RR	I2
ОПРОСИТЬ КАНАЛ	TCH	SI	D1 (B1)
ОПРОСИТЬ ВВОД-ВЫВОД	TIO	SI	D1 (B1)
ПРОВЕРИТЬ И УСТАНОВИТЬ	TS	SI	D1 (B1)
ПРОВЕРИТЬ ПО МАСКЕ	TM	SI	D1 (B1), I2
ПЕРЕКОДИРОВАТЬ	TR	SS	D1 (L, B1), D2 (B2) или D1 (, B1), D2 (B2)
ПЕРЕКОДИРОВАТЬ И ПРОВЕРИТЬ	TRT	SS	D1 (L, B1), D2 (B2) или D1 (B1), D2 (B2)
РАСПАКОВАТЬ	UNPK	SS	D1 (L1, B1), D2 (L2, B2) или D1 (, B1), D2 (, B2)
ПРЯМАЯ ЗАПИСЬ	WRD	SI	D1 (B1), I2
СЛОЖЕНИЕ С ОЧИСТКОЙ	ZAP	SS	D1 (L1, B1), D2 (L2, B2) или D1 (, B1), D2 (, B2)

В табл. П-2 используются следующие обозначения:

R1, R2, R3 — абсолютные выражения, определяющие номера общих регистров 0—15 или номера регистров с плавающей точкой 0, 2, 4 и 6;

- X2 — абсолютное выражение, определяющее номер регистра индекса 0—15;
- B1, B2 — абсолютные выражения, определяющие номера регистров базы 0—15;
- D1, D2 — абсолютные выражения, определяющие смещение, которое должно находиться в пределах от 0 до 4095;
- L, L1, L2 — абсолютные выражения, определяющие длины операндов. Значение L должно быть в пределах от 0 до 256, значения L1 и L2 — в пределах от 0 до 16;
- I2 — абсолютное выражение, которое определяет непосредственный операнд машинной команды. Значение выражения должно быть в пределах от 0 до 255;
- S1, S2 — абсолютные и переместимые выражения, определяющие неявный адрес,

### СПИСОК ЛИТЕРАТУРЫ

1. Каган Б. М. Электронные вычислительные машины и системы. — М.: Энергия, 1979. — 528 с.
2. Папернов А. А. Логические основы ЦВТ. — М.: Советское радио, 1972. — 592 с.
3. Селлерс Ф. Методы обнаружения ошибок в работе ЦВМ. — М.: Мир, 1972. — 310 с.
4. Основы технической эксплуатации ЭЦВМ / А. Л. Верцайзер, И. И. Буличев, Л. Н. Вишнеvский, Г. Г. Евсеев. — М.: Энергия, 1973. — 360 с.
5. Дроздов Е. А., Камарницкий В. А., Пятибратов А. П. Электронные вычислительные машины Единой системы. — М.: Машиностроение, 1976. — 672 с.
6. Фатеев А. Е., Ройтман А. И., Фатеева Т. П. Прикладные программы в системе математического обеспечения ЕС ЦВМ. — М.: Статистика, 1976. — 184 с.
7. Орлов И. А. Организация работы вычислительного центра. — М.: Энергия, 1978. — 182 с.
8. Журавлев Ю. П., Котелюк Л. А., Циклинский Н. И. Надежность и контроль ЭВМ / Под редакцией Ю. П. Журавлева. — М.: Советское радио, 1978. — 416 с.
9. Чжен Г., Мэннинг Е., Метц Г. Диагностика отказов цифровых вычислительных систем. — М.: Мир, 1972. — 232 с.
10. Миронов Г. А. Испытательные программы для контроля электронных цифровых машин. — М.: Наука, 1964. — 268 с.
11. Стэбли Д. Логическое программирование в системе IBM / 360 — М.: МИР, 1974. — 752 с.
12. Азеев А. А., Саприн В. В., Чеботов Ю. А. Организация и функционирование вычислительного центра. — М.: Статистика, 1977. — 216 с.
13. Журавлев Ю. П., Котелюк Л. А., Циклинский Н. И. Надежность и контроль ЭВМ. — М.: Советское радио, 1978. — 416 с.
14. Барзилович Е. Ю., Каштанов В. А. Некоторые математические вопросы теории обслуживания сложных систем, — М.: Советское радио 1971. — 272 с.

15. Барзилович Е. Ю., Каштанов В. А. Организация обслуживания при ограниченной информации о надежности системы. — М.: Советское радио, 1975. — 136 с.

16. Дедков В. К., Северцев Н. А. Основные вопросы эксплуатации сложных систем. — М.: Высшая школа, 1976. — 406 с.

17. Васюкович Ю. В. Некоторые методические вопросы технико-экономической оценки ЭВМ. — Вопросы радиоэлектроники. Сер. ЭВТ, 1972, вып. 1, с. 141—148.

18. Бондаренко Б. К., Рохмистров А. Н. О технико-экономической оценке мероприятий по техническому обслуживанию ЭЦВМ. — Известия ДЭТИ, 1968, вып. 71, с. 37—41.

19. Казначеев В. И. Диагностика неисправностей цифровых автоматов. — М.: Советское радио, 1975. — 256 с.

20. Оценка эффективности вычислительной системы по коэффициенту «стоимость — загрузка». Экспресс-информация «ВТ» № 25, 1977.

21. Вентцель Е. С. Исследование операций. — М.: Советское радио, 1972. — 552 с.

22. Соколенко В. М., Разин В. П., Широких В. Е. Опыт эксплуатации ЭВМ ЕС-1030. — В кн.: Механизация учета, отчетности и вычислительных работ. Вып. 19. — М.: ЦСУ СССР, 1978, с. 24—34.

23. Еремин Ю. И. Методика проведения текущего технического обслуживания электронного оборудования ЭВМ. — Препринт ВЦ СО АН СССР 28. — Новосибирск: 1976. — 27 с.

24. Г. П. Лопато. Исследование производительности и технико-экономической эффективности универсальной ЭВМ ЕС-1020. — В кн.: Алгоритмы и организация решения экономических задач, Вып. 2. — М.: Статистика, 1973, с. 23—31.

## АЛФАВИТНЫЙ УКАЗАТЕЛЬ

**Адрес головки** 169  
 — дорожки магнитного диска 169  
 — цилиндра магнитного диска 169  
**Алгоритм диагноза** 29  
 — Рота 188  
**Ассемблер** 118  
**Байт состояния НМД** 173  
 — — НМЛ 198  
 — уточненного состояния НМД 173  
 — — — НМЛ 162  
 — циклического контроля НМД 169  
**Бегущая единица** 150  
**Бегущий ноль** 150  
**Безотказность ЭВМ** 9  
**Бригада профилактики и ремонта** 228  
**Буферный язык входных сообщений** 128  
**Вес теста** 194  
**Вспомогательные команды тестовой программы** 119  
**Глагольный язык входных сообщений** 128  
**Глубина контроля** 119  
**Двойной счет** 115  
**Детектор** 83  
**Дублирование схем** 45, 48, 88  
**ЗИП** 7  
**Идентификатор гнездовой** 196  
 — поля счета 170  
**Информационные символы** 51  
**Код «2 из 5»** 62  
 — с проверкой четности 53  
 — Хемминга 55  
 — шахматный 156  
**Кодовое расстояние** 52  
**Коды обнаружения и исправления ошибок** 49  
 — избыточные 50  
 — корректирующие 51  
 — неравновесные 49  
 — постоянные и переменные 136  
 — простые 50  
 — равновесные 62  
 — равномерные 49  
 — систематические 51  
 — циклические 58  
**Команда ЗАПИСАТЬ** 162  
 — ЗАРЕЗЕРВИРОВАТЬ 162  
 — ПЕРЕМОТАТЬ 161  
 — ПОИСК ПО КЛЮЧУ 171  
 — ПОИСК ПО КЛЮЧУ И ДАННЫМ 171  
 — ПОИСК ПО РАВЕНСТВУ СОБСТВЕННОГО АДРЕСА 171  
 — СТЕРЕТЬ ПРОМЕЖУТОК 161  
 — СЧИТАТЬ 163  
 — — ДАННЫЕ 171  
 — — — В ОБРАТНОМ НАПРАВЛЕНИИ 163  
 — — ПЗП 172  
 — — СОБСТВЕННЫЙ АДРЕС 171  
 — — СЧЕТ 171

**КОМАНДА УПРАВЛЯТЬ ОБЛАСТЬЮ СЧЕТА** 172  
 — УСТАНОВИТЬ МАСКУ 172  
 — УСТАНОВКА 171  
 — — ГОЛОВКИ И ОРИЕНТАЦИИ 171  
 — — ЦИЛИНДРА 173  
 — УТОЧНИТЬ СОСТОЯНИЕ 161  
 — ШАГ НА ГРУППУ ЗОН ВПЕРЕД 163  
 — — — ЗОНУ ВПЕРЕД 163  
 — — — НАЗАД 167  
**Контроль аппаратурный** 27  
 — — внешний 88  
 — — оперативный 88  
 — ввода-вывода информации 90  
 — внутренних цепей передач и информации 92  
 — комбинированный 27  
 — по модулю 63  
 — работы сумматоров 93  
 — эффективность 119  
**Контрольная информация** 91  
**Корректирующая способность кода** 52  
**Коэффициент полезного действия теста** 119  
 — технического обслуживания ЭВМ 250  
**Коэффициенты загрузки оборудования** 255  
**Мажоритарный элемент** 60  
**Математическая и логическая проверка при программно-логическом контроле** 115  
**Материальное обеспечение обслуживания ЭВМ** 15  
**Метод критерия различности** 192  
 — лучший следующий или возврат к хорошему 189  
 — пересечения тестовых наборов 192  
 — раскрутки 203  
**Микропрограммы диагностические** 211  
**Мультипроцессорные системы** 48  
**Надежность, показатели** 10  
 — работы ЭВМ 10  
 — тестовой программы 120  
**Наладочный режим тестового контроля** 126  
**Наработка на отказ** 234  
**Неисправности схем контроля** 44  
**Организация эксплуатации** 14  
**Основные команды тестовой программы** 119  
**Останов** 46  
**Отказы устройств ЭВМ** 20, 248  
**Относительное быстродействие тестовой программы** 120  
**Отношение стоимость / производительность** 252, 256, 258  
**Отображение неисправности** 190  
**Ошибки в оборудовании** 44  
 — в программах 42  
 — в устройствах 43  
 — оператора 42  
 — систематические 210  
 — случайные 210



- Парафазная логика 98
- Плано-профилактические работы
  - 17
- Поле ключа магнитного диска 170
  - счета магнитного диска 169
- Полная таблица неисправностей 192
- Полнота охвата 118
- Принцип расширяющихся областей
  - сужающихся областей 121
- Проверка схем контроля 37
- Проверочные символы 61
  - тесты 115
- Программа исправляющая 209
  - обслуживания тестовой ленты 123
  - рабочая 216
- Программно-логический контроль 115
- Программные средства контроля ЭВМ 246
- Производительность ЭВМ 12
  - комплексная 149, 251
  - номинальная 249
  - относительная 250
  - системная 257
  - совокупная 250
- Профилактика 12
  - время функционального контроля 234
  - длительность 233
  - коэффициент эффективности 235
  - режимы 236
- Профилактические работы, график 231
  - — календарные 230
- Профилактический контроль, методика 240
  - — этапы 237
  - режим тестового контроля 126
- Процессор, контрольные точки 214, 245
- Работоспособность ЭВМ 9
- Режим адресного кода 156
  - записи и считывания единиц 156
  - — — нулей 156
  - мультипрограммный 253
  - падающего дождя 154
  - параллельной обработки программ 254
  - работы ЭВМ диагностический 212
  - разделения времени 254
- Резервирование аппаратуры 22
  - замещением 70
  - постоянное 69
- Реконфигурация 47
- Ремонт профилактический 40
  - текущий 39
- Ремонтпригодность ЭВМ 9
- Сбой 21
- Сервис 6
  - групповой 7
  - индивидуальный 6
  - централизованный 7
- Сигнал ошибки 29, 47
- Система диагностики 30
  - дуплексная 48
  - контроля 24
  - обнаружения ошибок 29
- Система обслуживания 14
  - оценка эффективности 38
  - требования к системе контроля 34
- Системы с реконфигурацией 48
  - тестов 120
  - устойчивые к ошибкам 48
- Словарь неисправностей 195
  - — гнездовой 196
  - — микротестов каналов 225
  - — — оперативной и локальной памяти 221
  - — — процессора ЕС-1022 215
  - — — точного соответствия 195
- Служба технического обеспечения ЭВМ 227
- Сокращенная таблица неисправностей 192
- Состав обслуживающего персонала 16
- Стоимость часа вычислительного функционирования 261
  - часа, техническая 262
  - эксплуатации ЭВМ 260
- Схема НЕЧЕТ 76
  - ЧЕТ 76
- Таблица неисправностей 191
- Табличные методы синтеза тестов 190
- Тест 118
  - арифметики с плавающей запятой 141
  - базовый 123
  - десятичной арифметики 141
  - ДЕТЕКТ Рота 190
  - диагностический 26, 118
  - — комбинационных схем 183
  - — магнитной ленты 197
  - — последовательных схем 188
  - микропрограммный ДГВХ 218
  - — базисный 214
  - — запуск 212, 226
  - — локальной памяти 218
  - — мультиплексного канала 222
  - — оперативной памяти 219
  - — останов 214
  - МОНИТОР 123, 245
  - наладочный 26
  - проверочный 26
  - секции 123
  - стандартного набора команд 141
  - эвристический 189
- Тестовые программы 118
- Тестовый контроль 118
- Техническое обслуживание ЭВМ 6
- Типовой пакет сменных магнитных дисков 168
- Типы неисправностей ЭВМ 237
- Тяжелый режим работы ЭВМ 127
- Управляющая программа тест-монитора 123
- Условная надежность 120
- Функциональный тест проверки магнитной ленты 160
- Эксплуатационная документация 17
- Эксплуатация ЭВМ 6
  - планирование 18
- Эффективность кода
- Эхо-контроль 91

## О Г Л А В Л Е Н И Е

Предисловие . . . . .	3
Глава первая. Эксплуатация ЭВМ . . . . .	6
1-1. Общие понятия . . . . .	6
1-2. Основные эксплуатационные характеристики . . . . .	9
1-3. Принципы организации эксплуатации . . . . .	14
Глава вторая. Контроль работы ЭВМ . . . . .	19
2-1. Классификация ошибок и неисправностей ЭВМ . . . . .	19
2-2. Основные задачи контроля и диагностики ЭВМ . . . . .	22
2-3. Структура системы контроля и диагностики ЭВМ . . . . .	25
Глава третья. Общие принципы обнаружения ошибок и ремонта ЭВМ . . . . .	28
3-1. Системы автоматического контроля и диагностики и их взаимодействие . . . . .	28
3-2. Основные требования к системе автоматического конт- роля и диагностики . . . . .	33
А. Общие требования к системе . . . . .	33
Б. Средства проверки схем контроля . . . . .	37
В. Оценка эффективности контроля . . . . .	38
3-3. Общий подход к ремонту машины . . . . .	38
А. Виды ремонтных работ . . . . .	39
Б. Основные виды ошибок и принцип подхода к ним . . . . .	42
В. Контроль специальных схем . . . . .	45
Г. Останов из-за ошибки . . . . .	46
Д. Работа на неисправной машине . . . . .	46
Е. Реконфигурация как средство достижения высокой надежности . . . . .	47
Глава четвертая. Основы построения схемного конт- роля . . . . .	49
4-1. Коды для обнаружения и исправления ошибок . . . . .	49
А. Общие сведения о кодах . . . . .	49
Б. Контроль передачи информации . . . . .	53
В. Контроль арифметических и логических операций . . . . .	63
Г. Сравнительные характеристики избыточных кодов . . . . .	67
4-2. Резервирование как метод устранения ошибок . . . . .	69
Глава пятая. Методы аппаратурного контроля и их реа- лизация . . . . .	71
5-1. Контроль четности . . . . .	71
5-2. Контроль с помощью кода Хэмминга . . . . .	79
5-3. Контроль с помощью равновесных кодов . . . . .	81
5-4. Пороговые схемы . . . . .	83

5-5. Схемы контроля по модулю . . . . .	85
5-6. Обнаружение ошибок методом аппаратурного контроля . . . . .	88
<b>Глава шестая. Проверочные (контролирующие) тесты</b> . . . . .	<b>115</b>
6-1. Программно-логический контроль . . . . .	115
6-2. Тестовый контроль . . . . .	118
6-3. Формирование системы тестов . . . . .	121
6-4. Система тестов ТЕСТ-монитор единой системы ЭВМ . . . . .	123
6-5. Работа оператора с системой ТЕСТ-монитор . . . . .	126
6-6. Тестовый контроль процессора . . . . .	131
6-7. Тесты в системе ТЕСТ-монитор для проверки процессора . . . . .	141
6-8. Проверка ОЗУ . . . . .	149
6-9. Тест проверки МОЗУ в системе ТЕСТ-монитор . . . . .	156
6-10. Проверка внешних запоминающих устройств . . . . .	159
<b>Глава седьмая. Диагностика неисправностей ЭВМ</b> . . . . .	<b>183</b>
7-1. Диагностика неисправностей элементов ЭВМ . . . . .	183
7-2. Метод активизации одномерного пути . . . . .	185
7-3. Табличные методы синтеза минимального набора тестов . . . . .	190
7-4. Словари неисправностей . . . . .	195
7-5. Диагностический тест магнитной ленты в системе ТЕСТ-монитор . . . . .	197
7-6. Построение системы автоматической диагностики ЭВМ методом раскрутки . . . . .	203
<b>Глава восьмая. Программно-аппаратурные методы контроля и диагностики ЭВМ</b> . . . . .	<b>207</b>
8-1. Сравнительная характеристика аппаратурных и программных методов контроля и диагностики ЭВМ . . . . .	207
8-2. Основные понятия программно-аппаратурных методов контроля и диагностики ЭВМ . . . . .	208
8-3. Обнаружение ошибок центрального процессора ЭВМ микропрограммными методами . . . . .	211
8-4. Микропрограммные тесты запоминающих устройств и каналов ввода-вывода . . . . .	218
<b>Глава девятая. Основы организации и планирования технического обслуживания ЭВМ</b> . . . . .	<b>227</b>
9-1. Цели и задачи технического обслуживания ЭВМ . . . . .	227
9-2. Планирование работ по техническому обслуживанию . . . . .	229
9-3. Количественные оценки профилактических мероприятий . . . . .	233
<b>Глава десятая. Автоматизация профилактических работ</b> . . . . .	<b>235</b>
10-1. Способы и методы профилактического контроля . . . . .	235
10-2. Организация профилактического контроля в ЕС ЭВМ . . . . .	244
<b>Глава одиннадцатая. Технико-экономическая оценка мероприятий по техническому обслуживанию ЭВМ</b> . . . . .	<b>249</b>
11-1. Выбор и обоснование критериев эффективности использования ЭВМ . . . . .	249
11-2. Экономическая эффективность системы технического обслуживания ЭВМ . . . . .	260
<b>Приложение</b> . . . . .	<b>263</b>
<b>Список литературы</b> . . . . .	<b>274</b>
<b>Алфавитный указатель</b> . . . . .	<b>276</b>

ИГОРЬ АЛЕКСЕЕВИЧ ОРЛОВ,  
ВАЛЕРИЙ ФЕДОРОВИЧ КОРНЮШКО,  
ВАЛЕРИЙ ВИКТОРОВИЧ БУРЛЯЕВ,  
ВЛАДИСЛАВ НИКОЛАЕВИЧ АВДЕЕВ

## ОСНОВЫ ЭКСПЛУАТАЦИИ И РЕМОНТА ЭВМ

Редактор *Б. Н. Севрюков*  
Редактор издательства *Л. Д. Никулина*  
Переплет художника *В. П. Осипова*  
Технический редактор *Н. П. Собакина*  
Корректор *И. А. Володяева*

ИБ № 2008 («Энергия»)

---

Сдано в набор 07.05.80. Подписано в печать 09.02.81. Т-00989. Формат 84×108<sup>1</sup>/<sub>32</sub>. Бумага типографская № 3, Гарн. шрифта литературная. Печать высокая. Усл. печ. л. 14,70. Уч.-изд. л. 16,42. Тираж 40 000 экз. Заказ 752. Цена 65 к.

---

Энергондат, 113114, Москва, М-114, Шлюзовая наб., 10

---

Набрано в орден Октябрьской Революции, ордена Трудового Красного Знамени Ленинградском производственно-техническом объединении «Печатный Двор» имени А. М. Горького Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли. 197136, Ленинград, П-136, Чкаловский пр., 15.

Отпечатано во Владимирской типографии «Союзполиграфпрома» при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли

• 600000, г. Владимир, Октябрьский проспект, д. 7