

ББК 32.97

Ф43

УДК 681.3

Фергусон Дж., Макари Л., Уилльямз П.

**Ф43 Обслуживание микропроцессорных систем: Пер. с англ.—
М.: Мир, 1989. — 336 с., ил.**

ISBN 5-03-001252-4

В книге американских специалистов рассматриваются вопросы контроля, диагностики и поддержания работоспособности микропроцессорных систем. Большой интерес представляют приложения, в которых дается сводка характеристик широко используемых микропроцессоров, приводятся схемы источников и регуляторов напряжения питания, описываются варианты схемной организации сигнатурных и логических анализаторов.

Для разработчиков и пользователей вычислительной техники, а также студентов соответствующих специальностей вузов.

**Ф 2404020000-046
041(01)-89 146-90**

ББК 32.97

Редакция литературы по информатике и робототехнике

**ISBN 5-03-001252-4 (русск.)
ISBN 0-13-581132-5 (англ.)**

**© 1987 John D. Ferguson, Louie Macari &
Peter Williams
© перевод на русский язык, «Мир», 1989**

Предисловие переводчика

Обеспечение работоспособности микропроцессорных систем на всех стадиях их проектирования, производства, отладки и эксплуатации невозможно без использования специальных методов и приборов, ориентированных на специфические особенности их архитектуры и конструктивного исполнения. Дело в том, что высокая степень интеграции компонентов микропроцессорных систем, а также простота их внутренних межсоединений и внешних связей, которые безусловно являются положительными факторами, затрудняют получение детальной информации о состоянии отдельных узлов, блоков и линий связи, поиск неисправностей. Обслуживание микропроцессорных систем требует привлечения дополнительных сведений и аппаратуры по сравнению с обслуживанием традиционных электронных систем и, в частности, вычислительных машин. В то же время оно должно базироваться на знании общих принципов функционирования электронных схем и на известных принципах логического подхода к анализу их работы. Развитие и совершенствование микропроцессорной техники сопровождаются разработкой и все более широким использованием различных по сложности и разнообразных по принципам действия специализированных средств для ее обслуживания.

Вопросы обслуживания микропроцессорных систем уже нашли отражение как в монографиях и учебниках по микропроцессорам, так и в книгах отечественных и зарубежных авторов, специально посвященных этой тематике¹⁾. В книге, предлагаемой вниманию читателей, сделана попытка охватить довольно широкий круг вопросов — от основ схемотехники и принципов создания микропроцессорных устройств до проблем построения приборов и систем, предназначенных для отладки микропроцессорных средств. Здесь описаны простые средства контроля цифровых схем, дан обзор архитектурных особенностей микропроцессорных систем, предложена классификация причин некорректной работы устройств и систем. Значительное внимание удалено таким специфичным методам отладки, как внутрисхемная эмуляция и сигнатурный анализ. В приложениях дана развернутая сводка характеристик некоторых наиболее популярных микропроцессоров. В книге удачно сочетаются сведения по аппаратным и программным аспектам. Излагаемый материал по сервисным средствам, различным по сложности, быстродействию и стоимости, имеет в значительной степени прикладной характер. Читатель получит представление о рациональном выборе этих средств в зависимости от конкретных особенностей обслуживаемых микропроцессорных систем.

Книга несомненно будет полезна инженерам, занимающимся вопросами разработки и эксплуатации микропроцессорной техники, а также студентам соответствующих специальностей.

B. M. Кисельников

¹⁾ См., например: Микропроцессоры: системы программирования и отладки/В. А. Мясников, М. Б. Игнатьев, А. А. Кочкин, Ю. Е. Шейнин.—М.: Энергоатомиздат, 1985; Уильямс Г. Б. Отладка микропроцессорных систем: Пер. с англ.—М.: Энергоатомиздат, 1988.

Предисловие

Обслуживание микропроцессорных систем предполагает овладение кругом знаний, отличным от того, который необходим для работы с традиционными электронными системами. Это не означает, что оно требует другого отношения к делу. Весьма уместным оказывается такой же, как и при работе с любыми электронными приборами, исследовательский метод, подразумевающий логически последовательный анализ функционирования устройств. И разработчики, и специалисты по обслуживанию могут быть уверены, что имеющийся у них опыт работы как с цифровыми, так и с аналоговыми схемами, а также с электромеханическими устройствами окажется применимым к микропроцессорам и системам, построенным на их основе.

Было бы неразумно отрицать наличие проблем, связанных с внедрением микропроцессорной техники, но при правильной оценке их влияние может быть сведено к минимуму. Авторы предполагают, что читатель знаком с основами электронной, в том числе цифровой, техники и логическими методами решения соответствующих задач. Мы не рассчитываем на всестороннее владение микропроцессорными средствами, однако более или менее глубокое знание принципов их работы окажется весьма полезным.

Целью, преследуемой в ч. I, является совместное рассмотрение указанных выше вопросов, чтобы читатели с различной предварительной осведомленностью оказались в равной степени подготовлены к восприятию новых методов, изложенных в основных разделах книги.

Часть II не должна рассматриваться как учебник по цифровой электронике или как еще одно из многочисленных пособий типа «Микропроцессоры для начинающих». Это скорее обзор состояния затронутой проблематики с точки зрения задач технического обслуживания. Такая точка зрения может оказаться не вполне привычной для инженера, имеющего опыт работы с электронными устройствами, хотя сведения, составляющие содержание первой части, и не покажутся ему незнакомыми. Читателю, в относительно большей степени подготовленному в области программирования, некоторые из высказываемых в ч. I «аппаратных» идей могут сначала тоже показаться странными, но, не проникшись этими идеями, невозможно всерьез говорить об обслуживании микропроцессорных систем.

В ч. II рассмотрены все известные средства обслуживания. При разработке микропроцессорных систем, в процессе их изготовления, в ходе обслуживания в период эксплуатации имеют место одни и те же общие трудности, для устранения которых могут быть применены эти средства. Виды информации, получаемые при использовании каждого из этих средств, могут различаться степенью применимости для конкретного пользователя. Разработчик должен знать, почему система не ведет себя должным образом, а специалист по обслуживанию, работающий в тесных временных и экономических рамках, отдаст предпочтение методам, обеспечивающим скорейшее устранение неисправностей.

В идеале каждый пользователь должен иметь в распоряжении полный набор профессионального оборудования.

В ч. III рассмотрены простые устройства, которые могут быть изготовлены собственными силами и использованы как для решения оперативных задач по обслуживанию микропроцессорных систем, так и в качестве недорогих учебных устройств. В связи с тем, что каждый пользователь может иметь свои индивидуальные особенности и задачи, схемы представлены в достаточно общей форме, предполагающей их конкретную реализацию силами опытных инженеров и техников-электронщиков.

В гл. 1 рассмотрены характеристики цифровых устройств и наиболее общие методы отыскания неисправностей. В гл. 2 описана архитектура микропроцессорных систем, причем акцент сделан на особенности, определяющие специфику подхода к их обслуживанию. В гл. 3 в таком же аспекте рассмотрены характеристики интерфейса микропроцессорных систем с периферийными устройствами. Глава 4 посвящена принципам поиска неисправностей, реализация которых не требует специального оборудования. При обсуждении в гл. 5 возможностей самоконтроля микропроцессорных систем предполагается, что обслуживаемая система сохраняет работоспособность в степени, достаточной для генерации тестовых наборов данных, по содержанию которых можно судить о правильности работы частей системы, подозреваемых в наличии неисправностей. Логические анализаторы, рассматриваемые в гл. 6, служат для оценки битовых информационных образов, последовательно формируемых на линиях системной шины и позволяющих как локализовать явные отказы компонентов, так и обнаруживать более трудно уловимые проявления некорректности временных соотношений. При передаче данных в системе возникают дополнительные причины отказов; в связи с этим в гл. 7 рассмотрены характеристики и процедуры контроля последовательных интерфейсов. Системная эмуляция как одно из основных средств систем разработки служит в первую очередь для решения задач проектирования микропроцессорных систем, однако в гл. 8 внимание сосредоточено на ее применении для поиска неисправностей. Глава 9 посвящена сигнатурному анализу, при котором осуществляется предварительное документирование функционирования системы с помощью специальных программ; комбинация битов для каждой ситуации является специфичной, и любое отклонение от нее рассматривается как признак неправильности работы системы. При поиске неисправностей в процессе конвейерного производства могут быть применены процедуры контроля, реализуемые с помощью ЭВМ; как показано в гл. 10, реальная тестовая аппаратура может различаться в широком диапазоне. В гл. 3 рассмотрены принципы построения тестового оборудования на основе микроЭВМ и разрабатываемых пользователем специализированных контролльных плат. В гл. 11 этот подход иллюстрируется на примере простого внутрисхемного эмулятора, организованного на базе стандартной микроЭВМ. Сигнатурные и логические анализаторы учебного назначения могут быть изготовлены в виде дешевых одноплатных устройств (гл. 12). Тестирование последовательных интерфейсов между ЭВМ и ее периферийными устройствами может быть осуществлено с помощью последовательных анализаторов и генераторов символов, рассматриваемых в гл. 13.

Дж. Фергусон
Л. Макари
П. Уилльямз

Часть I

Микропроцессорные системы

Глава 1

Цифровые устройства и системы

Для построения систем на базе микропроцессоров необходимы дополнительные схемы и устройства. Логические схемы разных серий существенно различаются по характеру функционирования, даже когда они выполняют одинаковые функции. Ниже кратко рассматриваются эти различия. Существуют контрольные устройства, позволяющие локализовать неисправность схемы с точностью до ее конкретного внешнего вывода; ниже рассматриваются преимущества и ограничения, присущие такому тестовому оборудованию.

Большую помощь при обслуживании любой системы оказывает хорошо написанное руководство. При знакомстве со всеми вопросами, затронутыми в данной книге, читатель должен проявить при формировании своего мнения искренний оптимизм, не обладая которым затруднительно заниматься обслуживанием микропроцессорных систем; итак, будем считать, что руководство весьма полезно, пока не обнаружится, что это не так.

К сожалению, очень часто руководство либо вообще отсутствует, либо составлено второпях. Оно должно содержать полную схему системы с указанием размещения компонентов, причем на схеме должны быть отмечены важные контрольные точки. Типичным примером проблемы, корректному решению которой поможет последнее обстоятельство, является измерение тактовой частоты. Дело в том, что счетчик и осциллограф, будучи подсоединенны к выходу генератора с высоким выходным сопротивлением, могут представлять собой нагрузку, вполне достаточную для искажения его частоты. Во избежание этого каждая контрольная точка, которую предполагается использовать в процессе изготовления и эксплуатации системы, должна быть отделена от схемы надежным буфером.

Руководство может содержать набор тестовых процедур, предназначенных для выявления основных видов неисправностей, возникновение которых предвидели разработчики. Только очень неразумный человек может пренебречь такими подсказками, хотя разработчики при написании руководства скорее всего располагают небольшой информацией о возможных отказах.

Следует понимать, что, поскольку конструктивное оформление микропроцессорной системы осуществляется с помощью печатных плат, любая типовая логическая схема с точки зрения обслуживания ведет себя *точно так же*, как и в устройствах предшествующих поколений. Хотя выполняемые функции в случае микропроцессорной системы и могут быть более *сложными*, характер электрических и механических отказов остается неизменным: замыкания печатных проводников, замыкания при пайке, обрывы. При работе с микропроцессорной системой может оказаться более затруднительным нахождение участка платы, где произошел отказ, но после того, как область предполагаемого отказа сведена к одной-двум интегральным схемам, могут быть применены хорошо известные методы тестирования. В данной главе будет дан краткий обзор этих методов; более полному их рассмотрению посвящено множество книг.

Когда следует применять простые методы тестирования, а когда более сложные, которые будут рассмотрены ниже? Если есть возможность воспользоваться автоматическими или полуавтоматическими тестами, применяйте их. Эти средства были разработаны профессионалами, имевшими значительно больше опыта и времени на раздумья, чем оказывается в вашем распоряжении, когда вы имеете дело с незнакомой системой. Если, к счастью, причина отказа проявляется явным образом, то оставленное — дело техники. Симптомов отказа, различимых на индикаторе или на выходном порте, может оказаться достаточно для формирования предположения о некорректно работающей задающей схеме, после чего контроль с помощью импульсного генератора и логического пробника позволяет быстро полностью идентифицировать отказ. В то же время при неудачном стечении обстоятельств у инженера по обслуживанию может в критический момент не оказаться под рукой ни подходящего анализатора, ни средств эмуляции. В этом случае единственным выходом остается возврат к традиционным методам анализа работы системы.

Назовем некоторые характерные виды информации, которые могут быть получены с помощью соответствующих пробников.

Статическая: логический 0; логическая 1; «плохой» уровень (значение напряжения между уровнями логических 0 и 1); обрыв; короткое замыкание.

Динамическая: наличие серии импульсов малой длительности; процесс изменения состояния; импульсные сигналы низкой частоты; импульсные сигналы обратной полярности; протекающий ток.

Уровень напряжения должен соответствовать логическому 0 или логической 1. Как было отмечено выше, величины напряжений, с гарантией обеспечивающие различие этих уровней, неодинаковы для разных видов схемотехники. В типовых ТТЛ-схемах

мых уровня напряжения, меньшие 0,8 В, служат для обозначения логического 0, а уровни, превышающие 2,4 В, — логической 1. Значения уровня в диапазоне от 0,8 до 2,4 В являются признаком неисправной работы, который должен индицироваться специальным сигналом пробника. Тем же сигналом обычно индицируется наличие обрыва, т. е. об обрыве и «плохом» логическом уровне пробник информирует одинаковым образом. Аналогично короткое замыкание с любым из проводников напряжения

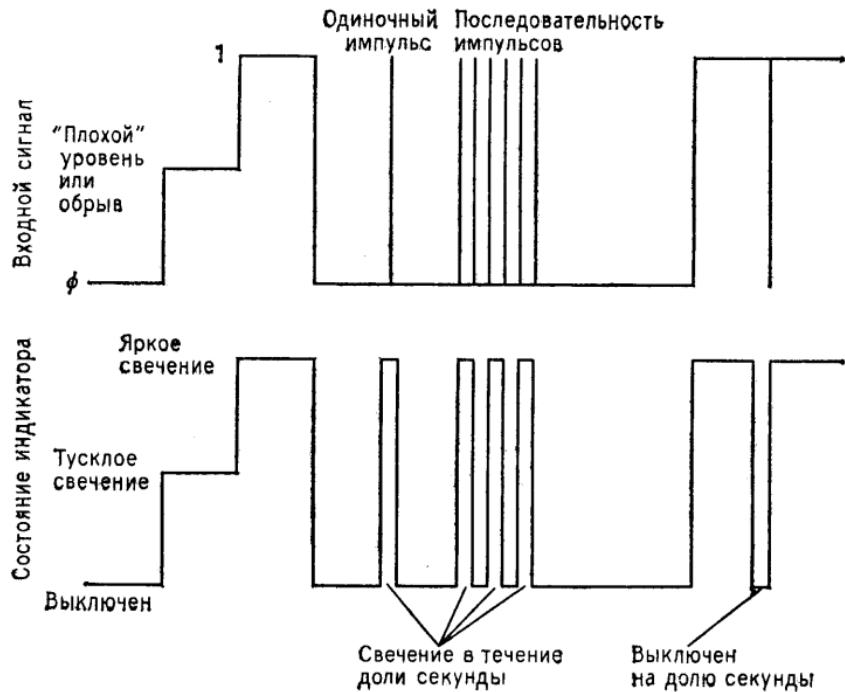


Рис. 1.1. Выявление статических и динамических логических состояний с помощью одиночного индикатора.

питания придает сигналу уровень, который однозначно воспринимается как допустимое значение. Отличить уровень реального выходного сигнала от уровня, являющегося следствием короткого замыкания сигнальной линии с линией питания, можно с помощью вольтметра или осциллографа.

Одного светодиода, снабженного соответствующим управляющим каскадом, достаточно для индикации трех видов возможных состояний сигнала:

- выключен
- тусклое свечение
- яркое свечение

- логический 0 (или короткое замыкание с общей шиной);
- обрыв или недопустимый (промежуточный между 0 и 1) уровень;
- логическая 1 (или короткое замыкание с шиной питающего напряжения).

Тестируировать можно не только статические сигналы. Если перед каскадом управления светодиодом ввести схему выявления переходного процесса и вибратор с одним устойчивым состоянием (одновибратор), то изменение значения входного сигнала преобразуется в импульс фиксированной длительности. При подключении пробника, оснащенного такой схемой, к точке схемы, в которой сигнал представляет собой узкий импульс длительностью 10 нс с полярностью, отрицательной по отношению к уровню 1, светодиод гаснет на долю секунды. Другая схема (задержки) блокирует аналогичную реакцию индикатора на последующие импульсные сигналы; в результате мерцающий сигнал светодиодного индикатора имеет частоту около 10 Гц при частоте входных импульсных сигналов индикатора, лежащей в диапазоне от этого значения до многих мегагерц.

Таким образом, с помощью одного светодиодного индикатора могут быть обнаружены как короткие перепады значения сигнала $0 \rightarrow 1$ и $1 \rightarrow 0$, так и серии повторяющихся импульсов (рис. 1.1).

С помощью двух индикаторов, называемых «верхним» и «нижним», может быть получена следующая информация о значении сигнала в контролируемой точке схемы:

Состояние индикаторов	Индцируемый сигнал
Верхний светится, нижний выключен	Логическая 1
Нижний светится, верхний выключен	Логический 0
Ни один не светится	«Плохой» уровень или обрыв
Оба мерцают	Серия импульсов низкой частоты
Оба тускло светятся	Серия импульсов высокой частоты (степень яркости позволяет судить о соотношении длительности импульса и паузы)

При добавлении в такой пробник третьего светодиода он мерцает при поступлении на вход пробника импульсной последовательности и совсем не светится, когда входной сигнал является статическим. Информация о соотношении между длительностью импульса и паузой при этом не выдается. Если же для индикации уровней логических 0 и 1 используются отдельные светодиоды, снабженные соответствующими пороговыми детекторами, может быть сформирована информация, приведенная на рис. 1.2.

Возможны и другие варианты организации работы логических пробников. Так, например, может быть применен режим фиксации, при котором светодиод сохраняет состояние свечения

даже при поступлении короткого одиночного импульса. Кроме того, изготовители придают пробникам различные параметры с точки зрения временных соотношений воспринимаемых импульсных сигналов. Пробники могут, например, реагировать на импульсы длительностью от 10 до 100 нс и вырабатывать при этом выходные импульсы длительностью от 50 до 300 мс. Существуют пробники, в которых возможно изменение пороговых уровней,

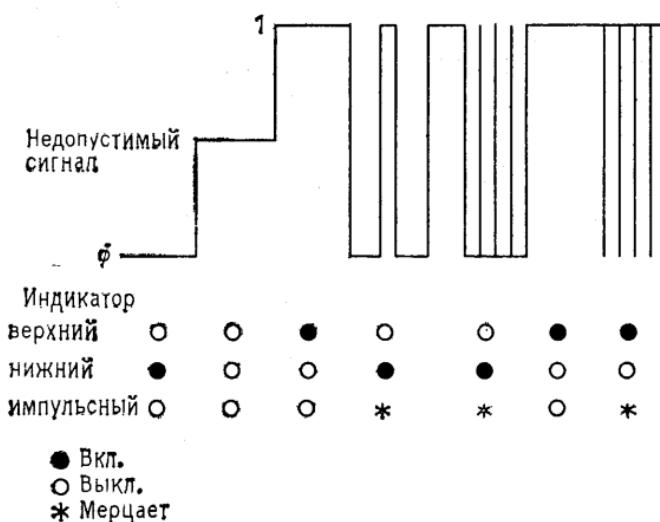


Рис. 1.2. Информация, получаемая с помощью логического пробника, снабженного тремя светодиодными индикаторами, включает сведения о соотношении длительности импульса и паузы в серии.

что позволяет использовать их для работы с различными видами схемотехники. Для максимальной реализации возможностей пробника обратитесь к руководству по его применению.

Токовый пробник

Некоторые виды отказов не могут быть обнаружены с помощью логических пробников, какими бы сложными они ни были. Так, логический пробник дает возможность установить наличие короткого замыкания в узле схемы, но не позволяет определить, в каком именно из устройств, связанных с этим узлом, имеет место это замыкание. Аналогично причиной замыкания между двумя точками схемы может быть либо внутренняя неисправность какой-либо интегральной схемы, либо непосредственное замыкание линий связи, проходящих через эти точки. В то же время нам необходимо узнать, где происходят неправильные перераспределения токов, не разрывая, если это возможно, проводников и не демонтируя интегральных схем. Подобные действия отнимают много времени, и их применение вле-

чет за собой риск, связанный с возможностью внесения дополнительных неисправностей, хотя в некоторых затруднительных ситуациях без них и не обойтись.

Обнаружение постоянного тока затруднено, хотя принципиально возможно; один из способов основан на использовании эффекта Холла. Трудным оказывается достижение чувствительности измерительного устройства, соответствующей уровням напряжения, фигурирующим в современных цифровых системах. Работа с импульсными токами представляет меньшие трудности, и соответствующий токовый пробник является мощным инструментом. Зонд пробника размещается вдоль проводника; при этом

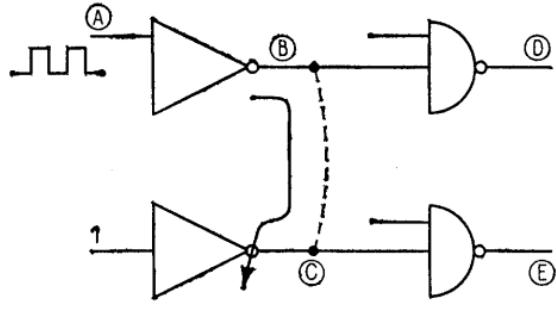


Рис. 1.3. Обнаружение короткого замыкания с помощью токового пробника.

обнаруживаются импульсные сигналы, имеющие место в проводнике. Чувствительность пробника может быть отрегулирована с таким расчетом, чтобы индикатор продолжал светиться. Затем пробник перемещают вдоль предполагаемого пути прохождения тока, пока индикатор не перестанет светиться, что свидетельствует о наличии непредусмотренного пути ответвления тока. Подчеркнем, что измерения в данном случае являются неконтактными: зонд должен тесно приблизиться к проводнику, но не касаться его.

Чтобы сравнить возможности потенциального и токового пробников, рассмотрим ситуацию, представленную на рис. 1.3. Серия импульсов, поступающая на вход А, должна в инвертированной форме иметь место в точке В, а затем в первоначальном виде (если не учитывать задержку) в точке Д. В точках С и Е никаких импульсных сигналов быть не должно. Потенциальный пробник может показать, что импульсы на выходе Д отсутствуют, а выходной сигнал имеет значение логической 1. Причина этого могут быть неисправность логического элемента, короткое замыкание выхода Д с шиной питания или наличие логического 0 в точке В. Перенесем потенциальный пробник в точку В. Если здесь есть серия импульсов, то, следовательно, либо неисправен выходной элемент, либо имеется короткое замыкание

в точке D. Отсутствие импульсной последовательности в точке B свидетельствует о том, что или неисправен входной инвертор, или в точке B произошло короткое замыкание с общей шиной, шиной питания или другой сигнальной линией.

Если в точке B сигнал логического 0, т. е. соответствует значению логической 1 в точке D, то, значит, выходной вентиль «хороший». Если входной инвертор исправен, токовый пробник покажет наличие серии импульсов в точке B. Отрегулируем чув-

ствительность так, чтобы входного сигнала было лишь достаточно для свечения индикатора. Будем перемещать пробник по направлению к выходу схемы, пока индикатор не погаснет. Осмотр платы скорее всего при этом позволит обнаружить перемычку из припоя между проводниками или другую подобную неисправность. Если таких неисправностей найдено не будет, следует просмотреть соседние участки схемы, такие, например, как точку C, на которые может происходить ответвление тока, чтобы выявить физическое размещение цепи короткого замы-

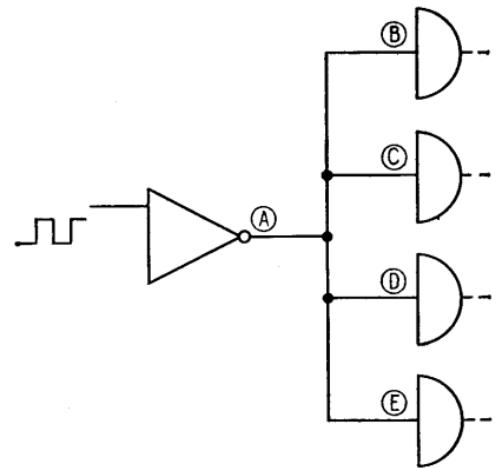


Рис. 1.4. Обнаружение короткого замыкания путем выявления неравенства входных токов.

кания, показанной на рисунке пунктирной линией, если утечка не происходит прямо на общую шину.

Когда выход логического элемента должен управлять несколькими входами, схема обычно рассчитывается таким образом, чтобы на все эти входы поступали токи примерно одинаковой величины. Так, в схеме, показанной на рис. 1.4, управляющий логический элемент должен вырабатывать импульсный ток 8 мА, чтобы на каждый из четырех управляемых входов поступал ток около 2 мА. Совсем необязательно, чтобы это разветвление происходило с высокой точностью, лишь токи должны быть одного порядка. Но и токовый пробник помогает различать их лишь при большом разбросе значений, который может иметь место только в случае неисправности. Разместим пробник так, чтобы он отреагировал на серию импульсов в точке A (если серию обнаружить не удается, значит, либо неисправен сам инвертор, либо произошли обрывы в соединениях со всеми четырьмя входами). Будем перемещать пробник по направлению ко всем четырем входам по очереди. Если индикатор продолжает светиться при движении только к одному из входов, весьма высока вероят-

ность наличия короткого замыкания. Если свечение не сохраняется при перемещении ни к одному из нагрузочных входов, в этом нет ничего удивительного: каждый из входных токов может быть слишком мал для восприятия его индикатором при ранее установленном пороге чувствительности. Если чувствительность индикатора можно регулировкой в достаточной степени повысить, то эти небольшие, но более или менее одинаковые токи должны быть обнаружены.

Логический пульсатор

В рассмотренных выше примерах предполагалось, что на схему поступает соответствующая серия импульсов. Если это не так, надо иметь возможность подачи сигналов на схему. Делать это можно, физически вторгаясь в схему, т. е. перерезая проводники, удаляя интегральные схемы и т. д. Однако против такого

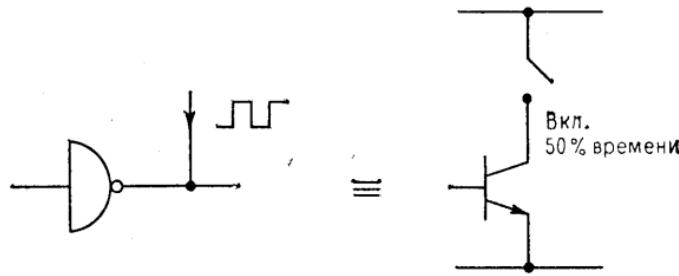


Рис. 1.5. Потенциальный пульсатор вызывает большое рассеяние мощности.

подхода уместны те же возражения, которые были приведены при рассмотрении вопроса об измерении токов. Действительно, подобные приемы допустимы, но лучше обойтись без них.

Нужно иметь устройство, которое можно подключить к любой точке системы и без опасности повреждения схемы вводить с его помощью серию импульсов в эту точку независимо от того, пребывает он в состоянии логического 0 или 1. Генератор импульсов типа «меандр» с выходным импедансом, намного меньшим, чем импеданс точки, к которой надо его подключать, для таких целей не подходит. На выход любого логического элемента, управляемого от такого генератора, будет подан избыточный ток, следствием которого явится повышенное рассеяние мощности. Так, при подаче серии прямоугольных импульсов на выход ТТЛ-элемента, находящийся в состоянии логического 0, в течение 50 % времени (т. е. в те интервалы, когда этот выход оказывается накоротко соединенным с положительным полюсом источника питания, рис. 1.5) будет происходить избыточное рассеяние мощности.

Если долю времени, когда будет происходить избыточное рассеяние, снизить до 1—2 %, то даже при достаточно больших пиковых значениях тока средняя величина рассеяния проблем не составит. Главное состоит в том, чтобы сделать импульсный генератор, форма выходного сигнала которого менялась бы в зависимости от статического уровня, имеющего место в точке схемы, куда этот сигнал подается. Для точки, в которой статический сигнал — логический 0, должны вырабатываться короткие импульсы логической 1, и наоборот. Таким образом, в течение каждой паузы между импульсами генератор должен выяснять состояние тестируемого узла и изменять в случае необходимости

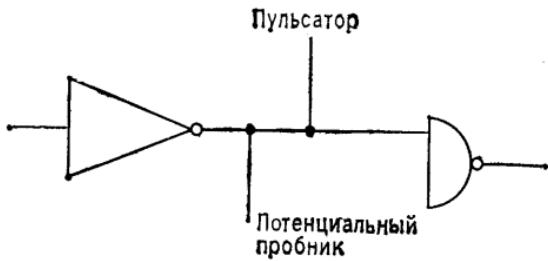


Рис. 1.6. Наблюдение сигналов токового пульсатора с помощью потенциального пробника.

полярность очередного импульса. Таким образом, выход генератора должен иметь три состояния. В состоянии разомкнутой цепи внутренний детектор выясняет логический уровень контролируемой точки и соответствующим образом определяет условия формирования следующего импульса. Эта закономерность соответствует изменению выходных сигналов триггерных схем, в которых входной импульс может изменить значение выходного сигнала на противоположное за счет срабатывания обратных связей.

С помощью пульсатора можно определить наличие коротких замыканий с общей шиной. Обладая внутренне присущей возможностью ограничения тока, пульсатор полностью определяет состояние выходного сигнала ТТЛ-элемента, подавляя собственное значение этого сигнала, а потенциальный пробник позволяет обнаружить сигналы, генерируемые пульсатором. При наличии короткого замыкания (или при уменьшении импеданса по отношению к общейшине ниже нормы) импульсы напряжения обнаружить не удается (рис. 1.6).

Воспользовавшись токовым пробником вместо потенциального, можно просмотреть печатные проводники во всех направлениях от места подключения пульсатора и определить пути протекания тока. Исчезновение серии токовых импульсов указывает на точку, в которой имеется короткое замыкание. С по-

мощью этого приема можно также выявить неисправный логический элемент, входящий в состав схемы «монтажное ИЛИ» (рис. 1.7)¹⁾. Такая схема по существу представляет собой набор параллельно включенных транзисторов, открывание любого из которых приводит к изменению логического значения выходного сигнала. Если один из этих транзисторов постоянно открыт, то с помощью входных импульсных сигналов невозможно определить, какой именно. Подавая токовые импульсы на выход логической схемы и перемещая токовый пробник по направлению к отдельным транзисторам, можно установить, какой из транзисторов находится постоянно в проводящем состоянии.

Большинство пульсаторов допускает подачу как одиночных импульсов, так и серий импульсов. Преимуществом первого из этих режимов является дальнейшее снижение риска, связанного с повышенным рассеянием мощности, а второй удобен при исследовании триггерных и счетных схем, а также схем задержки. Длительность выходных импульсов лежит в пределах от 10 нс до 10 мкс, что позволяет работать с ТТЛ- и МОП-устройствами, соответствующие значения токов имеют значения от 100 мА до менее 50 мА. Пульсатор для обслуживания КМОП-схем должен иметь возможность работать с напряжениями до 18 В, хотя подобные уровни маловероятны в микропроцессорных системах.

Предполагается, что при импульсном тестировании устройств их внутренние тактовые генераторы не работают либо вследствие неисправностей, причины которых подлежат выяснению, либо будучи принудительно отключены, с тем чтобы была уверенность, что наблюдению подлежат только управляемые тестовые импульсные сигналы.

Логические клипсы

Это многоконтактные контрольные устройства, которые устанавливаются на ИС и позволяют наглядно отображать состояние внешних выводов с помощью набора светодиодов. Они снаб-

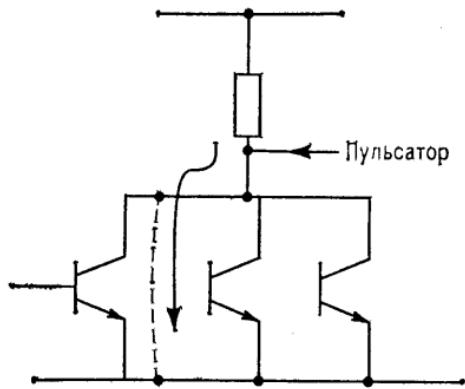


Рис. 1.7. Проверка схемы «монтажное ИЛИ» с помощью токового пульсатора и токового пробника.

¹⁾ Говоря точнее, схема, изображенная на рис. 1.7, представляет собой объединение «монтажным ИЛИ» выходов нескольких инверторов, т. е. реализует функцию ИЛИ-НЕ. — Прим. перев.

жены чувствительными входными схемами, способными воспринимать потенциалы, которые имеют место в контролируемых ИС и по понятным причинам не могут превышать напряжения питания, и использовать их для обеспечения работы внутренних схем и индикаторов клипов.

Если рабочая частота тактовых сигналов, поступающих на контролируемую ИС, достаточно низка, функционирование ИС можно наблюдать в динамике. Логические клипы очень удобны для обслуживания логических устройств, однако для работы с микропроцессорами они применимы в меньшей степени, так как число их контактов обычно не превышает 14 или 16. Их можно было бы использовать для наблюдения за работой 16-контактных ИС вспомогательного назначения, но получить с их помощью существенную информацию о работе, например, мультиплексируемых шин не удается, даже если установить клипсы на эти ИС.

Аналогичные недостатки присущи логическим компараторам. Логический компаратор служит для сопоставления сигналов тестируемой и эталонной интегральной схем, на которые входные сигналы поступают параллельно. С помощью компаратора можно выявить даже отклонения в протекании переходных процессов. Маловероятно, что компараторы будут широко использоваться при работе с микропроцессорными системами, для обслуживания которых разработаны средства логического и сигнатурного анализа, внутрисхемной эмуляции и т. д. Существенно новыми являются сами принципы обслуживания микропроцессорных систем по сравнению с традиционными устройствами на интегральных схемах. Все методы тестирования, рассматриваемые в данной главе, ориентированы на контроль отдельных ИС. Они являются, конечно, шагом вперед по сравнению с контролем на уровне схемных компонентов, но не отражают мышления, ориентированного на обслуживание систем сшинной организацией. Эти методы будут по-прежнему применяться для контроля плат интерфейса периферийных устройств, а также в ситуациях, когда посредством других тестов удается локализовать неисправность с точностью до ограниченной физической области, а уже пользуясь этими методами, выявить место отказа вплоть до визуального наблюдения.

Схемотехника логических ИС

Отказы могут происходить в отдельных логических элементах и цифровых схемах, расположенных в периферийных частях микропроцессорных систем. Проявляются они по-разному в зависимости от конкретного вида используемой схемотехники. Ниже приводятся краткое описание основных типов логических

схем и их характеристики, не заменяющие, однако, изучения паспортных данных, которыми снабжают ИС изготовители.

Самыми распространенными видами схемотехники являются следующие:

транзисторно-транзисторная логика (ТТЛ);

логика с эмиттерными связями (ЭСЛ);

инжекционная интегральная логика (I^2L);

структуры металл—окисел — полупроводник *p*-типа (*p*МОП);

структуры металл—окисел—полупроводник *n*-типа (*n*МОП);

комплементарные МОП-структуры (КМОП);

динамические МОП-структуры.

(В периферийных устройствах широко применяются ТТЛ- и КМОП-логика; их построение требует более глубокого изучения.)

Более старые виды схемотехники, такие, как резистивно-транзисторная логика (РТЛ) и диодно-транзисторная логика (ТЛ), в микропроцессорных системах практически не применяются.

У каждого из указанных видов схемотехники есть свои преимущества. Так, ЭСЛ отличается высоким быстродействием, хотя некоторые разновидности ТТЛ приближаются к ней по этому параметру. Как *p*МОП-, так и затем МОП-логика широко применяются в микропроцессорах, а КМОП-схемотехника пользуется предпочтением, когда важно уменьшение потребляемой мощности. Динамические МОП-структуры используются для построения различных запоминающих устройств; они имеют простую организацию, в которой логическое состояние определяется зарядом емкости, внутренне присущей логическому элементу. I^2L применяется в интегральных схемах.

ТТЛ

Этот вид схемотехники имеет, пожалуй, наибольшее число разновидностей, так как длительность его использования позволила выявить многие из присущих ему ограничений и способы их преодоления. Типовой инвертор, схема которого приведена на рис. 1.8, входит в состав ИС, конструктивно оформленной в 14-контактном корпусе типа DIP (dual in line package — корпус с двухрядным расположением выводов). В ИС размещаются четыре таких инвертора, что позволяет рассматривать такую ИС как «шестнадцатиальный инвертор». Максимальное напряжение на базе транзистора T1 составляет около 2 В, оно представляет собой сумму падений напряжения на трех *pn*-переходах, смешенных в прямом направлении (базово-коллекторном переходе T1 и базово-эмиттерных переходах T2 и T3). Когда потенциал

эмиттера T1 поднимается до значения, отличающегося от этого напряжения меньше чем на 0,6 В, эмиттерный ток падает до нуля. Весь ток, протекающий через резистор 4 кОм, проходит при этом через коллектор T1 и базу T2, открывая транзисторы T2 и T3. При этом напряжение на базе T4 опускается ниже 1 В (суммы коллекторно-эмиттерного напряжения насыщенного T2 и базово-эмиттерного напряжения T3), что делает невозможным пребывание во включенном состоянии T4 из-за наличия диода в его эмиттерной цепи. При открытом T3 и запертом T4 выходной

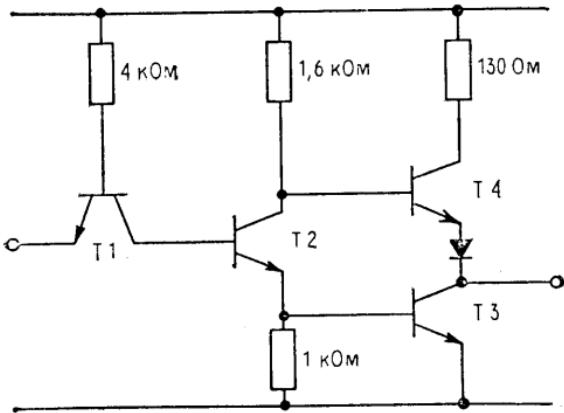


Рис. 1.8. Типовой ТТЛ-инвертор.

сигнал имеет значение логического 0. Типичной является ситуация, когда T3 пропускает ток до 16 мА без превышения выходным напряжением уровня 0,4 В. Важность этого обстоятельства будет показана ниже.

Вследствие того что входные цепи таких элементов имеют распределенные индуктивность и емкость, при подаче на них высокочастотных импульсных сигналов могут возникнуть «дребезги» с существенными отрицательными выбросами. Обычно в состав ТТЛ-элемента включаются диоды, соединяющие каждый вход с общей шиной, как показано на рис. 1.9. Любой отрицательный бросок входного напряжения ограничивается величиной около $-0,7$ В, что исключает возможные повреждения или некорректное функционирование схемы вследствие таких бросков.

Выходная цепь типового ТТЛ-элемента, часто называемая двухтактным выходом, может переключаться в обоих направлениях с почти одинаковой скоростью. Параллельное включение выходов сложных инверторов небезопасно, так как, если инверторы пребывают в противоположных логических состояниях, через открытые транзисторы T4 одного из них и T3 другого про текает недопустимо большой ток, который может вызвать отказ

схемы. В некоторых ситуациях желательно параллельное объединение выходов логических элементов с целью реализации функции «монтажное ИЛИ».

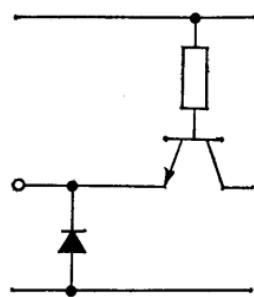


Рис. 1.9. Защита входной цепи с помощью диода.

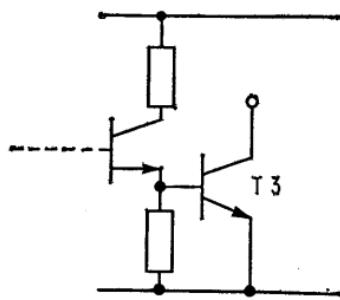


Рис. 1.10. Выход с открытым коллектором.

В схеме на рис. 1.10 выходной каскад элемента содержит только транзистор Т3, и такие выходы могут быть запараллелены в любом количестве. Когда хотя бы один из транзисторов Т3 открыт, выходной сигнал имеет значение логического 0. Обычно между общей точкой коллекторов транзисторов Т3 и положительной шиной питания включается резистивная или другая пассивная нагрузка. Скорость переключения элемента в этом случае ниже, чем при использовании сложного инвертора, особенно при переходе от логического 0 к логической 1.

Рассмотрим еще один вариант построения ТТЛ-элемента, характеризующийся повышенным быстродействием. Во-первых, в нем уменьшены величины сопротивлений, чтобы увеличить значения токов и тем самым ускорить заряд внутренних емкостей. Во-вторых, выходной транзистор Т4 заменен на дарлингтоновскую пару с целью увеличения тока, потребляемого нагрузкой, а также идущего на заряд паразитных емкостей (рис. 1.11). Противоположный прием использован в маломощных ТТЛ-схемах, в которых номинальные значения резисторов увеличены, в результате чего снижается быстродействие, но зато и уменьшается потребление мощности. Эти разновидности построения элементов

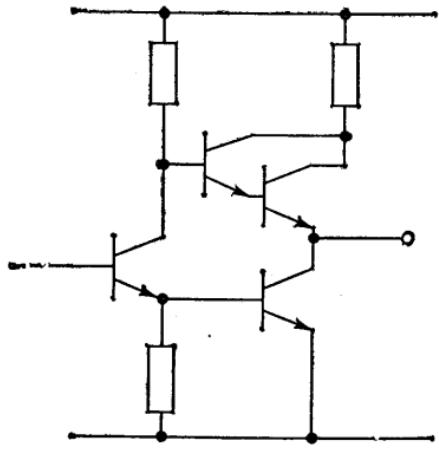


Рис. 1.11. Дарлингтоновский выходной каскад мощного ТТЛ-элемента.

можно применять в сочетании с таким расчетом, чтобы в частях устройств, критичных по быстродействию, использовались схемы высокоскоростные, но потребляющие повышенную мощность. Совместное применение разнотипных элементов выдвигает проблему, заключающуюся в том, что выходы маломощных схем не могут нагружаться множеством быстродействующих вентилей с присущими им повышенными требованиями к входному току. Каждая разновидность элемента характеризуется вполне определенной нагрузочной способностью, т. е. максимальным количеством стандартных входов, которыми он может управлять.

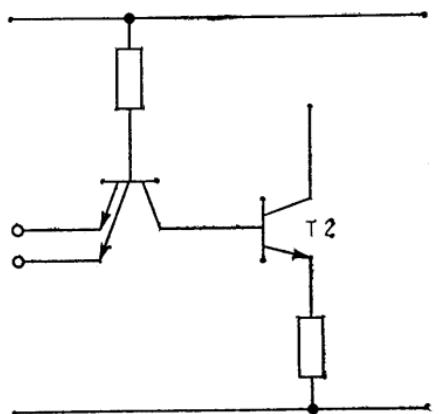


Рис. 1.12. Многоэмиттерный вход логического элемента.

Обычно эта величина порядка 10, но маломощные ТТЛ-элементы, потребление энергии которых составляет десятую долю типового значения, имеют во столько же раз меньшую нагрузочную способность по отношению к типовым элементам. Маломощный ТТЛ-элемент лишь на пределе своих возможностей может управлять высокоскоростным ТТЛ-элементом. В таких ситуациях необходимы специальные средства. Следует подчеркнуть, что, как показано на рис. 1.12, конфигурация схем в интегральном исполнении может существенно отличаться от общепринятой для схем с дискретными компонентами. Функции И/ИЛИ реализуются на основе многоэмиттерных транзисторов.

При подаче логического 0 на любой из эмиттеров такого транзистора входной ток T2 направляется через этот эмиттер и T2 выключается. Аналогично в И²Л-схемах используются многоколлекторные структуры. С целью повышения быстродействия ТТЛ-элементов принимают меры для предотвращения насыщения входящих в них транзисторов. Насыщение выражается в том, что транзистор приводится в открытое состояние избыточным входным током, который смещает в прямом направлении базово-эмиттерный переход и приводит к накоплению заряда на этом переходе. После прекращения поступления входного тока накопленный на базово-эмиттерном переходе заряд оставляет транзистор открытым в течение десятков наносекунд. В элементах, построенных по другим схемотехническим принципам, например в ЭСЛ, токи после протекания через одни цепи направляются в другие без необходимости преодоления насыщения каких-либо транзисторов. Задержки при этом уменьшаются на

порядок, но зато проявляются такие недостатки, как относительно высокая статическая потребляемая мощность и меньшая помехоустойчивость (т. е. разрыв между напряжением на вы-

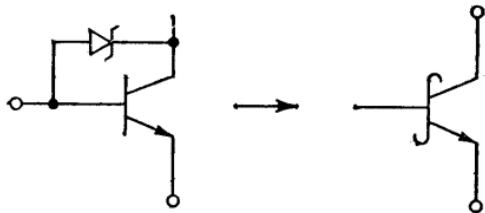


Рис. 1.13. Транзистор с ограничительным диодом Шоттки, предотвращающим насыщение.

ходе и тем критическим уровнем напряжения, который на входе следующего логического элемента может быть уже не распознан в качестве корректного уровня).

Введение в схему элемента специального диода, называемого диодом Шоттки, иллюстрируется рис. 1.13. Это переход металл—полупроводник, который может быть автоматически сформирован за счет небольшой модификации процесса изготовления ИС.

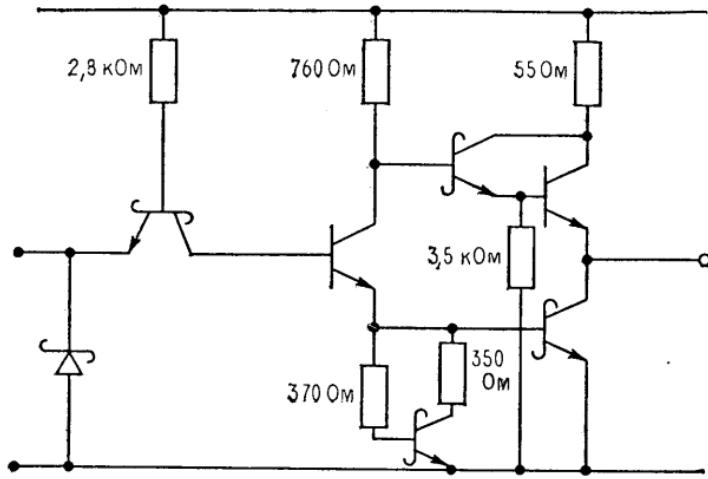


Рис. 1.14. ТТЛ-элемент повышенного быстродействия на транзисторах Шоттки.

Падение напряжения на нем при прямом смещении составляет около 0,4 В, т. е. существенно меньше, чем на кремниевом *р-п*-переходе. При поступлении в базу транзистора тока от какого-либо источника потенциал базы поднимается до 0,7 В и коллекторная проводимость увеличивается. Через сопротивление коллекторной нагрузки протекает большой ток и напряжение на коллекторе падает до уровня насыщения. Как только напряжение на коллекторе опускается примерно до 0,4 В, диод Шоттки оказывается смещенным в прямом направлении и начинает проводить ток. При этом избыточный базовый ток ответвляется в

коллекторную цепь и накопление заряда на базово-эмиттерном переходе сводится к минимуму. Таким образом, конфигурация схемы не претерпевает изменений, логические уровни по существу сохраняются, но транзисторы, снабженные диодами Шоттки, уже не переходят в состояние насыщения. Быстродействие схемы возрастает без необходимости увеличения потребляемой мощности (или эта мощность может быть снижена при сохранении имеющегося быстродействия). Типовая схема с диодами Шоттки, представленная на рис. 1.14, в функциональном отношении эквивалентна схеме, приведенной на рис. 1.8 и являющейся частью шестнадцатеричного инвертора.

Характеристики некоторых видов логических элементов класса ТТЛ приведены в табл. 1.1. Из рис. 1.15, а видно, каким образом оцениваются уровни токов. Если вход ТТЛ-элемента

Таблица 1.1. Характеристики ТТЛ-элементов

	Разновидности ТТЛ			
	Стандарт- ная	Мало- мощная	Быстродей- ствующая	ТТЛ Шоттки
Напряжение питания, В	5	5	5	5
Ток логического 0, мА	5,5	0,46	10	9
Ток логической 1, мА	1,3	0,18	4,2	4,25
Нагрузочная способность	10	10	10	10
Потребляемая мощность, мВт	10	1	22	20
Задержка выходного сигнала при смене 1 на 0, нс	8	30	6	3
Задержка выходного сигнала при смене 0 на 1, нс	12	30	6	3
Помехоустойчивость в состоянии логического 0, мВ	400	400	400	700
Помехоустойчивость в состоянии логической 1, мВ	400	400	400	300
Максимальное входное напряжение при логическом 0, мВ	800	700	800	800
Минимальное входное напряжение при логической 1, В	2	2	2	2
Максимальное выходное напряжение при логическом 0, мВ	400	300	400	500
Минимальное выходное напряжение при логической 1, В	2,4	2,4	2,4	2,7
Входной ток логического 0, мА	1,6	0,18	2,0	2,0
Входной ток логической 1, мкА	40	10	50	100
Выходной ток логического 0, мА	16	2	20	20

заземлен, то при напряжении питания 5 В падение напряжения на входном переходе составляет примерно 0,7 В, а на сопротивлении — около 4,3 В. Номинальная величина сопротивления в первом варианте схемы 4 кОм, что определяет ток в пределах 1,1 мА. Эта величина снижается, если входной логический 0 по-

ступает с предыдущего ТТЛ-каскада и может представляться в виде уровня, который зависит от числа управляемых входов и

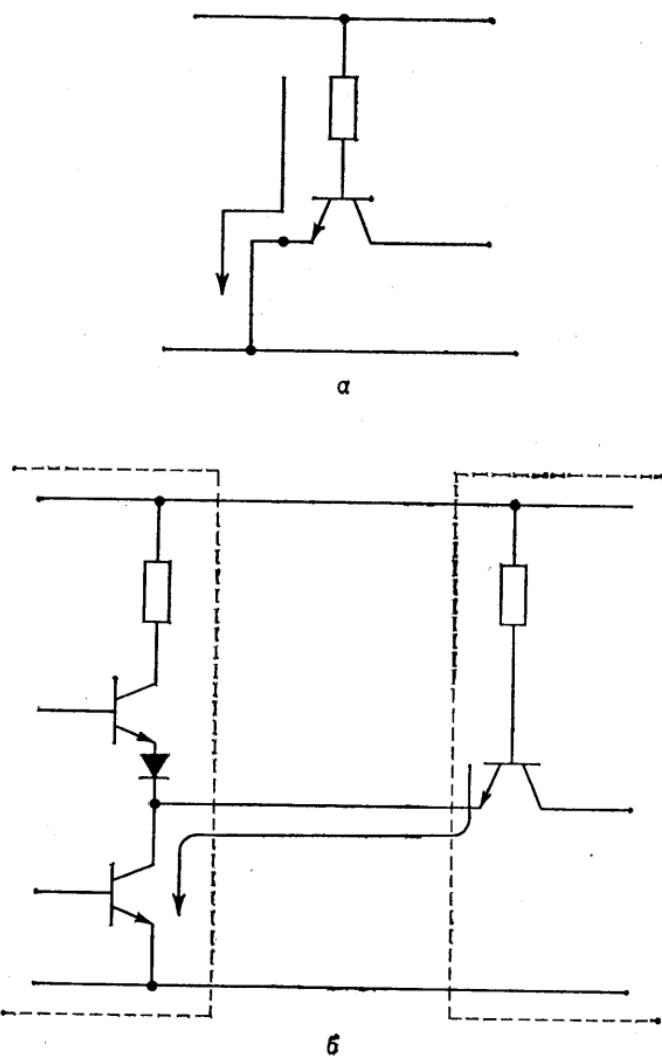


Рис. 1.15. а — входной ток в состоянии логического 0, $I \sim [(5 \text{ В} - 0,7 \text{ В}) / 4 \text{ кОм}] \text{ мА}$; б — протекание втекающего выходного тока при логическом 0 на выходе.

может достигать 0,4 В. Поэтому вполне можно считать, что входной ток логического 0 имеет величину около 1 мА. Допуски на значения сопротивлений обычно весьма широки, что диктуется соображениями максимального увеличения выхода годных ИС¹⁾ и соответствующего снижения их цен. С запасом на возможный

¹⁾ Под «выходом годных» принято понимать долю работоспособных ИС в общем числе изготовленных ИС.— Прим. перев.

разброс сопротивлений изготовители обычно указывают, что втекающий ток от каждого входа, управляемого данным выходом, может достигать 1,6 мА. Входные токи для других видов ТТЛ-схем могут различаться более чем в десять раз. Когда на вход поступает логическая 1, входной ток должен быть пренебрежимо мал, так как входной переход смещен в обратном направлении. Худший случай будет предусмотрен, если считать, что при логической 1 на входе имеет место ток утечки величиной 40 мкА, а обеспечение такого тока с гарантией находится в пределах возможностей выходного каскада любого логического элемента.

Значения выходных параметров можно оценить с помощью схемы на рис. 1.16. При логическом 0 в открытом состоянии находится только один транзистор, причем это состояние — с насыщением в обычном ТТЛ-элементе и без насыщения в схеме ТТЛ-Шоттки. Для логического 0 указывается максимальный уровень 0,4 В. В состоянии логической 1 работают еще два *pn*-перехода: базово-эмиттерный переход транзистора T4 и диод в его эмиттерной цепи. Если ток нагрузки велик, то результирующий ток

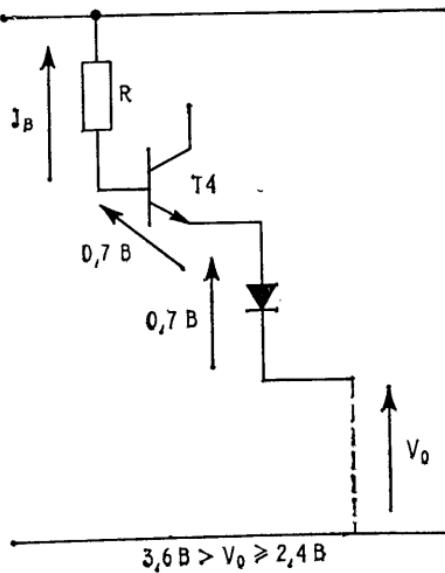


Рис. 1.16. Снижение уровня выходного сигнала из-за падения напряжения на компонентах схемы.

базы T_4 определяет значительное падение напряжения на базовом сопротивлении R . Величины напряжений, приведенные на рис. 1.16, показывают, что падение напряжения на этом сопротивлении может достигать 1,2 В, когда при большой нагрузке уровень выходного сигнала снижается до 2,4 В. (При отсутствии нагрузки он может подниматься до 3,6 В.)

ТТЛ-элемент должен реагировать на все напряжения, превышающие 2,4 В, как на логическую 1, даже при наихудшем сочетании температуры, нагрузки и других условий функционирования. Тем самым определяется запас надежности $(2,4 - 2,0) \text{ В} = 0,4 \text{ В}$, обычно называемый помехоустойчивостью. Никакая импульсная помеха, меньшая чем 0,4 В, поступив на вход ТТЛ-элемента, не будет воспринята как логический 0. На практике статический выходной сигнал может значительно превышать 2,4 В, а минимальный входной сигнал, воспринимаемый как логическая 1, может быть значительно меньше 2,0 В, в результате

чего практическая помехоустойчивость существенно выше теоретической. Однако это обстоятельство не является гарантированным и не может быть принято в расчет при проектировании.

Выходное напряжение логического 0 не должно превышать 0,4 В. В то же время входы нагружаемых на него элементов воспринимают любое напряжение, меньшее 0,8 В, как достоверный логический 0. Таким образом, и в этом случае запас помехоустойчивости составляет 0,4 В.

Импульсные характеристики: переходные процессы

Сказанное выше достаточно определенно указывает, каким образом можно контролировать параметры логических элементов в их статическом состоянии. Когда уровни логических 0 и 1 имеют близкие к предельным значения, они крайне чувствительны к переходным процессам вне зависимости от источника происхождения этих процессов. К сожалению, сами ТТЛ-элементы могут быть причиной недостоверных переходных состояний схемы. Ранее мы предполагали, что в любой момент времени открыт лишь один из транзисторов двухтактного выходного каскада. Это допущение несправедливо в процессе смены логических значений. При логическом 0 на выходе нижний транзистор выходного каскада находится в состоянии насыщения. В процессе перехода к выходному сигналу 1 сначала начинает проводить верхний транзистор. Емкостный заряд базы нижнего транзистора удерживает его в открытом состоянии примерно 10 нс, и в течение этого интервала времени оба транзистора открыты, потребляя, хотя и не долго, большой ток от источника питания.

Даже если цепи основного источника питания должны образом развязаны, реактивные составляющие распределенного сопротивления печатных проводников отвечают на этот токовый импульс генерацией колебаний напряжения питания. В соответствии с общепринятой практикой на печатных платах устанавливаются развязывающие конденсаторы с таким расчетом, чтобы ни одна ИС не находилась далее нескольких сантиметров от какого-либо из этих конденсаторов. Емкость конденсаторов выбирается так, чтобы на каждый двухтактный выход приходилось примерно 2000 пФ, т. е. 100 нФ на плату, на которой размещено десять корпусов, содержащих в общей сложности 50 логических элементов. Еще один большой tantalовый конденсатор с номинальным значением емкости примерно 10 мкФ может быть установлен между проводниками питающего напряжения в месте подачи питания на плату. Для повышения эффективности этих конденсаторов они должны хорошо работать в широком диапазоне частот и обладать небольшой самоиндуктивностью (как, например, конденсаторы на керамических дисках), иметь

короткие установочные выводы и быть установлены как можно ближе к ИС, которые они обслуживают.

Медленные переходные процессы

В большинстве случаев значения входных сигналов логических элементов изменяются очень быстро, и выходной сигнал сменяется при этом с максимальной скоростью. Однако в некоторых случаях, когда входные сигналы поступают от аналоговых устройств, их изменения могут продолжаться более микросекунды. При этом в течение короткого интервала времени ТТЛ-схема работает в линейном режиме, т. е. выполняет функции усилителя напряжения, преобразуя каждый милливольт изменения входного напряжения в изменение выходного на несколько милливольт. При наличии паразитных связей и фазовых сдвигов на очень высоких частотах вполне возможно возникновение нестабильности в работе схемы. На выходе вместо плавного перехода может иметь место «дребезг», при котором происходят многократные переходы через пороговые уровни переключения последующих логических элементов. Для преодоления последствий подобных колебательных процессов вводится буферный каскад с внутренней регенеративной обратной связью, обеспечивающей формирование ступенчатого сигнала. Медленное изменение входного сигнала буферным каскадом не воспринимается до тех пор, пока этот сигнал не достигает некоторого критического уровня, при котором выходной сигнал буфера скачком меняет свое логическое значение. Буферный каскад обладает гистерезисными свойствами, благодаря которым его выходной сигнал не меняет своего значения в обратном направлении, пока входной не вернется к значению, значительно меньшему, чем то, при котором произошло переключение выходного. Тем самым реализуется защита от ложных переключений, которые могли бы возникать из-за дребезга на входе.

ТТЛ-элемент с тремя состояниями

В системах с шинной организацией к одной общей линии могут быть подключены выходы множества логических элементов, но в любой момент времени только один из них должен пребывать в активном состоянии, т. е. иметь возможность управлять линией. Этот принцип широко используется в микропроцессорных системах. На рис. 1.17 приведена простая схема реализации ТТЛ-элемента, в которой через вход транзистора T5 осуществляется разрешение или запрещение выходного сигнала. Когда на этом входе логический 0, транзистор T5 заперт и ТТЛ-элемент работает как обычно. При подаче на вход T5 логической 1

он открывается и потенциал базы T4 приближается к потенциальному общей шины, в результате чего последний запирается. Одновременно приобретает потенциал логического 0 коллектор T2, и через базово-коллекторный переход этого транзистора шунтирует на общую шину весь управляющий ток транзистора T3 (включая его составляющую, поступающую от T1). Таким образом, оба выходных транзистора T3 и T4 оказываются запертыми, т. е. выход логического элемента пребывает фактически

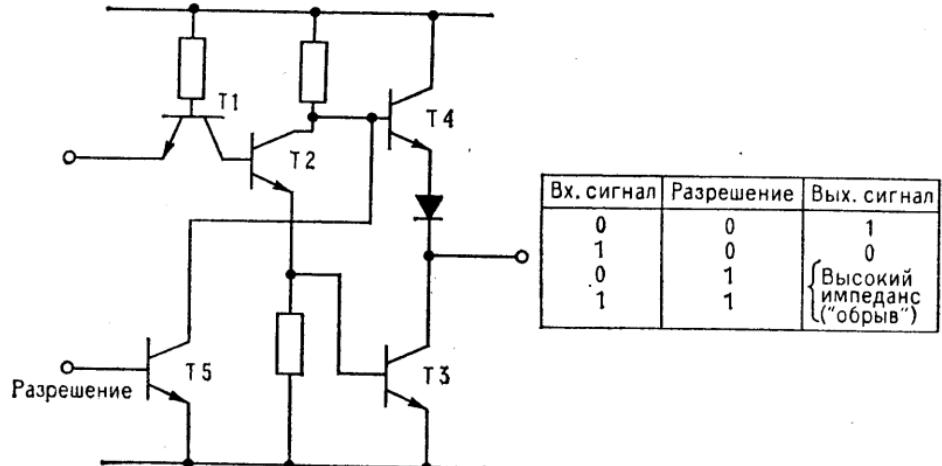


Рис. 1.17. Элемент ТТЛ с тремя состояниями выходного сигнала.

в состоянии обрыва по отношению ко всем другим точкам схемы. Через него может поступать только пренебрежимо малый ток утечки. Типичная неисправность элемента заключается в том, что один из транзисторов выходного каскада оказывается в проводящем состоянии, вследствие чего линия, с которой соединен этот выход, приобретает уровень логического 0 или 1. Когда линия шины постоянно имеет тот или иной логический уровень, с помощью токового пробника обычно можно выявить точку, в которой происходит утечка тока, т. е. обнаружить неисправный логический элемент.

Отыскание неисправностей

Следует рекомендовать следующие простые приемы проверки схемы:

1. Контроль напряжения питания в различных точках платы как в статических, так и в переходных режимах (надо при этом иметь уверенность, что применяемый осциллограф имеет достаточно широкую полосу частот; полоса пропускания 50 кГц может оказаться недостаточной для выявления высокочастотных импульсных сигналов).

2. Проверка уровней напряжения логических сигналов, которые могут сказываться на помехоустойчивости схемы; эта проверка возможна, если все логические вентили могут быть приведены в статическое состояние.

3. Проверка уровней логических сигналов в динамических режимах с помощью тщательно откалиброванного осциллографа.

Логика с эмиттерными связями [ЭСЛ]

Интегральные схемы этого типа, как правило, применяются только в быстродействующих устройствах системы. Эмиттеры транзисторов, на базы которых подаются входные сигналы A, B и C (рис. 1.18), объединены между собой, а также с эмиттером опорного транзистора. С помощью сопротивления 1,2 кОм ток

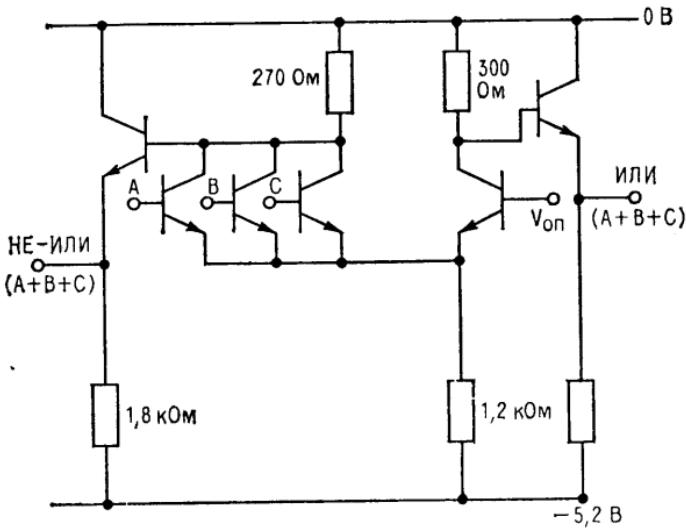


Рис. 1.18. Логический элемент типа ЭСЛ.

в общей эмиттерной цепи поддерживается в достаточной мере постоянным, но может быть направлен в ту или другую часть схемы в зависимости от того, имеют ли все сигналы, А, В и С, уровень ниже V_{op} . Если уровень какого-то из этих сигналов или их всех становится больше, чем V_{op} , весь ток направляется через сопротивление 270 Ом. Электрические уровни логических сигналов выглядят не вполне привычно: для логического 0 — около $-1,7$ В, для логической 1 — примерно $-0,8$ В. Разница между уровнями невелика; это обстоятельство определяет относительно высокое быстродействие данного вида схемотехники, достигаемое, впрочем, за счет снижения помехоустойчивости. Кроме того,

скорость переключения логического элемента в принципе высока и в схеме отсутствуют токовые переходные процессы, так как ни один транзистор не переходит в состояние насыщения и при переключениях элемента происходит лишь перераспределение токов.

Элемент ЭСЛ обладает высокой нагрузочной способностью (25), потому что каждая подключаемая база представляет собой небольшую нагрузку для эмиттерного повторителя. Схема обеспечивает получение парафазного выходного сигнала, что делает ненужным использование инверторов. Для сопряжения ЭСЛ-схем с ТТЛ- и КМОП-схемами требуются специальные преобразователи уровней. Типичный ЭСЛ-элемент может работать с тактовой частотой, почти в два раза превышающей возможности даже элемента ТТЛ-Шоттки (200 вместо 125 мГц).

Интегральная инжекционная логика (И²Л)

В этом виде схемотехники инвертор упрощен до минимума: это транзистор с несколькими открытыми коллекторами (рис. 1.19). Не требуется никаких внутренних сопротивлений, что радикально сказывается на плотности компоновки ИС, так как

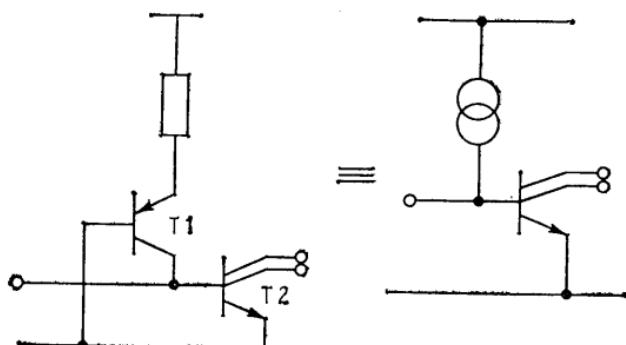


Рис. 1.19. Логический элемент типа И²Л.

каждый резистор занимает площадь, раз в десять большую, чем транзистор. Управляющий ток базы формируется с помощью транзистора T1 и внешнего сопротивления. Транзистор T2 с открытыми коллекторами, связанный с последующими логическими элементами, пропускает этот ток на общую шину, когда находится в открытом состоянии. Перепады напряжений малы, что в сочетании с возможностью варьирования статических токов позволяет достичь соотношений быстродействия и мощности, делающих схемотехнику И²Л конкурентоспособной с ТТЛ- и МОП-схемотехникой.

nМОП/pМОП/кМОП-структуры

Организация *nМОП*- и *pМОП*-логических элементов аналогична, на рис. 1.20 и 1.21 она показана на примере *nМОП*-структур. В схеме на рис. 1.21, а все МОП-транзисторы работают в режиме обогащения, при котором затвор должен быть смещен в прямом направлении по отношению к истоку, чтобы транзистор открылся. Это удобно с точки зрения функционирования активных транзисторов T_1 и T_2 , но не вполне рационально для транзистора T_3 , выполняющего роль пассивной нагрузки. Чтобы эта нагрузка пропускала ток, затвор нагрузочного транзистора соединен с истоком. К сожалению, ток нагрузки при таком включении существенно изменяется в зависимости от уровня выходного сигнала элемента, который влияет на потенциал затвора и истока. Этот серьезный недостаток схем, в которых все МОП-транзисторы работают в режиме обогащения, в значительной

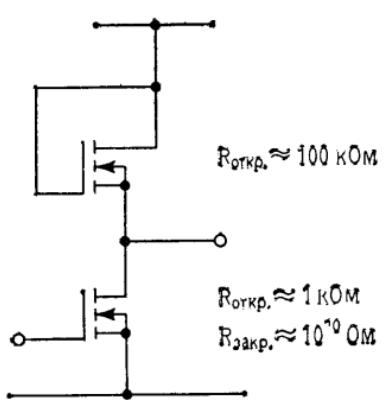


Рис. 1.20. Использование транзистора в качестве нагрузочного сопротивления логического элемента.

ногого сигнала элемента, который влияет на потенциал затвора и истока. Этот серьезный недостаток схем, в которых все МОП-транзисторы работают в режиме обогащения, в значительной

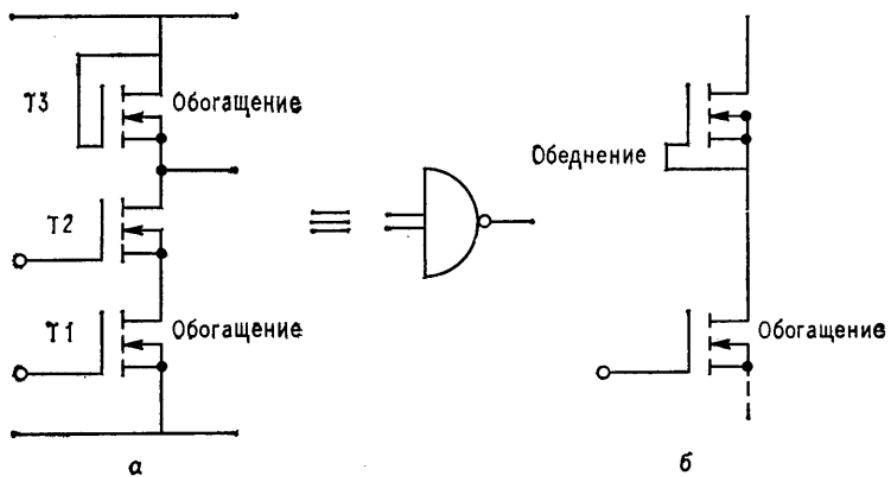


Рис. 1.21. а — логический элемент И-НЕ на *nМОП*-структурах; б — совместное использование МОП-транзисторов с обогащением и с обеднением.

степени сводит на нет их преимущество, заключающееся в простоте изготовления ИС на приборах одного типа. Сказанное в полной мере относится и к схемам, целиком реализованным на транзисторах, работающих в режиме обеднения. Канал МОП-

транзистора с обеднением открывается, когда затвор смешен по отношению к истоку в обратном направлении; это означает, что при положительном питании стоков необходимо иметь отдельный источник отрицательного питания для затворов.

В идеальном случае активные транзисторы должны работать в режиме обогащения, а нагрузочный транзистор — в режиме обеднения, как показано на рис. 1.21, б. Процесс изготовления ИС при этом усложняется, так как связан с необходимостью более точного дозирования примесей для формирования приборов двух типов, но зато улучшается работа логических схем.

МОП-структуры *n*-типа обладают рядом преимуществ по сравнению с *p*-канальными. Они допускают большую плотность компоновки и имеют более высокое быстродействие, так как электроны, являющиеся носителями тока в *n*-структуратах, обладают большей подвижностью, чем дырки, выполняющие эту роль в *p*-канальных транзисторах.

В инверторах в качестве нагрузки могут быть использованы транзисторы с обогащением или с обеднением. Более предпочтительны приборы, работающие в режиме обеднения, так как они обладают свойством автоматического ограничения тока при превышении потенциалом затвора-истока некоторого порога, называемого уровнем смыкания канала.

Видное место занимает еще одна разновидность МОП-структур — так называемая КМОП-логика. На первых этапах ее развития комплементарными МОП-структурами активно занималась только фирма RCA, которая первой их изготовила. В настоящее время большинство изготовителей ИС включает КМОП-схемы в состав своей продукции, а в изделиях некоторых компаний КМОП-схемотехника является доминирующей.

Известно, что КМОП-логика имеет множество потенциальных преимуществ. Ее относительно высокая стоимость рассматривалась как вполне приемлемая для тех областей применения, где особенно важны малое потребление мощности и высокая помехоустойчивость. Постепенно по мере расширения объемов производства и совершенствования технологии различия в стоимости КМОП-ИС и обычных МОП-ИС стерлись и КМОП-ИС стали занимать все большее место. В состав КМОП-элемента входят *n*-канальные и *p*-канальные МОП-транзисторы, работающие в режиме обогащения. Схема КМОП-инвертора приведена на рис. 1.22, а. Диодная схема защиты входной цепи (рис. 1.22, б) формируется автоматически в процессе изготовления ИС. Тем самым исключается проблема, характерная для первых КМОП-структур, в которых статические заряды в сочетании с присущим КМОП-схемам высоким входным импедансом могли привести к образованию потенциалов, достаточных для повреждения тонкой изолирующей области, наличие которой принципиально

важно для работы структуры. Диоды демпфируют эти потенциалы, а также напряжения, возникающие на линиях питания при переходных процессах. Наличие защитных диодов никак не отражается на функционировании КМОП-элемента, поэтому их не показывают на функциональных схемах устройств. При отсутствии неисправностей в КМОП-устройствах анализ их работы с помощью осциллографа позволяет наблюдать последовательности импульсов с амплитудой, почти достигающей 5 В; всякое уменьшение этой амплитуды дает повод для беспокойства и поиска причин отклонения.

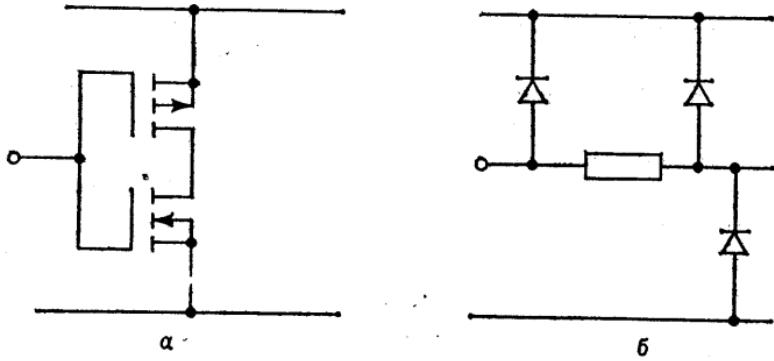


Рис. 1.22. а — инвертор на комплементарных МОП-транзисторах; б — диодная схема защиты входа.

Высокий входной импеданс КМОП-структур может вызвать затруднения при наличии длинных проводников, по которым поступают входные сигналы. Начала и концы таких длинных линий должны быть снабжены оконечными схемами, соответствующими естественному импедансу линий (здесь применимы положения теории линий передачи). При низком управляющем импедансе и высоком импедансе нагрузки возможно возникновение как на переднем, так и на заднем фронте каждого импульса колебательных переходных процессов, сравнимых по амплитуде с самим импульсом. Предохранительные диоды препятствуют повреждению МОП-структур, но при этом в крайних ситуациях одиночный импульс может быть воспринят как множество импульсов. Разработчику следует воспользоваться преимуществом, каким является высокий входной импеданс КМОП-структур, предусмотрев резистивные оконечные схемы с соответствующим сопротивлением, являющимся пренебрежимо малой дополнительной нагрузкой.

Если превышены ограничения по нагрузочной способности, амплитуда выходных сигналов становится меньше 5 В, так как полевые транзисторы входят в режим ограничения тока. После прохождения этого уровня размах напряжения падает пропор-

ционально сопротивлению нагрузки. Предположим, что при некотором сопротивлении нагрузки амплитуда выходного напряжения уменьшилась до величины 4 В, при которой все КМОП-элементы, нагруженные на данный выход, будут по-прежнему правильно реагировать на выходной сигнал. Снижение сопротивления нагрузки, например, на 30 % приведет примерно к такому же уменьшению амплитуды выходного сигнала. Результирующей амплитуды, меньшей 3 В, уже может оказаться недостаточно для надежного управления последующими входами. Для повышения нагрузочной способности КМОП-элемента можно на его выходе установить ТТЛ-буфер, причем для таких целей обычно используется разновидность ТТЛ-элементов с пониженным потреблением мощности.

Неправильно спроектированные КМОП-схемы могут стать причиной и других видов некорректного функционирования. Если входы неиспользуемых логических элементов не подключить к какой-либо из линий питания, их высокие входные импедансы в сочетании с паразитными параметрами схемы могут стать причиной генерации высокочастотных колебаний, которые, проходя через паразитные связи, могут искажать входные сигналы других элементов.

Эти особенности присущи и другим типам цифровых устройств на МОП-структуратах. Их, как правило, не применяют во входных и выходных каскадах систем. Наибольшее распространение МОП-структуры получили при проектировании запоминающих устройств с произвольной выборкой (ЗУПВ), постоянных запоминающих устройств (ПЗУ), электрически программируемых ПЗУ (ЭППЗУ), в которых они отделены буферными схемами от внешних устройств. Поиск неисправностей в этом случае осуществляется не на уровне отдельных ИС, а на уровне платы с использованием системных соединительных шин.

Быстродействующие КМОП-ИС

Ряд крупных фирм — изготовителей полупроводниковых приборов приступил относительно недавно к производству быстродействующих КМОП-ИС¹⁾, входящих в группу серий 74' логических интегральных схем. Они совместимы по входам и выходам со стандартными ТТЛ-ИС и с КМОП-ИС серии В.

¹⁾ ИС, относящиеся к другим видам схемотехники, в том числе ЭСЛ и И²Л, занимают на рынке полупроводниковых приборов свои собственные места; так, например, ЭСЛ-ИС применяются в быстродействующих микропроцессорных системах с разрядно-модульной организацией. Вследствие их высокого быстродействия необходимо применение специальных методов тестирования, для реализации которых обычно лабораторные осциллографы оказываются непригодны. Контроль логических уровней осуществляется достаточно просто, однако измерение малых перепадов требует тщательной ка-

Быстродействующие КМОП-структуры обладают всеми достоинствами обычных КМОП-вентилей и, кроме того, некоторыми дополнительными преимуществами, которые приближают их по характеристикам к стандартным ТТЛ-ИС. В качестве этих дополнительных преимуществ укажем следующие:

- уменьшенная задержка распространения сигнала (около 10 нс для серии 7400);
- пониженное рассеяние мощности (сохраняется присущая КМОП-схемам зависимость мощности от рабочей частоты, но можно считать, что на частоте 1 мГц мощность рассеяния быстродействующих КМОП-элементов примерно такая же, как у стандартных ТТЛ-схем);
- выходные токи истока и стока порядка 4 мА.

Быстродействующие КМОП-элементы могут иметь как обычные выходы, так и выходы с тремя состояниями; некоторые из таких элементов без буферизации используются для построения схем генераторов.

При нормальных уровнях логических сигналов в любой момент времени открыт только один из выходных транзисторов КМОП-схемы. Когда на входе логический 0, в проводящем состоянии пребывает верхний транзистор, так как напряжение затвор — исток большое. В то же время напряжение затвор — исток нижнего транзистора равно нулю и он не пропускает тока. Выходной сигнал элемента имеет значение логической 1; через выходные транзисторы протекает пренебрежимо малый ток покоя. Существенно, что при этом уровень выходного сигнала очень близок к напряжению источника питания и имеется очень большой запас между этим уровнем и порогом, который при переключении элемента воспринимается последующими элементами как начало изменения логического сигнала. Когда на вход КМОП-элемента поступает логическая 1, открывается нижний транзистор, а верхний запирается. В процессе изменения уровня выходного напряжения имеет место короткий интервал времени, когда включены оба выходных транзистора и от источника потребляется в импульсном режиме значительный ток. Эти токо-

либровки осциллографов и логических пробников. ИС этих видов схемотехники относительно редко применяются в бытовой и промышленной аппаратуре; при их использовании рекомендуется обратиться к документации, предоставляемой изготовителем.

Еще одна важная разновидность схемотехники — динамическая логика. Принципы ее работы описаны в гл. 2, где опять-таки не преследуется цель подробного описания внутреннего устройства схем. Динамические ИС применяются в конструктивно оформленных запоминающих устройствах, тестирование которых осуществляется через системную шину. Контроль отдельных ИС может быть весьма затруднителен, так как данные, хранящиеся в устройствах такого типа, требуют систематической «регенерации» по сигналам от внешнего тактового генератора или от самого микропроцессора.

вые импульсы в совокупности с энергией, затрачиваемой на по-переменный заряд и разряд паразитных и внутренних емкостей, являются единственной формой потребления мощности в идеальных КМОП-структурах. В статическом состоянии КМОП-схемы мощность практически не потребляется. Она начинает приближаться к величинам, характерным для ТТЛ и других видов схемотехники, только при очень высокой частоте переключений, измеряемой мегагерцами. В системе обычно работают на максимальных скоростях лишь немногие логические элементы, поэтому среднее потребление мощности остается небольшим.

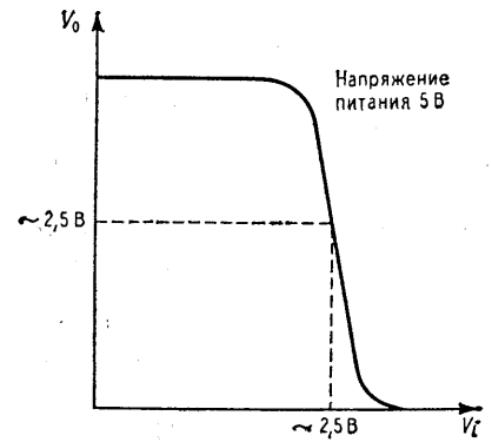


Рис. 1.23. Симметричная переходная характеристика КМОП-схемы — основа высокой помехоустойчивости.

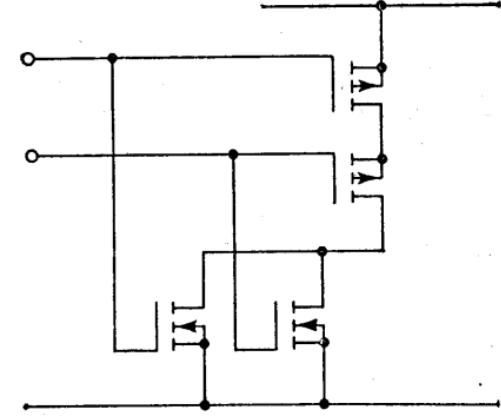


Рис. 1.24. Простой логический элемент на КМОП-транзисторах.

На рис. 1.23 приведена типичная переходная характеристика КМОП-элемента. Выходное напряжение остается близким к 5 В, пока входное не превышает 2 В, и падает почти до нуля, когда входное достигает 3 В. Таким образом, запас помехоустойчивости составляет 2 В, т. е. КМОП-элемент не реагирует на входные помехи, имеющие амплитуду до 2 В.

Схемы, реализующие любые логические функции, компонуются из параллельно и последовательно включенных МОП-транзисторов, как показано на рис. 1.24. КМОП-технология все более широко применяется при изготовлении аналоговых ИС и может быть использована для совместной реализации аналоговых и цифровых функций в таких, например, устройствах, как анало-го-цифровые преобразователи.

Отказы, связанные с отклонениями уровней сигналов, для КМОП-схем менее характерны, чем тривиальные обрывы и короткие замыкания. Все внутрисхемные логические элементы имеют небольшие нагрузки благодаря высоким входным импе-

дансам последующих ступеней схемы. Появление логических уровней, сильно отличающихся от 0 и 5 В, можно ожидать лишь в случаях, когда схема управляет какой-либо внешней нагрузкой.

Заключение

Для построения законченных систем наряду с микропроцессорами необходимы дополнительные схемы и устройства. Логические схемы, изготавляемые на различной конструктивно-технологической основе, значительно отличаются по своим характеристикам, даже если они реализуют одинаковые функции. Эти различия были кратко рассмотрены в данной главе. Существуют тестовые устройства, с помощью которых причины отказов могут быть локализованы с точностью до внешнего контакта устройства; выше были сопоставлены их преимущества и недостатки.

Глава 2

Микропроцессорные системы

Компоненты системы

Микропроцессоры — это цифровые интегральные схемы, предназначенные для выполнения последовательностей операций над двоичными данными. Сама выполняемая операция определяется информацией, получаемой процессором в ходе его последовательных действий, и может, например, представлять собой арифметическое действие над данными, которые предварительно считаются извне и находятся в микропроцессоре. Различные операции, которые может выполнять микропроцессор, называются командами, а последовательность команд принято называть программой. Микропроцессоры являются центральными устройствами микроЭВМ и микроконтроллеров, которые могут выполнять самые разнообразные функции.

Однако микропроцессор не может существовать сам по себе, для построения микропроцессорной системы его необходимо связать с другими устройствами. Для хранения программ и данных необходимы постоянная и оперативная память. Чтобы подключить к системе периферийные устройства, такие, как клавиатуры, принтеры, дисплеи, требуются другие дополнительные схемы. В случае применения микропроцессорной системы для решения задач управления возникает необходимость в схемах сопряжения ее с контролируемой или управляемой системой. В подобных интерфейсных схемах имеют место сочетания как цифровых, так и аналоговых сигналов. Устройства, предназначенные для управления прохождением данных между микропроцессорной системой и средой, в которой она работает, называются устройствами ввода-вывода.

Структурная схема микропроцессорной системы представлена на рис. 2.1. Кроме процессора, памяти и устройств ввода-вывода система обычно содержит некоторые дополнительные устройства, выполняющие функции генерации тактовых импульсов, начальной установки системы, выбора устройств и буферизации сигналов. Существуют также однокристальные микропроцессоры, в которых процессор, постоянная и оперативная память и схемы ввода-вывода размещаются на одном кристалле.

Микропроцессор обменивается информацией с другими компонентами системы, посылая и принимая сигналы по линиям связи, большинство которых объединяется в три основные группы, называемые шинами или магистралями; в их числе шина данных, адресная шина, шина управляющих сигналов.

Кроме того, имеются линии питания, необходимые для подачи напряжения на все устройства системы. Генератор такто-

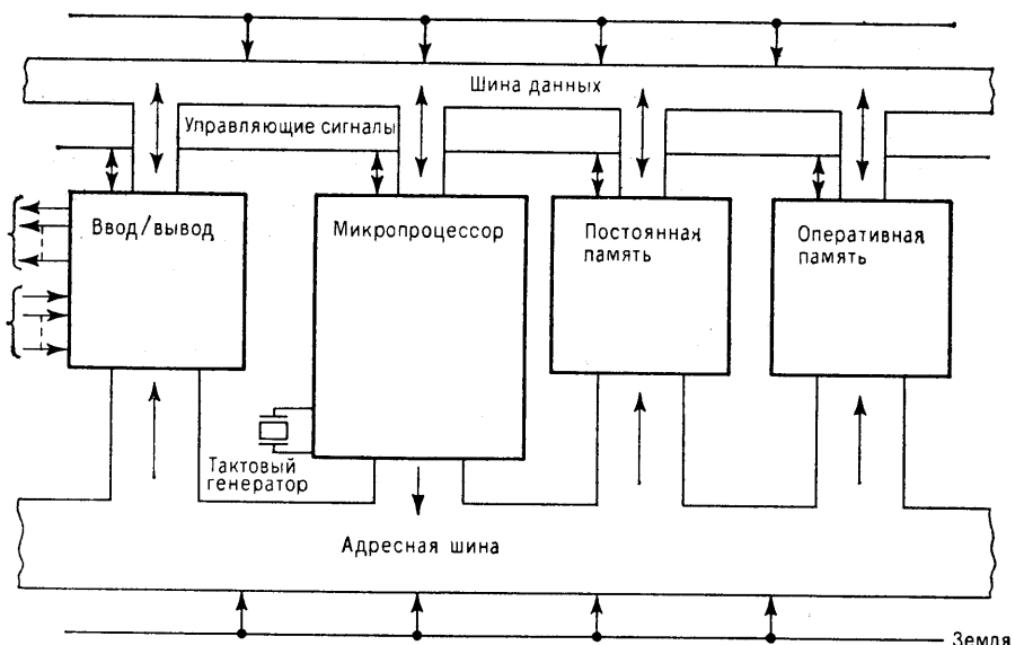


Рис. 2.1. Структурная схема микропроцессорной системы.

вых импульсов вырабатывает синхросигналы, управляющие выполнением действий внутри процессора. Обычно у него есть по крайней мере один буферизированный выход, используемый для тактирования других устройств, работа которых должна быть синхронизирована с работой процессора.

Составные части системы, показанные на структурной схеме, более подробно будут описаны ниже в данной главе.

Системные шины

(а) Шина данных

Шина данных представляет собой набор двунаправленных сигнальных линий, по которым параллельно передается двоичная информация (по каждой линии передается один бит). Первые микропроцессоры имели, как правило, 8-битовые шины данных, а в более поздних появились 16- и 32-разрядные шины.

Число линий, образующих шину (называемое также шириной шины), определяет максимальную величину числа, которое может быть представлено передаваемой по шине двоичной информацией; поэтому желательно, чтобы процессоры могли за один прием передавать как можно большее количество битов данных. (8 бит позволяют представить числа от 0 до 255, а 16 бит — от 0 до 65 535.)

(б) Адресная шина

Адресная шина — это группа сигнальных линий, посредством которых процессор указывает устройство, в которое или из которого должна быть передана информация. Каждое двоичное число, поданное на адресную шину, может обозначать конкретную

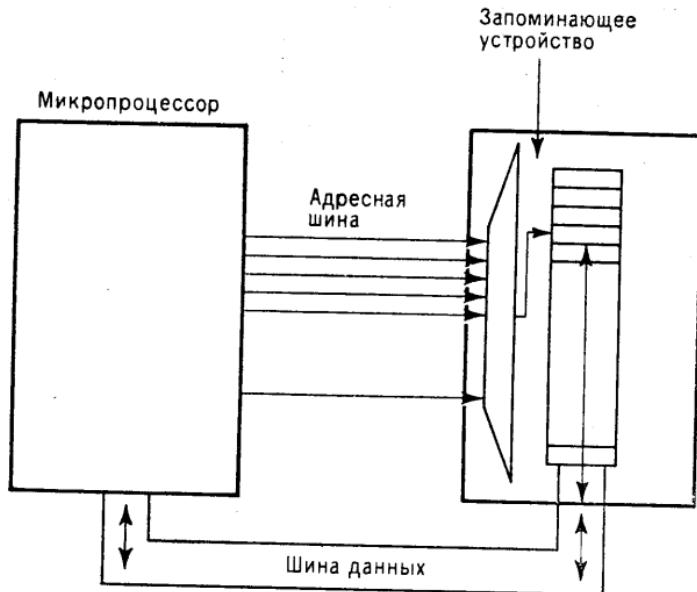


Рис. 2.2. Указание микропроцессором ячейки памяти для передачи информации по шине данных.

область памяти, которая, будучи выбрана, оказывается соединенной с процессором через шину данных и может обмениваться с ним данными.

Выбираемая таким образом область памяти может представлять собой либо ячейку запоминающего устройства, либо регистр, входящий в состав одной из схем ввода-вывода. Принцип адресации иллюстрируется рис. 2.2. Для расширения возможностей микропроцессора нужно, чтобы он мог адресовать большое количество областей памяти, а значит, и число линий в адреснойшине должно быть значительно больше, чем в шине дан-

ных. Восьмиразрядные процессоры обычно имеют 16 адресных линий и могут адресовать до 65 536 ячеек. Шестнадцатиразрядные процессоры, как правило, снабжены 20-разрядными адресными шинами, обеспечивающими возможность обращения к числу ячеек, превышающему миллион.

Адресные линии обычно являются односторонними, процессор управляет ими в ходе выполнения команд. Подаваемая на них информация определяет адресат очередной пересылки данных.

Существуют ситуации, в которых другие устройства, взяв на себя управление процессорными шинами, могут сделать пересылки данных более эффективными, чем под управлением процессора. В качестве таких устройств могут использоваться другие процессоры или устройства ввода-вывода, такие, например, как контроллер дисковой памяти.

Необходимо, чтобы процессор имел возможность предоставить в распоряжение других устройств свои шины данных и адреса, а также сигналы, служащие для управления пересылками данных. Достигается это с помощью особых управляющих сигналов, которые приводят определенные схемы, связанные с нужными линиями, в состояние высокого выходного импеданса.

Этот метод пересылки данных обычно называют прямым доступом к памяти (ПДП). Некоторые процессоры располагают сигналами управления шинами, обеспечивающими высвобождение шин по окончании выполнения текущей команды. Для других требуется дополнительные логические схемы и шинные формирователи, чтобы шины могли высвобождаться в определенные моменты цикла выполнения команды.

ПДП, служащий для обновления содержимого памяти и реализуемый на основе использования циклов обращения к ней, может явиться источником недоразумений, если осуществляется на фоне тестирования памяти. В процессе контроля памяти в нее обычно записываются какие-то наборы данных, а затем проверяется наличие в памяти именно этих наборов. Если одновременно с тестированием памяти осуществляется прямой доступ к ней, данные в памяти искажаются и выдается сообщение о некорректной работе памяти. Значит, необходимо, чтобы перед прогоном тестов памяти возможность ПДП была заблокирована.

Характерной особенностью современных микропроцессоров является использование адресной шины для передачи не только адресов, но и других видов информации за счет временного мультиплексирования шины. При этом адресные линии могут быть применены для передачи данных и других сигналов процессора, т. е. достигается экономное использование внешних выводов ИС микропроцессора. Мультиплексирование адресных шин

используется, например, в микропроцессорах 8085, NSC800, 8088, 8086. Работа таких шин будет описана ниже при рассмотрении временных соотношений сигналов.

В некоторых процессорах адресация ввода-вывода осуществляется отдельно от адресации памяти за счет наличия специальных команд, называемых портовыми операциями и не полностью использующих адресные возможности шины. Для перехода от адресации ввода-вывода к адресации памяти и наоборот нужны специальные управляющие сигналы.

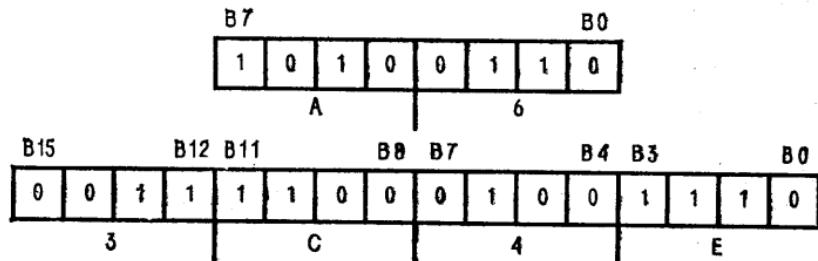


Рис. 2.3. Шестнадцатеричное представление 8-битовых и 16-битовых двоичных чисел.

Для работы с двоичными числами, содержащими более четырех битов, целесообразно иметь способ представления таких чисел, не требующий их записи в виде цепочек единиц и нулей. Одним из таких способов является шестнадцатеричная запись, в которой предусмотрен свой символ для обозначения любого из возможных 4-битовых двоичных чисел. Эти символы, а также десятичные значения 4-разрядных двоичных чисел приведены в табл. 2.1.

Таблица 2.1. Шестнадцатеричные символы и десятичные значения 4-разрядных двоичных чисел

Двоичное число	Шестнадцатеричный символ	Десятичное значение	Двоичное число	Шестнадцатеричный символ	Десятичное значение
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

В случае необходимости представления двоичных чисел, содержащих более четырех цифр, они разбиваются на группы по четыре двоичных разряда, начиная с младшего, и каждая груп-

па заменяется ее шестнадцатеричным представлением. Использование этого приема для изображения как данных, так и адресной информации иллюстрируется рис. 2.3.

Известен также восьмеричный способ записи двоичных чисел, широко применяемый в 16-разрядных мини-ЭВМ и некоторых микропроцессорах, разработанных на их основе, в котором для обозначения 3-битовых групп, на которые разбивается двоичное число, используются десятичные символы (0, 1, ..., 7). Таким образом наибольшее 16-разрядное двоичное число, представляемое в шестнадцатеричной записи как FFFF, в восьмеричной системе имеет вид 177777. В дальнейшем мы будем пользоваться шестнадцатеричной формой представления информации.

(в) Шина управляющих сигналов

Это набор линий для передачи различных сигналов, вырабатываемых процессором и другими устройствами системы. Состав этих сигналов определяется типом процессора, и они могут иметь самые разнообразные функции. Те из них, которые являются выходными сигналами процессора, используются для управления пересылками данных, а также в качестве ответных на входные управляющие сигналы.

Входные управляющие сигналы обычно служат для того, чтобы заставить процессор выполнить определенные действия, или для запросов к нему со стороны других устройств. Это следующие виды сигналов.

Тактовые сигналы. Все процессоры имеют входы для тактовых сигналов, вырабатываемых тактовым (обычно кварцевым) генератором, выход которого непосредственно подсоединяется к процессору. Внутренние схемы процессора обычно в свою очередь вырабатывают одну или две буферизированные серии тактовых сигналов, которые могут быть использованы другими устройствами системы. На рис. 2.4 приведены три типовые схемы включения тактовых генераторов.

Сигнал начальной установки. Эта линия служит для приведения микропроцессора в известное состояние, будучи в котором он может приступить к выполнению программы. В результате подачи сигнала начальной установки в счетчик команд загружается значение, соответствующее адресу первой команды.

Обычно в состав системы входит внешняя по отношению к микропроцессору схема, которая поддерживает линию начальной установки в активном состоянии на протяжении включения питания и снимает сигнал начальной установки после того, как напряжения на всех линиях питания примут установленные значения. Кроме того, в системе, как правило, предусматри-

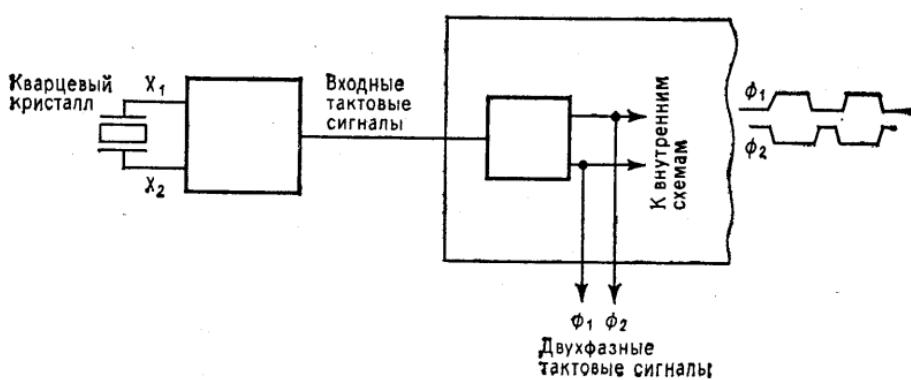
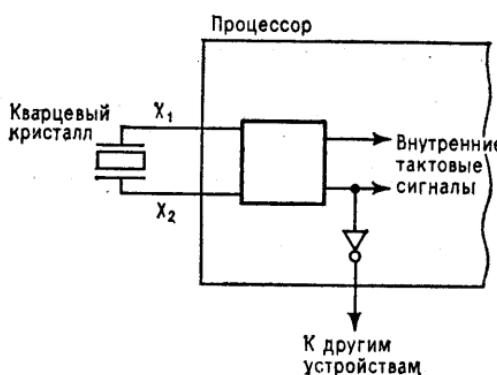
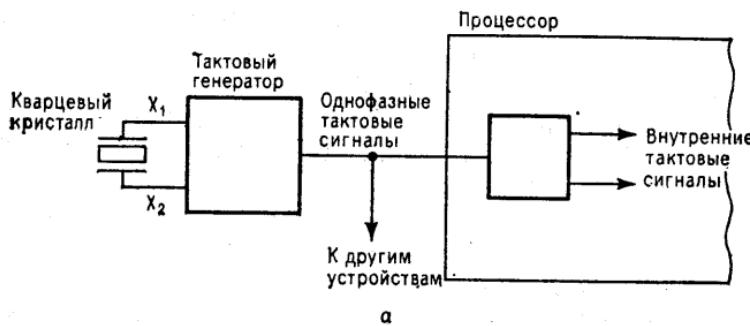


Рис. 2.4. Генераторы тактовых сигналов. а — однофазное внешнее тактирование; б — встроенный генератор тактовых сигналов; в — формирование двухфазного тактового сигнала из однофазного входного.

вается кнопка начальной установки, позволяющая привести систему в исходное состояние без выключения питания. Иногда имеется буферизированный сигнал начальной установки, с помощью которого может осуществляться инициализация других устройств системы, для которых это требуется.

Сигналы управления пересылками данных. Это один или несколько сигналов, вырабатываемых компонентами системы для задания направлений передачи данных. Ввод данных в микропроцессор обычно называют операцией чтения, а вывод — операцией записи. Иногда для указания выполняемой операции пересылки служит одна линия чтения/записи (R/W). В этом случае значение сигнала 1 указывает на то, что должно быть произведено чтение, а значение 0 информирует о том, что процессор собирается выполнить запись данных. В других процессорах для тех же целей применяются два сигнала: RD и WR (Чтение и Запись), оба из которых при отсутствии пересылок пребывают в неактивном состоянии. По мере необходимости активное значение приобретает лишь один из них.

Если в системе используется не один, а несколько способов адресации, необходимы дополнительные сигналы для указания и прекращения пересылок информации с памятью и с устройствами ввода-вывода.

Когда предусмотрены два сигнала, например MEMREQ и IOREQ (Запрос Памяти и Запрос Ввода-Вывода), в любой момент времени в активном состоянии может быть лишь один из них. При использовании одной линии, например IO/M (Ввод-Вывод/Память), вид пересылки данных определяется двумя значениями этого сигнала. Более подробно различные варианты управления пересылками рассматриваются в разделе, посвященном временным соотношениям.

Сигналы состояния. Эти сигналы несут информацию о стадии рабочего цикла микропроцессора, имеющей место в данный момент, например о том, что происходит выборка команды из памяти.

Сигналы прерываний. Они заставляют процессор прервать нормальную последовательность выполнения программы и обратиться к адресу, поступающему из внешних по отношению к микропроцессору устройств или имеющемуся в нем. Механизм прерываний позволяет системе реагировать на события, происходящие асинхронно по отношению к ее внутренним тактовым сигналам и текущему состоянию выполняемой программы.

Сигналы управления шинами. У некоторых микропроцессоров имеются сигналы, с помощью которых может быть снижена скорость работы процессора, что может потребоваться, например, при работе с медленной памятью. По этим сигналам выполнение операций чтения и записи может быть растянуто путем введения дополнительных циклов тактирования или переводом процессора в состояние ожидания, с тем чтобы относительно

медленно работающие устройства имели достаточно времени для выполнения своих действий.

Если процессор имеет возможность высвободить шины и переложить управление ими на другое устройство, для организации этой передачи управления необходимы по крайней мере два сигнала. Первый из них — это запрос на предоставление шин, а второй — подтверждение запроса, получив которое, другое устройство приступает к пересылке данных.

Основные временные соотношения

Операции, кратко описанные в предыдущем разделе, происходят в определенной последовательности, между событиями выдерживаются соответствующие интервалы с таким расчетом, чтобы до начала пересылок между устройствами данные приобретали устойчивые статические значения.

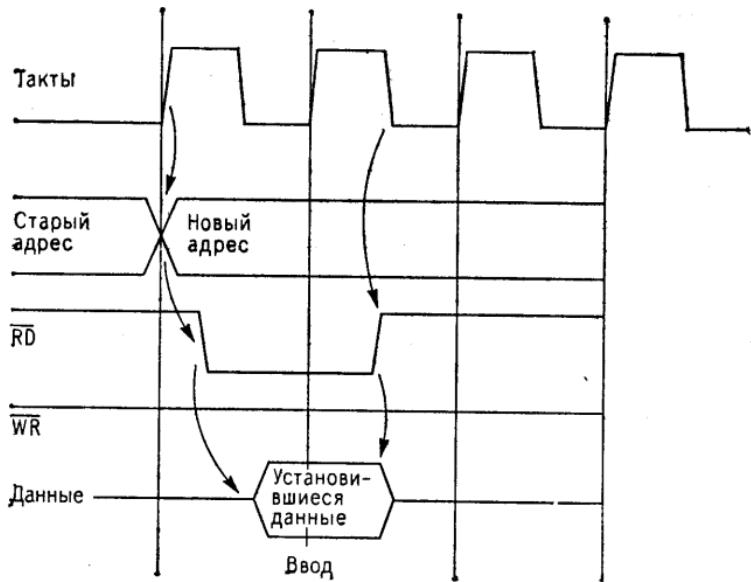


Рис. 2.5. Цикл чтения данных.

Рассмотрим микропроцессор, располагающий двумя сигналами для управления пересылками: \overline{RD} и \overline{WR} . Если он должен прочесть данные из памяти, то первое действие заключается в подаче адреса нужных данных на адресную шину. После этого может быть подан сигнал на линию \overline{RD} . Спустя определенное непродолжительное время, в течение которого устанавливаются адрес и данные на шине приобретают достоверные значения, их можно зафиксировать в регистре процессора, который назначен адресатом. Аналогичная последовательность событий имеет

место и в процессе записи данных с той разницей, что информацию на шину данных помещает процессор и эта информация

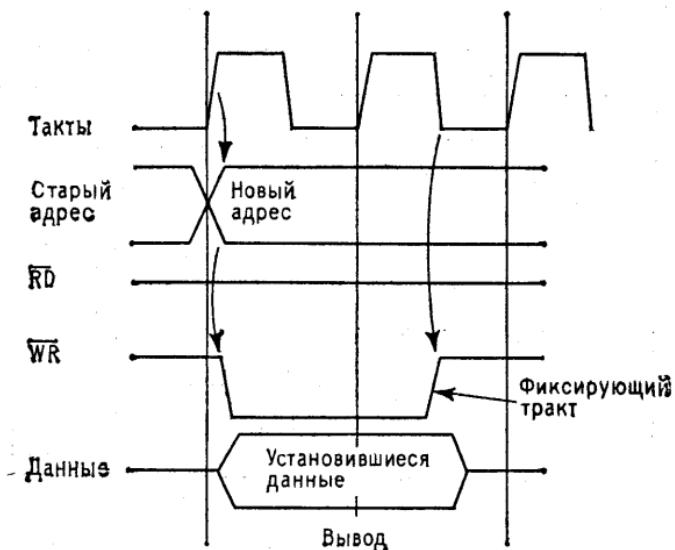


Рис. 2.6. Цикл записи данных.

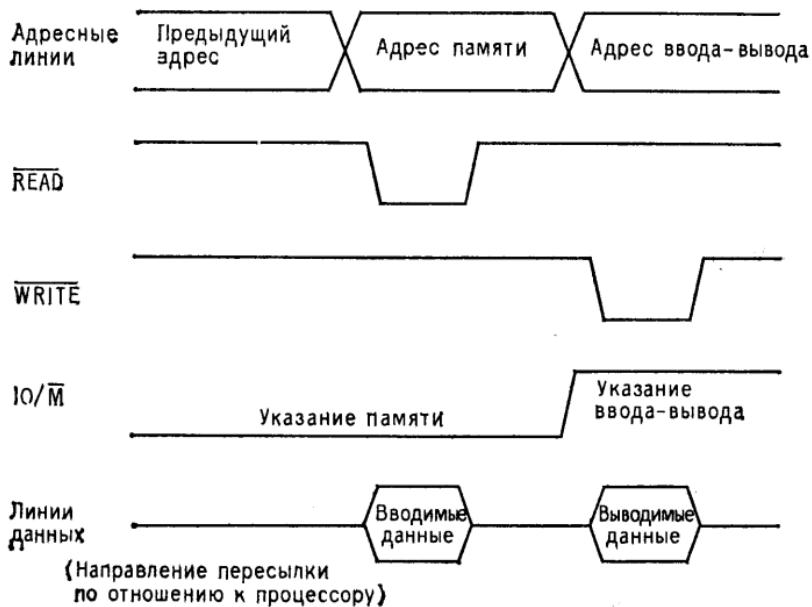


Рис. 2.7. Типовая последовательность обмена данными с памятью и системой ввода-вывода.

сохраняется на шине на протяжении действия управляющего сигнала \overline{WR} . На рис. 2.5 и 2.6 представлены типичные временные диаграммы операций чтения и записи.

В системах с отдельной адресацией ввода-вывода при выполнении чтения и записи с участием памяти подается, кроме того, сигнал, указывающий, что обмен информацией происходит

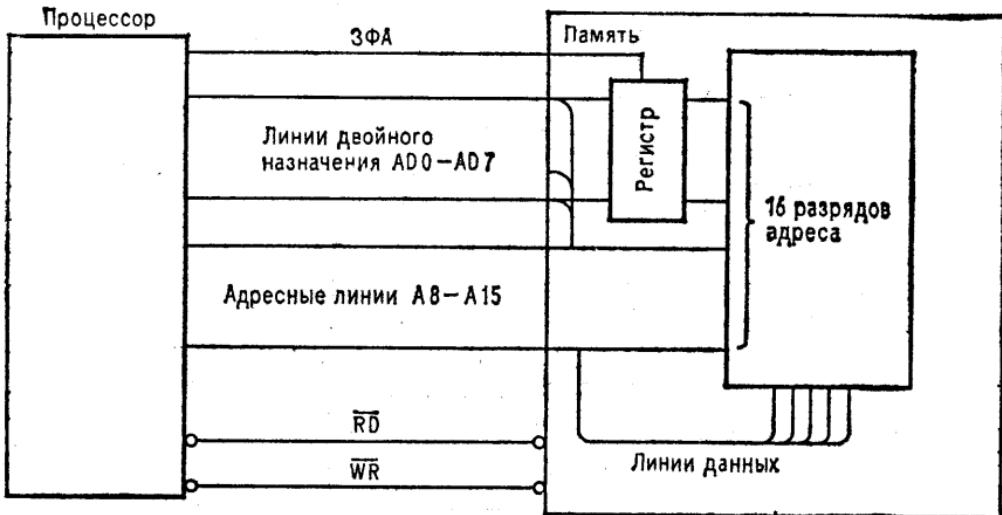
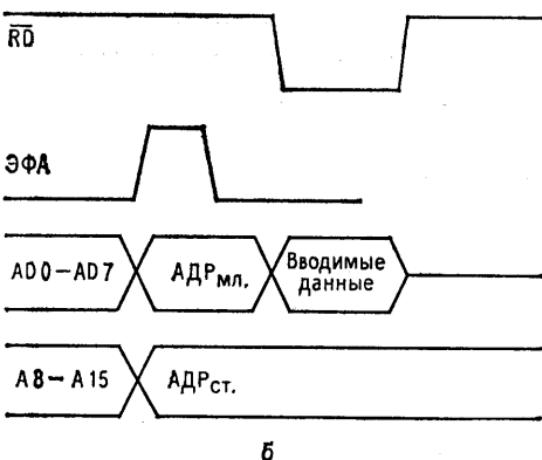
*а**б*

Рис. 2.8. Мультиплексирование шины для передачи адреса и данных: *а* — блок-схема; *б* — временные диаграммы.

именно с памятью. Рис. 2.7 иллюстрирует ситуацию, когда для указания — с памятью или с устройствами ввода-вывода — происходит обмен информацией, служит одна сигнальная линия и после чтения данных из памяти выполняется запись в систему ввода-вывода. Тактовые сигналы на диаграмме не показаны, так как их число и временные параметры определяются конкретным типом процессора.

Еще один метод пересылки, который был кратко упомянут выше, характеризуется несколько иными временными соотношениями: он основан на мультиплексном использовании одной шины для передачи и адресов, и данных. В системах, где этот метод применен, необходимы регистры-фиксаторы, в которых сохраняется адресная информация, пока по адресным линиям передаются данные. Эти фиксаторы могут быть либо встроены в память или другие устройства системы, либо реализованы в качестве дополнительных устройств, связанных с микропроцессором. Такие микропроцессоры, как 8085 и NSC800, в которых применено мультиплексирование адресных шин, обычно комплектуются наборами соответствующих устройств, структурно совместимых с процессорами, которые позволяют обойтись без дополнительных демультиплексирующих схем.

На рис. 2.8 показаны структура системы с мультиплексируемой шиной и временные диаграммы операции чтения данных. Управляющий сигнал «Запись в фиксатор адреса (ЗФА)», служащий для запоминания адресной информации, подается лишь в начале пересылочной последовательности. Восемь старших разрядов адреса процессор передает по соответствующим линиям адресной шины в течение всей пересылки, а восемь младших имеют место на младших линиях шины только на протяжении действия сигнала ЗФА. Задним фронтом этого сигнала младшие разряды адреса записываются в регистр-фиксатор адреса, входящий в состав памяти, после чего оказывается готовым полный 16-разрядный адрес. Теперь при операции чтения линии AD0—AD7 становятся линиями ввода данных, и данные пересылаются по ним из памяти в процессор.

Запоминающие устройства и распределение памяти

Запоминающие устройства (ЗУ), применяемые в микропроцессорных системах, подразделяются на два типа: допускающие только чтение и допускающие как чтение, так и запись. Полупроводниковые ЗУ обоих типов являются устройствами с так называемой произвольной выборкой, т. е. обращение к любой их ячейке осуществляется с одинаковой легкостью, однако ЗУ, допускающие только чтение, принято называть постоянными (ПЗУ), а допускающие и чтение, и запись, — запоминающими устройствами с произвольной выборкой (ЗУПВ). Запоминающие устройства на других носителях, например на магнитных лентах, пузырьковых магнитных доменах и гибких дисках, которые могут применяться в микропроцессорных системах для построения вспомогательной памяти, являются, напротив, устройствами с последовательной выборкой, так как время, необходимое для получения или запоминания данных, зависит при работе с такими ЗУ от положения адресуемой области памяти.

Полупроводниковое запоминающее устройство представляет собой матрицу запоминающих элементов (или ячеек), каждый из которых может хранить один бит. Ячейки сгруппированы в слова, каждое из которых содержит один или множество битов и может быть выбрано путем подачи двоичного адреса на имеющийся в памяти дешифратор, который работает таким образом, что при подаче любого адреса на его вход оказывается выбранным только одно слово (дешифратор «1 из 2^n », где n — число

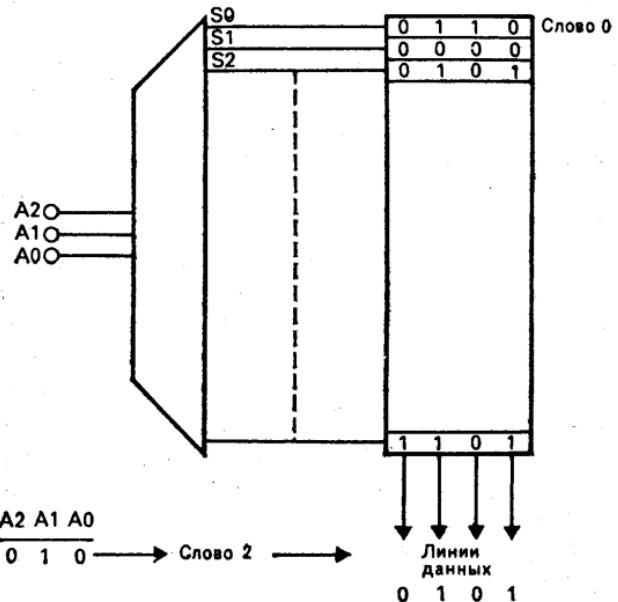


Рис. 2.9. Выбор ячейки памяти дешифратором и подключение ее к линиям пересылки данных.

адресных линий). Этот принцип выборки иллюстрируется применительно к ПЗУ на рис. 2.9. Если память имеет большой объем, то она обычно снабжена двумя дешифраторами — строка и столбца, с помощью которых выбирается слово, находящееся на пересечении выбранных строки и столбца (рис. 2.10). Кроме того, как правило, имеется один или несколько управляющих сигналов, которым должно быть придано активное значение, чтобы оказалось выбранным конкретное запоминающее устройство. Линии данных тех ЗУ, которые в данный момент не выбраны, пребывают в состоянии высокого управляющего импеданса.

Для ЗУПВ необходимы управляющие сигналы, указывающие направление пересылки данных. Это может быть достигнуто с помощью одной управляющей линии R/\bar{W} , когда на нее подан сигнал высокого уровня, выбранное слово памяти подается на линии данных, а когда этот сигнал имеет низкий уровень,

данные, поступившие на шину данных, заносятся в выбранную область памяти, замещая ранее находившиеся в ней данные. При такой кодировке сигнала R/W его положительным фронтом данные фиксируются в нужной области памяти. На рис. 2.11 показана типичная последовательность прохождения информации по линиям адресов, данных и управления.

Содержимое в ПЗУ может быть занесено при изготовлении путем масочного программирования, если объем выпуска доста-

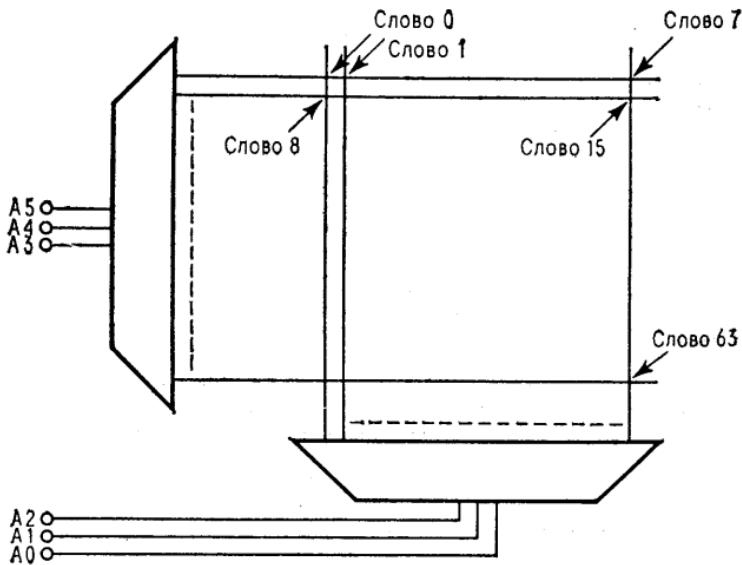


Рис. 2.10. Выбор слов в памяти с помощью дешифраторов по двум адресным координатам x и y .

точно велик, чтобы были оправданы затраты на постановку этого технологического процесса. На этапе разработки системы, а также при малой повторяемости содержимого ПЗУ в различных системах обычно применяется память со стиранием. Программируемые ПЗУ (ППЗУ), содержимое которых может быть стерто и записано вновь, бывают двух основных типов: стираемые путем ультрафиолетового облучения (стираемые программируемые — СППЗУ) и электрически перепрограммируемые (электрически перепрограммируемые — ЭППЗУ). Существуют также ПЗУ, программируемые только один раз пережиганием плавких перемычек.

ЗУПВ бывают двух типов: статические и динамические. Статические ЗУ сохраняют информацию, пока на них подано напряжение питания; энергонезависимыми, т. е. сохраняющими информацию при выключении источника питания, являются также ЗУПВ на малоэнергопотребляющих КМОП-структуратах, снабженные вспомогательным аккумуляторным питанием. Содержи-

мое динамической памяти должно подвергаться регенерации путем систематической активизации адресных линий, даже когда пересылок данных не происходит.

В состав микропроцессорной системы, как правило, входят и ПЗУ, и ЗУПВ. ПЗУ служит для хранения неизменяемых про-

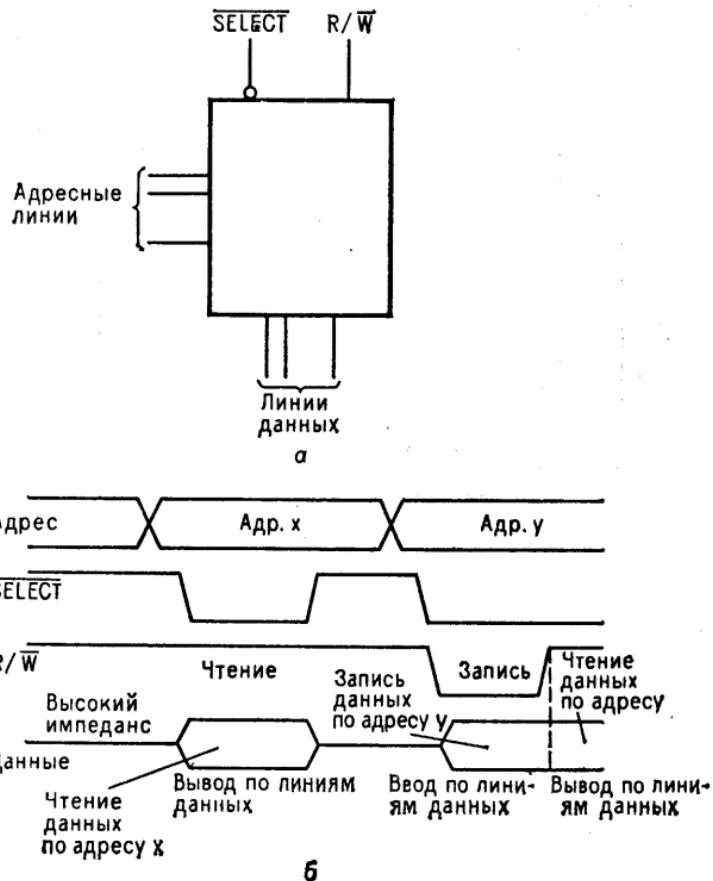


Рис. 2.11. Обращение к памяти: а — блок-схема; б — временные диаграммы последовательно выполняемых операций чтения и записи.

грамм и данных, а ЗУПВ — для размещения переменных данных, полученных при выполнении программы. Когда для хранения программ применяются запоминающие устройства на дисках и лентах, программы, подлежащие исполнению, обычно пересылаются из них в ЗУПВ.

Декодирование адресов

Когда в состав системы входит несколько запоминающих устройств, необходимо каким-то схемным путем обеспечить выборку в любой момент времени не более чем одного из них.

Соответствующая схема обычно представляет собой дешифратор, декодирующий старшие разряды адреса памяти, посредством которых задается размещение отдельных ЗУ в адресном пространстве системы. Это разделение памяти может быть описано с помощью диаграммы, обычно называемой картой распределения памяти.

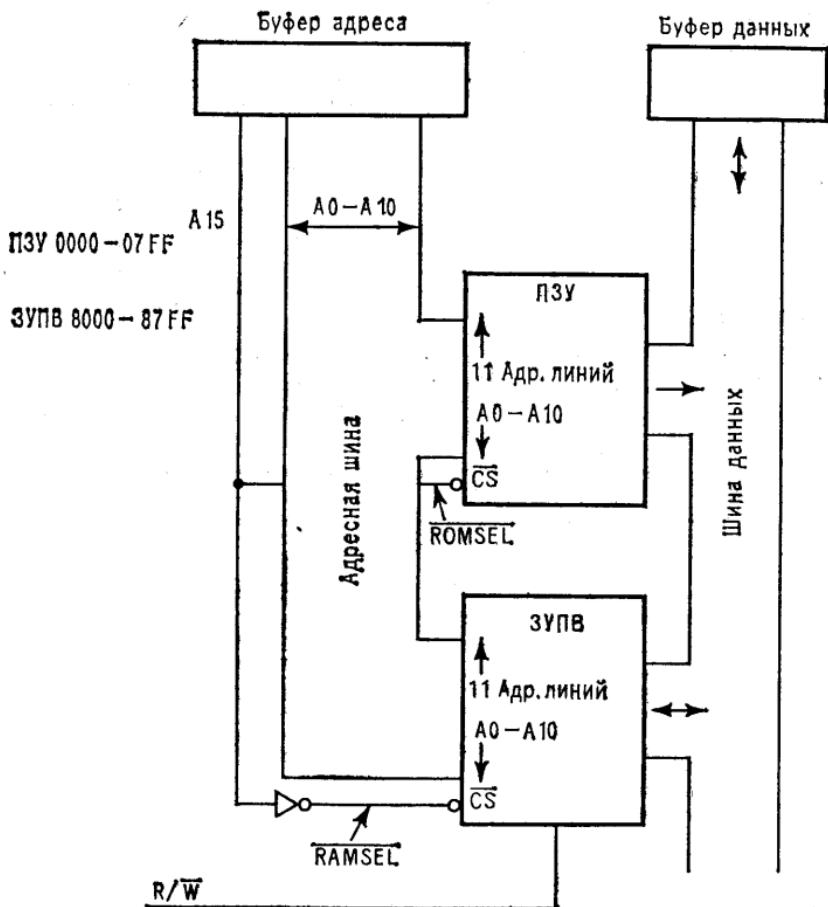


Рис. 2.12. Использование разряда адреса A15 при выборе микропроцессором одного из двух запоминающих устройств.

Рассмотрим небольшую микропроцессорную систему, для которой необходимы ПЗУ объемом 2К с нулевым начальным адресом и дополнительный блок ЗУПВ объемом 2К для хранения переменных данных.

Один из способов подключения этих двух ЗУ к адреснойшине показан на рис. 2.12. Благодаря наличию инвертора в любой момент времени может быть выбран лишь один из этих блоков. При $A15 = 0$ выбирается ПЗУ, а при $A15 = 1$ — ЗУПВ. Так как каждое ЗУ имеет объем 2К, адреса, лежащие в диапа-

зоне 0000—07FF, ставятся в соответствие ПЗУ, а адреса 8000—87FF оказываются принадлежащими ЗУПВ. Однако не только эти адреса составляют адресное пространство, в котором размещаются ПЗУ и ЗУПВ-системы.

Рассмотрим переход от адреса 07FF к адресу 0800. Так как разряд A15 сохраняет при этом нулевое значение, выбранным по-прежнему остается ПЗУ.

Таким образом, по выданному микропроцессором адресу 0800 имеет место доступ к тем же словам ПЗУ, что и по адресу 0000. Следовательно, диапазон адресов 0800—08FF обес печивает получение тех же слов, что и диапазон 0000—07FF. Это явление называется «многослойной укладкой»; суть его в том, что описанное ПЗУ с его диапазоном адресов 0000—07FF шестнадцать раз «укладывается» в той части адресного пространства, для которого разряд адреса A15 имеет значение 0. Аналогичным образом оказывается, что одни и те же слова ЗУПВ выбираются в диапазонах адресов 8000—87FF, 8800—8FFF и т. д. до F800—FFFF. Кarta распределения памяти для рассматриваемой конфигурации системы представлена на рис. 2.13. Подобная организация памяти используется в одноплатной системе Z80, рассматриваемой в приложении 2.

На рис. 2.14 показана схема декодирования адресов памяти, в которой для выбора одного из четырех блоков памяти используется дешифратор 3×8 (т. е. имеющий три входа и восемь выходов). Сигнал на каждом из выходов дешифратора, пребыва в активном состоянии, выбирает, как показано на схеме, диапазон адресов 2К в поле памяти, адресуемой процессором; поэтому два первых блока ПЗУ не имеют повторяющихся слоев. Блоки ЗУПВ имеют объем всего по 1К, поэтому каждый из них дважды фигурирует в диапазоне адресов 2К, выбираемом со ответствующим выходом дешифратора. Кара распределения

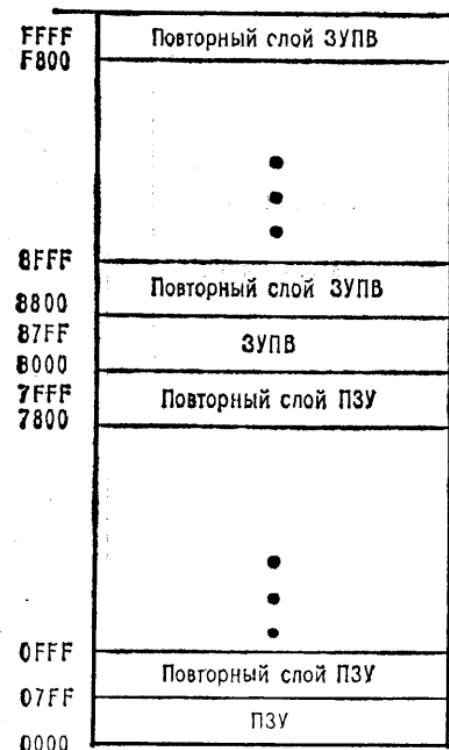


Рис. 2.13. Кара распределения памяти в системе, показанной на рис. 2.12.

памяти приведена на рис. 2.15. Такой способ использования адресного пространства применяется в некоторых одноплатных микроЭВМ, таких, например, как AIM65 и SDK85.

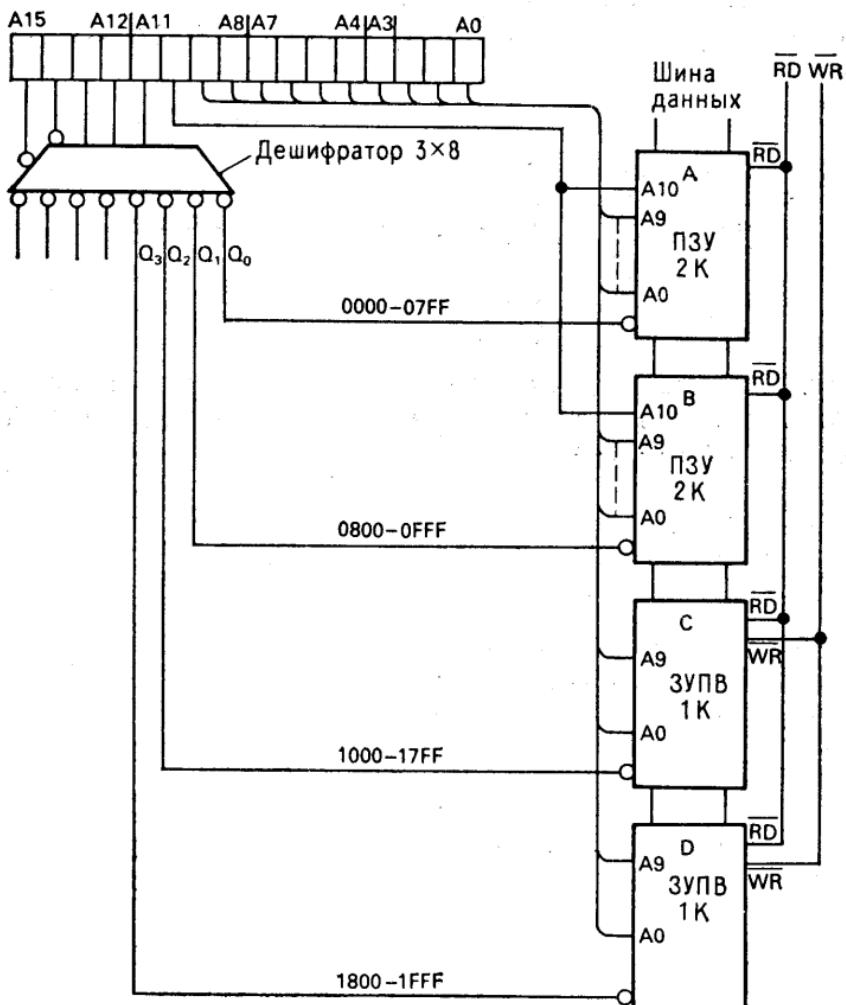


Рис. 2.14. Микропроцессор с четырьмя блоками памяти.

Применение ПЗУ для декодирования адресов

В некоторых системах для декодирования адресов вместо обычных дешифраторов используются ПЗУ с плавкими перемычками. Возможность сопоставления выходных сигналов такой декодирующей схемы различным диапазонам адресов позволяет повысить гибкость адресации памяти системы, что иллюстрируется примером на рис. 2.16, где показано ПЗУ объемом 32 восьмиразрядных слова, осуществляющее выборку блоков памяти разного объема.

Как и в любой схеме декодирования адресов, необходимо обеспечить, чтобы в любой момент времени осуществлялась выборка не более чем одного блока памяти. Это означает, что единственным допустимым набором данных, который может быть занесен в 8-разрядное декодирующее ПЗУ, является следующий:

1111	1111	FF	Ни один из выходов не активен
1111	1110	FE	
1111	1101	FD	
1111	1011	FB	
1111	0111	F7	Активен один из выходов
1110	1111	EF	
1101	1111	DF	
1011	1111	BF	
0111	1111	7F	

В данном случае линия выборки считается находящейся в активном состоянии, когда управляющий сигнал на ней (Y_C) имеет низкий уровень.

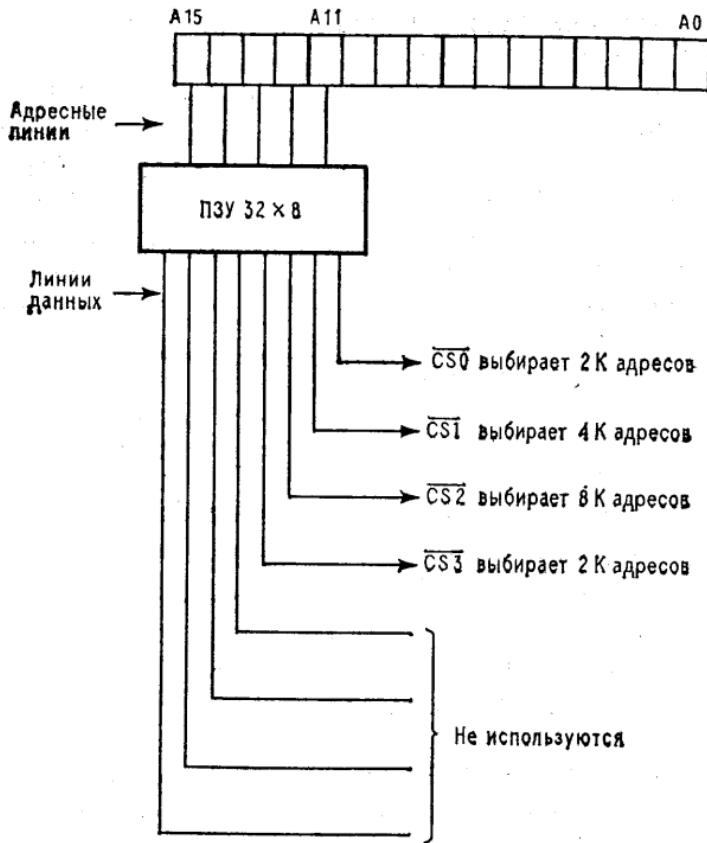
Особенности функционирования динамической памяти

Для предотвращения потери данных содержимое динамической памяти должно постоянно регенерироваться. Значения данных в каждой ячейке такой памяти определяются зарядом внутренней емкости ячейки, которая, как правило, состоит из одного транзистора, что создает предпосылки для создания запоминающих устройств с большой плотностью хранения, высоким быстродействием и малым рассеянием мощности. Регенерация необходима для пополнения зарядов, убывающих из-за наличия утечек.

Для адресации динамических ЗУ обычно используется мультиплексирование адресных линий; адрес при этом должен представляться в виде двух составных частей. С помощью двух стробирующих сигналов — сигнала записи адреса строки (RAS) и сигнала записи адреса столбца (CAS) — адресная



Рис. 2.15. Карта распределения памяти с повторными слоями адресации блоков 1К, входящих в состав системы, которая показана на рис. 2.14.



<u>Диапазон адресов процессора</u>	<u>Адреса ПЗУ 32x8</u>	<u>Данные в ПЗУ 32x8</u>
0000 - 07FF	00	11111110 - FE
0800 - 0FFF	01	
1000 - 17FF	02	11111101 - FD
1800 - 1FFF	03	
2000 - 27FF	04	
2800 - 2FFF	05	11111011 - FB
3000 - 37FF	06	
3800 - 3FFF	07	11110111 - F7
4000 - FFFF	08	
	1F	11111111 - FF

↑
 CS3 | CS1
 ↓
 CS2

Рис. 2.16. Применение ПЗУ для выборки запоминающих устройств в микропроцессорной системе.

информация фиксируется в памяти. Адрес строки обычно представляется собой младшую половину полного адреса, а адрес столбца — старшую половину. Для передачи адресов строки и столбца в нужной последовательности между адресной шиной процессора и адресными линиями динамического ЗУ устанав-

ливается мультиплексор. Передача информации происходит обычно в следующем порядке.

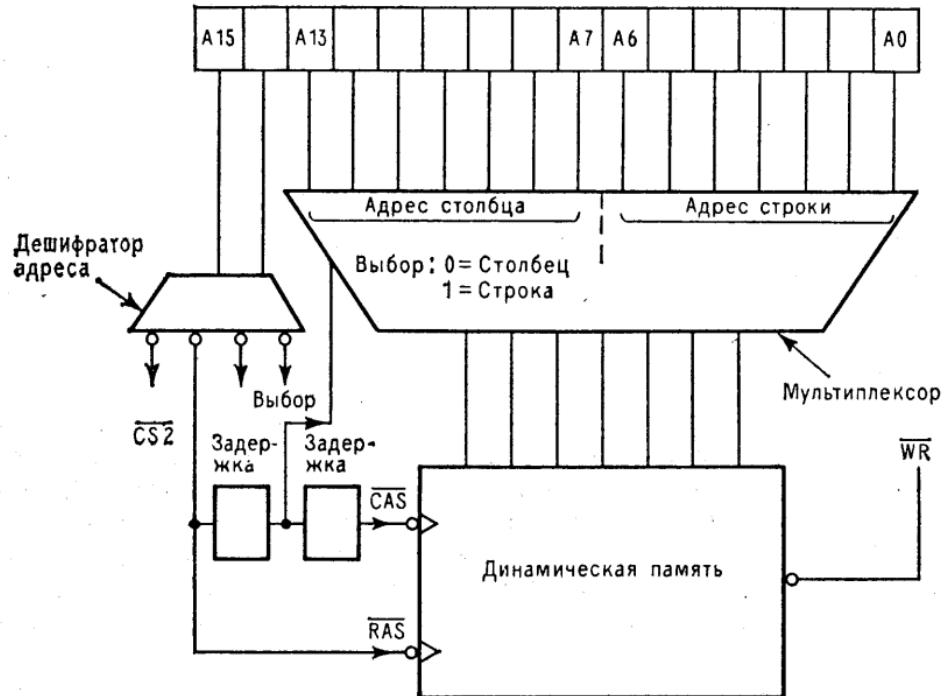
На адресные линии памяти поступает адрес строки. Сигналы RAS и CAS первоначально имеют высокий уровень. Сигнал RAS принимает низкий уровень для фиксации адреса строки. После небольшой задержки на адресные линии подается адрес столбца, и для его записи придается низкий уровень сигналу CAS. Если осуществляется чтение информации из памяти, то после обработки адреса декодирующими схемами на выходе ЗУ появляются данные.

В случае записи информации должны иметь место достоверные входные данные и в течение некоторого времени после окончания сигнала CAS продолжает сохраняться активное значение сигнала разрешения записи. Обратите внимание, что одновременно с активным состоянием сигнала CAS продолжает сохранять низкий уровень сигнал RAS. Довольно часто сигнал CAS формируется из сигнала RAS с помощью схем задержки, которые, кроме того, управляют заменой адреса строки на адрес столбца в интервале между отрицательным перепадом сигнала RAS и таким же изменением сигнала CAS. Построение схемы управления с использованием этих приемов иллюстрируется рис. 2.17. Отметим, что на этой схеме не показано, каким образом на адресную шину подаются адреса регенерации.

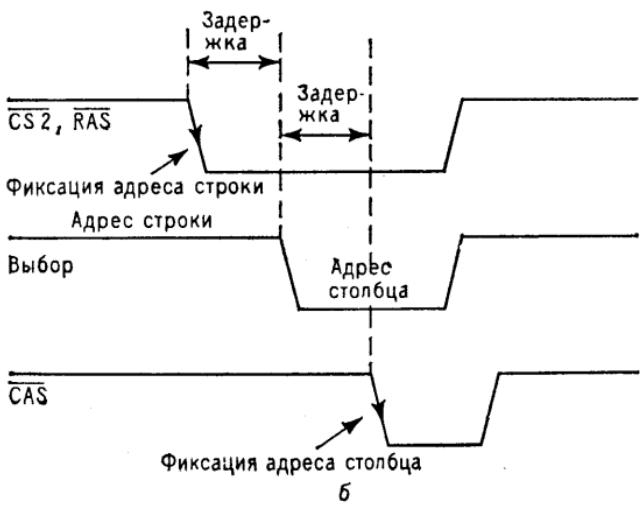
Регенерация выполняется каждый раз, когда происходит чтение или запись данных. Одновременно регенерируются все ячейки, соответствующиециальному адресу строки (кроме той ячейки, содержимое которой подвергается изменению при выполнении записи). Кроме того, выполняются дополнительные действия по регенерации, чтобы обеспечить уверенную систематическую регенерацию всех ячеек с периодом, как правило, не меньшим, чем 2 мс.

Обычно тем или иным способом осуществляется последовательный перебор адресов всех строк, в ходе которого подается только сигнал RAS. Происходит это в те промежутки времени, когда процессор не обращается к динамической памяти. Для реализации данной процедуры применительно к схеме, показанной на рис. 2.17, необходимы дополнительный мультиплексор и 7-разрядный двоичный счетчик. Еще некоторые дополнительные логические схемы требуются для того, чтобы предотвратить выработку сигнала CAS из сигнала RAS и полностью блокировать действия по регенерации, когда к памяти обращается микропроцессор.

Микропроцессор Z80 снабжен средствами, упрощающими сопряжение с динамической памятью. В нем имеется внутренний



а



б

Рис. 2.17. а — организация системы с динамической памятью; б — упрощенная временная диаграмма.

регенерационный регистр с автоматическим инкрементом, содержащее которого подается на адресную шину каждый раз после цикла выборки команды (или эквивалентного ему цикла в командах с блочной процедурой выполнения). Сигнал RFSH, служащий для указания момента выдачи регенерационного адреса, может быть использован для активизации сигнала RAS и блокирования выработки сигнала CAS. Дополнительного мультиплексора не требуется, так как его функции по существу реализованы внутри процессора.

Микропроцессор

Доступ к внутренним схемам микропроцессора с целью наблюдения логических значений и формы логических сигналов, а также для устранения неисправностей невозможен. Однако необходимо знать его внутреннее устройство и функционирование. Прежде чем приступить к изучению видов отказов и способов выявления неисправностей, следует разобраться, как работает микропроцессор при их отсутствии. В этом разделе описывается общая структура микропроцессора, его функционирование, а также использование хранящихся в памяти двоичных чисел в качестве последовательности команд, которую он может исполнять.

Микропроцессоры представляют собой цифровые большие интегральные схемы (БИС), предназначенные для выполнения относительно простых операций, иначе называемых командами, которые считываются и исполняются последовательно с большой скоростью. К числу внутренних схем микропроцессора относятся многоразрядные регистры, параллельные тракты данных, буферы для подключения внешних устройств, многофункциональные схемы, логические схемы синхронизации и управления. Многофункциональные схемы предназначены для реализации простых арифметических и логических действий над двоичными числами, находящимися в регистрах процессора, и пересылок данных как внутри процессора, так и между ним и внешними устройствами. Схемы синхронизации и управления задают порядок действий процессора, для выполнения функций синхронизации им необходимы постоянно поступающие тактовые импульсы.

Каждая операция выполняется в определенной последовательности в течение нескольких циклов синхронизации, количество зависит от сложности операции. Содержание операций определяется командами, т. е. информационными словами, принимаемыми микропроцессором в регистр, называемый регистром команд. Эти слова поступают на дешифратор, связанный

с управляющей логикой, которая устанавливает тип операции и выполняет ее.

В 8-битовом регистре команд может находиться одно из 256 различных чисел (от 00000000 до 11111111 в двоичном представлении, т. е. от 00 до FF в шестнадцатеричном); таким образом, 8-разрядный микропроцессор в общем случае имеет 256 различных команд. Микропроцессор Z80 и его аналоги являются в этом отношении исключениями, так как коды некоторых их операций имеют длину несколько байтов (для полного определения операции в этом случае необходимо чтение из памяти более чем одного 8-битового числа).

Способность выполнять множество различных операций в последовательности, которую задает пользователь с помощью данных, размещаемых во внешней памяти, делает микропроцессор мощным и гибким инструментом. Эта способность не является, однако, принципиально новой для цифровых устройств, так как можно считать, что многие из таких устройств уже при среднем уровне интеграции допускают реализацию ограниченного числа команд.

Рассмотрим, например, двоичный реверсивный счетчик с начальной установкой. Можно считать, что он представляет собой устройство, выполняющее три операции в зависимости от значений двоичных сигналов, управляющих направлением счета и начальной установкой следующим, например, образом:

- 00 — счет на вычитание;
- 01 — пребывание в начальном состоянии;
- 10 — счет на прибавление;
- 11 — пребывание в начальном состоянии.

(При единичном значении сигнала управления направлением счета осуществляется счет на прибавление, а при единичном значении сигнала начальной установки сохраняется начальное состояние счетчика.)

Последовательность значений, принимаемых этими битами в сочетании с серией импульсов, поступающих на вход тактирования, определяет порядок работы схемы по подсчету входных сигналов. Значит, эта простая система может выполнять «команды», однако она не способна выбирать их из памяти. Обобщенная структура микропроцессора, соответствующая приведенному выше описанию, приведена на рис. 2.18.

Более детальная картина организации микропроцессора включает адресную шину и регистры для хранения адресной информации. В 8-разрядных микропроцессорах эти регистры, как правило, имеют 16-разрядный формат, их загрузка производится по внутренней 8-разряднойшине данных в два приема. С адресной шиной обычно связаны три регистра: счет-

чик команд (СК), указатель стека (УС) и регистр адреса (РА).

Счетчик команд используется в процессоре для указания адресов, по которым в памяти находятся команды; после считывания каждой команды автоматически производится его инкремент. Чтобы разработчик системы мог разместить первую

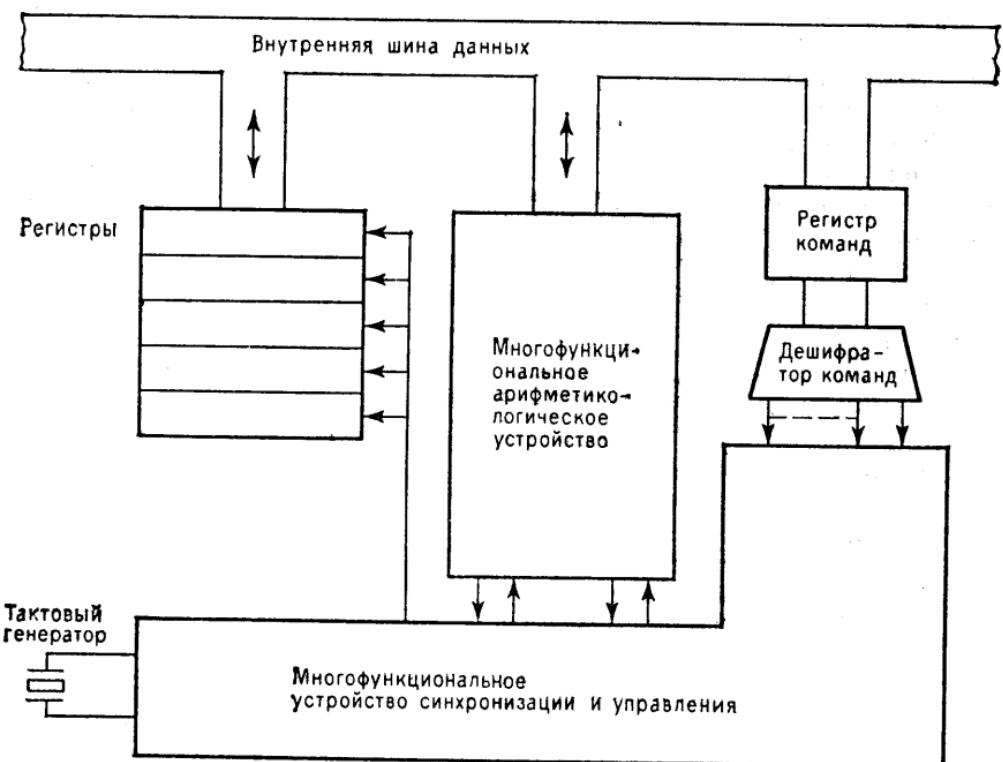


Рис. 2.18. Укрупненная структурная схема микропроцессора.

команду программы по определенному адресу, при подаче сигнала начальной установки в счетчик команд загружается фиксированное значение. После этого по данному адресу может быть считана первая команда программы; обычно этот исходный адрес относится к ПЗУ.

В некоторых процессорах загрузка начального значения в счетчик команд выполняется автоматически. В качестве примера назовем микропроцессор 8080 и его архитектурные аналоги, в которых устанавливается нулевой начальный адрес. В других процессорах загрузка счетчика команд производится из двух фиксированных областей памяти, что дает разработчику системы возможность выбора при размещении программы в памяти. Микропроцессор 6502 получает начальный адрес из ячеек

FFFF и FFFD, а микропроцессор 6800 — из ячеек FFFE и FFFF.

Автоматический инкремент счетчика команд обеспечивает их последовательное чтение из памяти. Дополнительная гибкость в использовании счетчика достигается с помощью команд и входных сигналов, посредством которых в него могут быть занесены новые значения и тем самым изменена последователь-

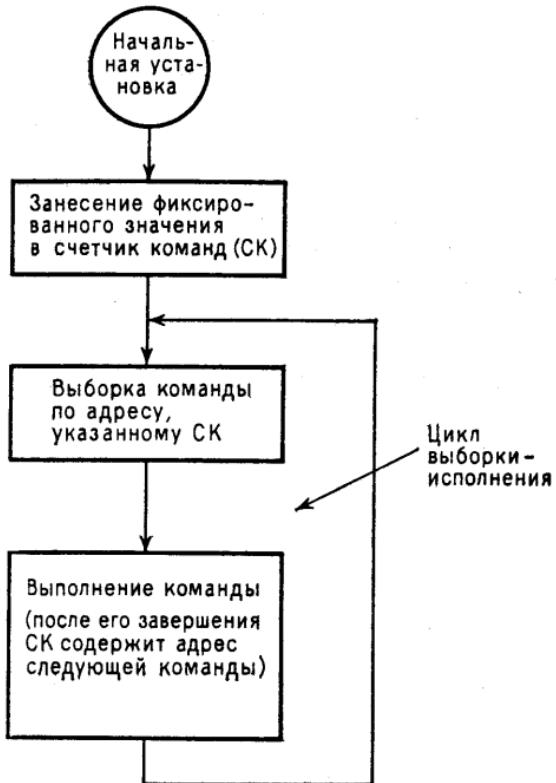


Рис. 2.19. Микропроцессорный цикл выборки-исполнения.

ность выполнения команд. В результате становится возможной организация программных циклов, обращение к произвольным участкам программы и к другим программам, которые служат для обработки так называемых прерываний, происходящих не всегда синхронно с тактовыми сигналами процессора.

Микропроцессор не только считывает команду, но должен и выполнить ее перед считыванием следующей. Эта последовательность действий, которую называют циклом выборки-исполнения, показана на рис. 2.19. Регистр указателя стека служит для указания адреса ЗУПВ, по которому может быть временно размещено содержимое процессорного регистра. Значение указателя стека автоматически подвергается инкременту и декре-

менту в зависимости от того, записываются или считываются данные по указываемому адресу. Зона памяти, используемая командами, которые выполняются с привлечением указателя стека, называется стеком и функционирует по принципу «последним записан — первым считан». При написании программ, содержащих команды записи и чтения стека, обычно называемые соответственно занесением в стек и извлечением из стека, необходимо обращать внимание, чтобы количество команд обоих типов было сбалансировано.

Под действием некоторых команд и сигналов содержимое программного счетчика «заталкивается» в стек, чтобы сохранить адрес очередной команды в последовательности, естественный ход которой нарушается. Команды, при выполнении которых имеют место стековые операции, называются командами обращения к подпрограммам; они позволяют переходить к широко используемым стандартным программам из различных частей основной программы.

Сигналы, инициирующие действия со стеком, называются сигналами прерываний; с их помощью переходы к подпрограммам осуществляются под влиянием определенных событий, происходящих асинхронно по отношению к выполняемой программе. В процессе обращения к подпрограмме или обработки прерывания в счетчик команд заносится новый адрес команды. Теперь выполняется программа, начинаяющаяся с этого адреса и заканчивающаяся так называемой командой возврата из подпрограммы, посредством которой хранящееся в стеке значение программного счетчика «выталкивается» из него, после чего возобновляется выполнение прерванной программной последовательности.

Если происходит занесение в стек или извлечение из него значений 16-разрядных слов, для каждой такой операции необходимы две пересылки данных. Когда необходимо осуществлять передачи данных между процессором и областями памяти, адреса которых не находятся ни в счетчике команд, ни в указателе стека, используется регистр адреса. Тем самым расширяются возможности процессора по запоминанию и извлечению данных из памяти. Иногда бывают предусмотрены средства загрузки содержимого в этот регистр из других регистров.

На рис. 2.20 приведена более подробная схема микропроцессора, на которой показан вычислительный блок наряду с рассмотренными выше адресной шиной и регистрами обработки информации. К их числу относится арифметико-логическое устройство (АЛУ), представленное на схеме вместе с одним из регистров общего назначения, относительно более тесно взаимодействующим с АЛУ, который называется аккумулятором или регистром А. В нем обычно размещаются результаты арифме-

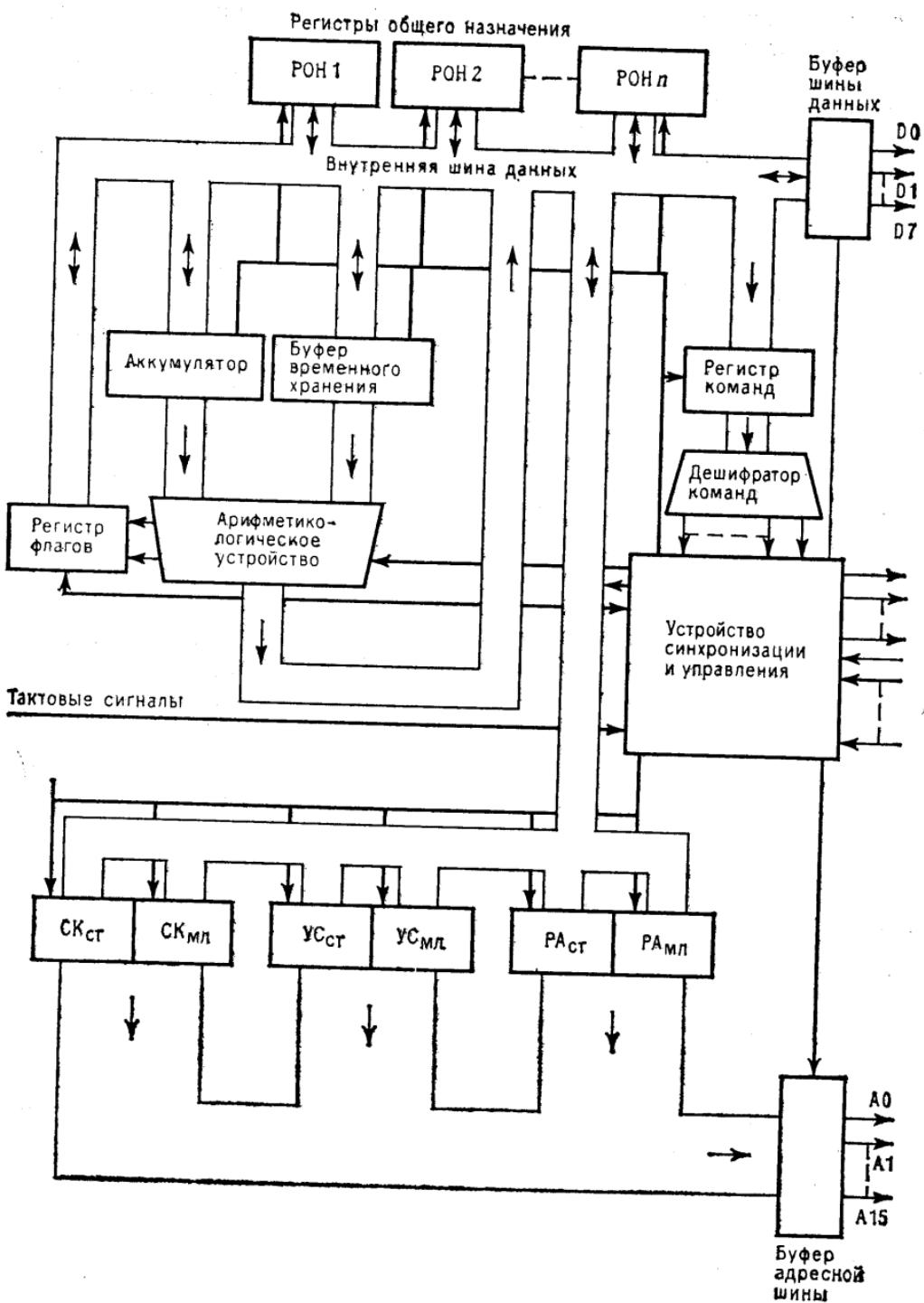


Рис. 2.20. Структура микропроцессора.

тических операций, аналогично тому как это происходит в регистре результата карманного калькулятора.

Еще один регистр, обычно связанный с вычислительным блоком микропроцессора, — это регистр флагов. Он состоит из нескольких независимо управляемых триггеров, которые, будучи установлены в единичное состояние, указывают на различные признаки результата, такие, например, как

- равенство результата нулю;
- знак результата (положительный или отрицательный);
- четность суммы двоичных цифр результата (четная или нечетная);
- наличие переноса из старшего разряда или заема в этот разряд.

Загрузка некоторых регистров процессора, обработка их содержимого и пересылка данных из них в другие регистры могут осуществляться с помощью команд программы. Это так называемые пользовательские регистры. Другие регистры недоступны программисту и играют роль буферов для временного хранения информации.

Работа микропроцессора

Чтобы описать различные команды, характерные для 8-разрядных микропроцессоров, необходимо более подробно рассмотреть описанный в предыдущем разделе цикл выборки-исполнения. Эти команды обычно бывают трех форматов: однобайтовые, двухбайтовые (хотя Z80 и его более поздние версии располагают слегка расширенным набором форматов за счет возможности наращивания дополнительными байтами поля кода операции).

Самыми простыми являются команды, при выполнении которых обрабатывается информация, уже находящаяся в регистрах процессора. Происходит, например, передача данных из одного регистра в другой или содержимое какого-то регистра прибавляется к содержимому регистра А. Эти команды называются однобайтовыми, так как для полного их определения достаточно занести 8-разрядный код операции в регистр команд. Для реализации других команд в регистры процессора необходимо, кроме кода операции предварительно занести из памяти один или несколько байтов данных; такие команды называются двухбайтовыми и трехбайтовыми.

На рис. 2.21 представлен более детально кратко описанный выше цикл выборки-исполнения, обеспечивающий реализацию многобайтных команд. Первым шагом процесса выборки-исполнения является считывание в регистр команд данных, получаемых из памяти по адресу, содержащемуся в счетчике команд.

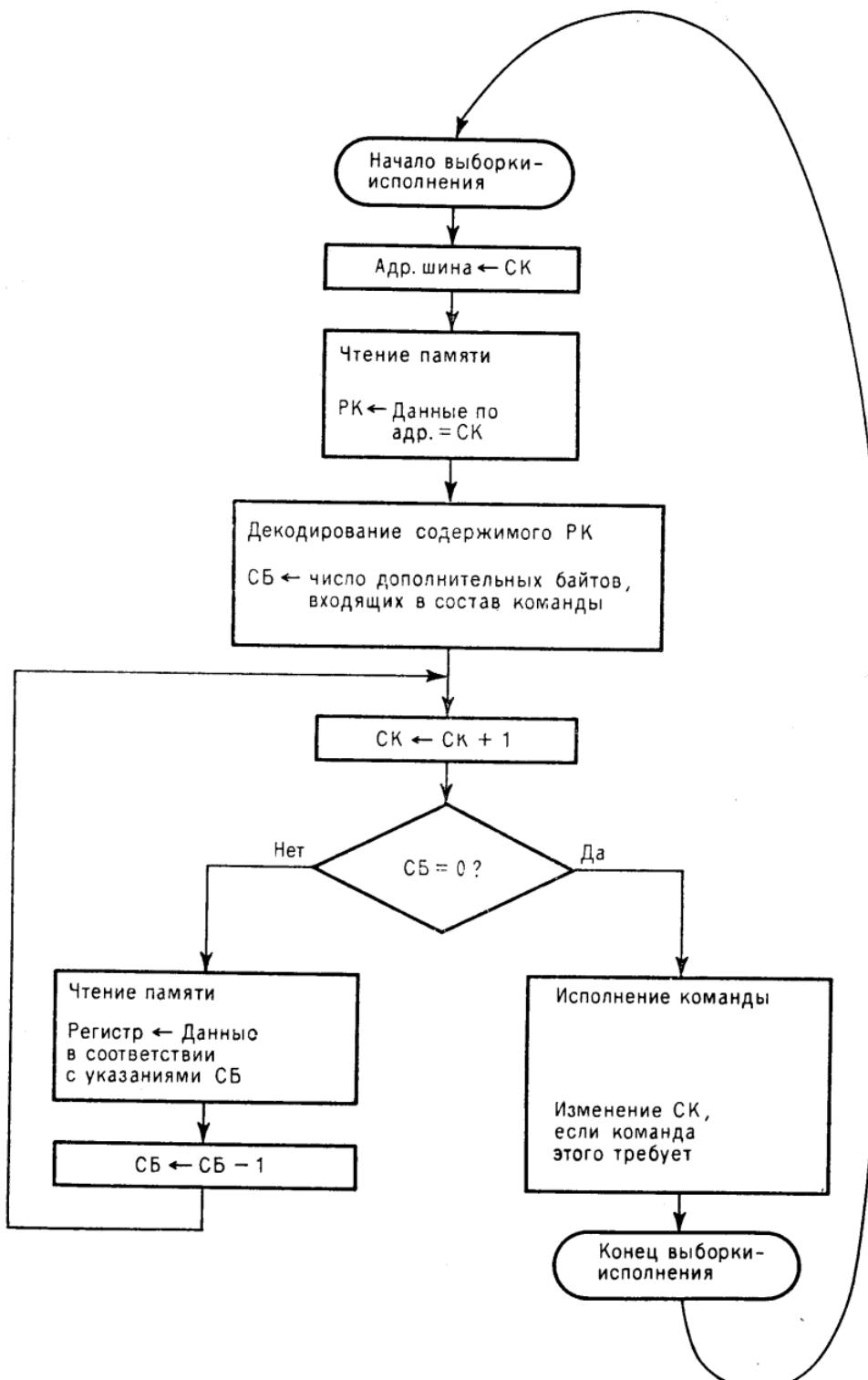


Рис. 2.21. Подробное представление цикла записи-исполнения.

Затем код операции расшифровывается и в соответствии с ним находящемуся в устройстве управления счетчику байтов (СБ) присваивается значение, указывающее, сколько еще байтов должно быть считано из памяти для полного формирования команды. Содержимое счетчика команд увеличивается на единицу для образования нового адреса. Если значение СБ равно нулю, что говорит об отсутствии необходимости считывания дополнительных байтов для формирования команды, происходит исполнение команды. Если при этом содержимое счетчика команд не изменяется, то оно укажет на адрес команды, которая должна выполняться следующей. В том случае, когда исходное значение счетчика байтов не равно нулю, данные, находящиеся в памяти по адресу, который содержится в счетчике команд,читываются в буферный или какой-то другой регистр и производится декремент СБ. Количество циклов, выполняемых по левой кольцевой ветви схемы, определяется начальным значением счетчика байтов, которое присваивается СБ в результате декодирования первого байта команды. В соответствующей точке цикла осуществляется инкремент счетчика команд, в котором после этого оказывается адрес, следующий за адресом текущей команды. Когда счетчик байтов обнуляется, опять происходит исполнение команды, после чего процессор приходит в состояние готовности к получению следующей команды, как было описано выше.

Команды, которые изменяют значение счетчика команд, могут заменить его содержимое либо на данные, взятые из других регистров, либо на два информационных байта, считываемых из памяти после кода операции, либо на слово, извлекаемое из стека. Стек, кроме того, может быть использован для запоминания содержимого счетчика команд перед занесением в этот счетчик новой информации.

Чтобы составить более полное представление о работе микропроцессора, целесообразно рассмотреть выполнение некоторых команд. Команды, широко используемые при написании тестовых программ, объединяются в следующие группы¹⁾:

- (а) команды арифметических и логических операций;
- (б) команды пересылки данных;
- (в) команды переходов;
- (г) команды обращения к подпрограммам, обработки прерываний и действий со стеком;
- (д) другие команды.

¹⁾ Здесь, по сути дела, дана наиболее общая классификация команд (не только применяемых для составления тестов) микропроцессоров с фиксированными наборами команд. — Прим. перев.

(а) Команды арифметических и логических операций

В число команд этой группы входят сложение, вычитание, команды логических операций (И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ), сравнения, инкременты, декременты и циклические сдвиги. Сюда же можно отнести команды побитовой проверки, имеющиеся у Z80.

Операция сравнения служит для сопоставлений значения данных с константой или другими данными, находящимися в регистре. Исходные данные в результате выполнения операций сохраняются. Следовательно, различные виды команд сравнения могут быть использованы в составе программ тестирования памяти для контроля ее способности сохранения данных. Типичная последовательность действий при таком контроле имеет следующий вид:

- 1) запись данных в память;
- 2) выдержка короткого интервала времени;
- 3) чтение данных из памяти;
- 4) сравнение считанных данных с исходными.

Примечание: в наборах команд некоторых микропроцессоров имеются команды, при выполнении которых обрабатывается или используется содержимое пар регистров. В приводимых ниже примерах пары регистров H и L, D и E процессоров Z80 и 8085 обозначены соответственно как пары HL и DE. Это просто регистры общего назначения, входящие в набор регистров микропроцессора, которые могут использоваться как 16-битовые комбинации. Примеры операций сравнения:

CMP #055H	;в 6502 сравнение содержимого регистра A с числом 55H
CMP B	;в 8085 сравнение содержимого регистров A и B
CP (HL)	;в Z80 сравнение содержимого регистра A с данными из памяти по адресу, находящемуся в регистровой паре HL

Примечание: когда результат сравнения указывает на равенство, устанавливается флаг нуля.

Отдельные биты могут быть проверены с использованием команды AND или команд BIT в случае микропроцессора Z80. Необходимо помнить, что при использовании команды AND исходные данные не сохраняются, поэтому их необходимо перед выполнением операции где-то записать.

AND # 02 ;6502, результат равен 0, если бит 1 регистра A
;не равен 1

Операции инкремента и декремента служат для выполнения счета при управлении перемещающимися указателями памяти. Повторяющийся декремент регистра до достижения нулевого

значения его содержимого применяется в качестве одного из способов реализации задержки.

Примеры:

DEX	;6502, декремент X, при достижении результатом нулевого
INX H	;значения устанавливается флаг нуля
DEC DE	;8085, инкремент указателя, находящегося в паре HL, на 1
	;Z80, уменьшение 16-разрядного числа в HL на 1

Циклические сдвиги могут быть использованы для формирования сдвигаемых наборов битов, загружаемых в память при ее тестировании. С помощью циклических сдвигов можно, кроме того, организовать проверку битов слова; для этого биты циклически сдвигаются в разряд переноса, где их значения можно проанализировать. Команды циклических сдвигов входят в стандартные подпрограммы, предназначенные для преобразования данных из последовательной формы представления в параллельную и обратного преобразования.

(6) Пересылка данных

С помощью команд пересылки осуществляются пересылки данных из одного регистра в другой, из регистров в память и из памяти в регистры. Особенно удобны команды пересылки, позволяющие пересылать данные в последовательно расположенные ячейки памяти и из таких ячеек. Для их реализации в 6502 используются режимы индексной адресации, а в Z80 и 8085 — режимы регистровой косвенной адресации.

Примеры:

;6502	INX	
	STA 0500H,X	;запись содержимого регистра A по адресу 500+X
;8085	INX H	
	MOV M,A	;запись содержимого регистра A по адресу, указывающему парой HL (на один адрес вперед по отношению к предыдущей передаче)

Команды ввода-вывода служат для пересылки данных в порты или из портов в системах, где предусмотрена портовая адресация. Если микропроцессор не располагает отдельной адресацией портов (как, например, 6502 и 6800) или аппаратно реализовано отображение портов ввода-вывода в адреса памяти (как в Z80 и 8085), то для ввода-вывода применяются команды с обычной адресацией памяти.

Примеры:

OUT 20H	;8085, вывод содержимого регистра A в порт 20H
OUT(PORTA),A	;Z80, вывод содержимого регистра A в порт A

В микропроцессоре Z80 предусмотрена кроме относительно простых команд ввода-вывода, имеющихся у 8085, возможность блочного вывода информации. Пример:

OUT1 ;содержимое ячейки памяти с адресом, находящимся в HL,
;выводится в порт, адрес которого находится в регистре C,
;и производится декремент регистра В

(в) Команды переходов

Эти команды дают программисту возможность организовывать ход выполнения программы в зависимости от результатов каких-то проверок, а также строить программные циклы. При условных переходах адрес команды, которая должна выполняться следующей, определяется на основании анализа состояния флагов.

Примеры:

:6502	DEX	;декремент регистра X
	BNE LOOP1	;переход к циклу LOOP1 и повторение цикла до тех ;пор, пока в результате команды DEX не сформируется ;нулевой результат
:8085	RAR	;правый циклический сдвиг A
	JNC LOOP2	;переход к циклу LOOP2 и выполнение его до по- ;явления переноса в результате сдвига

Возможности условных переходов у микропроцессора 6502 ограничены, так как при их выполнении используется относительная адресация. В 8085 и Z80 условный переход может производиться в любой ячейке памяти, однако в Z80 возможны и относительные переходы.

:Z80	DEC B	
	JR NZ LOOP3	;переход к циклу LOOP3 и повторение ;его, пока не будет нулевым результат ;DEC B

Команды безусловного перехода позволяют обойти при выполнении участки программы и организовывать постоянные циклы.

(г) Обращение к подпрограммам, обработка прерываний и действия со стеком

Фрагмент программы, который должен быть использован неоднократно, можно хранить в памяти в одном экземпляре и входить в него из разных точек программы с помощью команды обращения к подпрограммам

JSR адрес	:6502, обращение к подпрограмме
CALL адрес	:8085, Z80, обращение к подпрограмме

Применение такой команды требует наличия ЗУПВ, так как адрес команды, которая должна выполняться следующей (находящийся в счетчике команд), заносится в стек, после чего выполнение программы продолжается, начиная с адреса, указанного в текущей команде. В подпрограмме имеется команда, дополняющая команду обращения; в соответствии с ней восстанавливается значение счетчика команд, имевшее место перед обращением к подпрограмме, и возобновляется нормальная последовательность выполнения программы, из которой произошло обращение

RTS ;6502, возврат из подпрограммы
RET ;8085, Z80, возврат

Прерывания — это входные сигналы, по которым процессор обращается к какой-то подпрограмме сразу же по завершении выполнения текущей команды. Такая подпрограмма обычно называется программой обслуживания прерывания; в ее составе имеется команда, обеспечивающая возврат управления прерванной программе.

RTI ;6502, возврат из программы обработки прерывания
RET ;8085, Z80
RETI ;Z80, возврат из прерывания
RETN ;Z80, возврат из немаскируемого прерывания

Иногда прерывания реализуются, если только это разрешено с помощью специальной команды

EI ;8085, Z80, разрешение прерываний
CLI ;6502, разрешение прерываний

Существуют и соответствующие команды для запрещения прерываний: DI и SEI. Однако некоторые прерывания, называемые немаскируемыми, не могут быть запрещены. Для определения адресов программ обслуживания немаскируемых прерываний в микропроцессоры 8085 и Z80 должна быть введена соответствующая информация.

Для работы со стеком служат команды записи в стек и извлечения из стека. В микропроцессоре 6502 по этим командам каждый раз записывается в стек или считывается из него один байт. Например,

PHA ;6502, занесение содержимого регистра A в стек
PLA ;6502, считывание содержимого регистра A из стека

В микропроцессорах 8085 и Z80 при выполнении аналогичных команд пересыпается одновременно по два байта. Например,

PUSN PSW ;8085, занесение в стек содержимого регистра A и значений флагов
PU SN HL ;Z80, занесение в стек содержимого регистров H и L

При выполнении первой из этих команд в стек заносятся содержимое регистра А и значения флагов, для чего в стеке открываются две ячейки.

(д) Другие команды

В наборе команд любого микропроцессора имеется команда, по которой не выполняются никакие действия, кроме выборки команды из памяти и инкремента счетчика команд, — так называемая команда NOP (Нет операции). Команды NOP используются при тестировании микропроцессоров в режимах свободного прогона, причем команда подается на внешние выводы микропроцессора, отсоединенные от системной шины, путем соответствующей коммутации. Кроме того, команды NOP применяются для образования коротких задержек между операциями для выравнивания времени выполнения различных ветвей программы.

Существуют команды, которые воздействуют только на флаги и на регистр А. (Заметим, что для этих целей могут быть использованы и некоторые из команд, рассмотренных выше.) Примеры таких команд:

CLC	;6502, сброс бита переноса
SEC	;6502, установка бита переноса
STC	;8085, установка бита переноса
CMC	;8085, инверсия бита переноса
XOR A	;Z80, сброс аккумулятора и бита переноса
OR A	;Z80, сброс бита переноса

Как Z80, так и 8085 имеют команды загрузки содержимого пары HL в счетчик команд, при которой происходит переход по адресу, хранящемуся в этой паре.

Весьма удобна для организации циклов входящая в набор команд Z80 команда DJNZ. При ее выполнении производятся декrement регистра В и переход по относительному адресу, пока не устанавливается флаг нуля; после этого цикл прекращается.

Схемы ввода-вывода

Простейшие схемы, которые могут служить для осуществления ввода и выводов данных в микропроцессорной системе, — это буферные регистры и фиксаторы, подсоединеные кшине данных. Выбор нужного устройства может быть при этом осуществлен с помощью дешифратора, подключенного к адреснойшине. Пример такой схемы представлен на рис. 2.22.

Сейчас широко применяются специализированные устройства ввода-вывода, изготавливаемые многими фирмами. Эти устройства обычно снабжены регистрами, в которые может быть

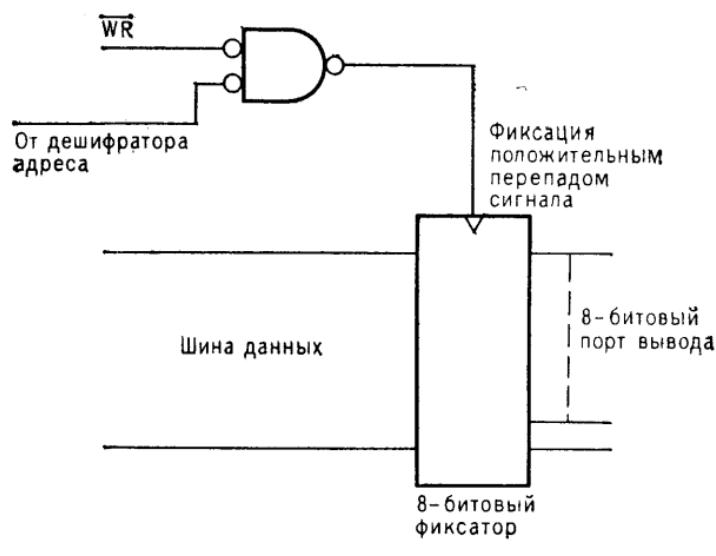
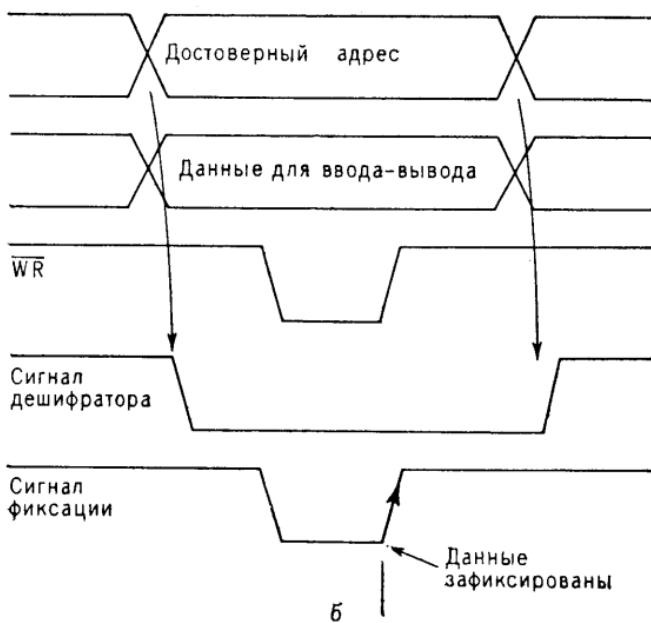
*a**б*

Рис. 2.22. Простая схема вывода: *а* — структура; *б* — временная диаграмма.

занесена информация, определяющая функции устройства. Возможность программирования функций ИС делает построенные на них устройства более гибкими в применении, чем в случае реализации на дискретных компонентах.

Типичное устройство ввода-вывода состоит из нескольких буферизированных портов, которые могут быть запрограммированы

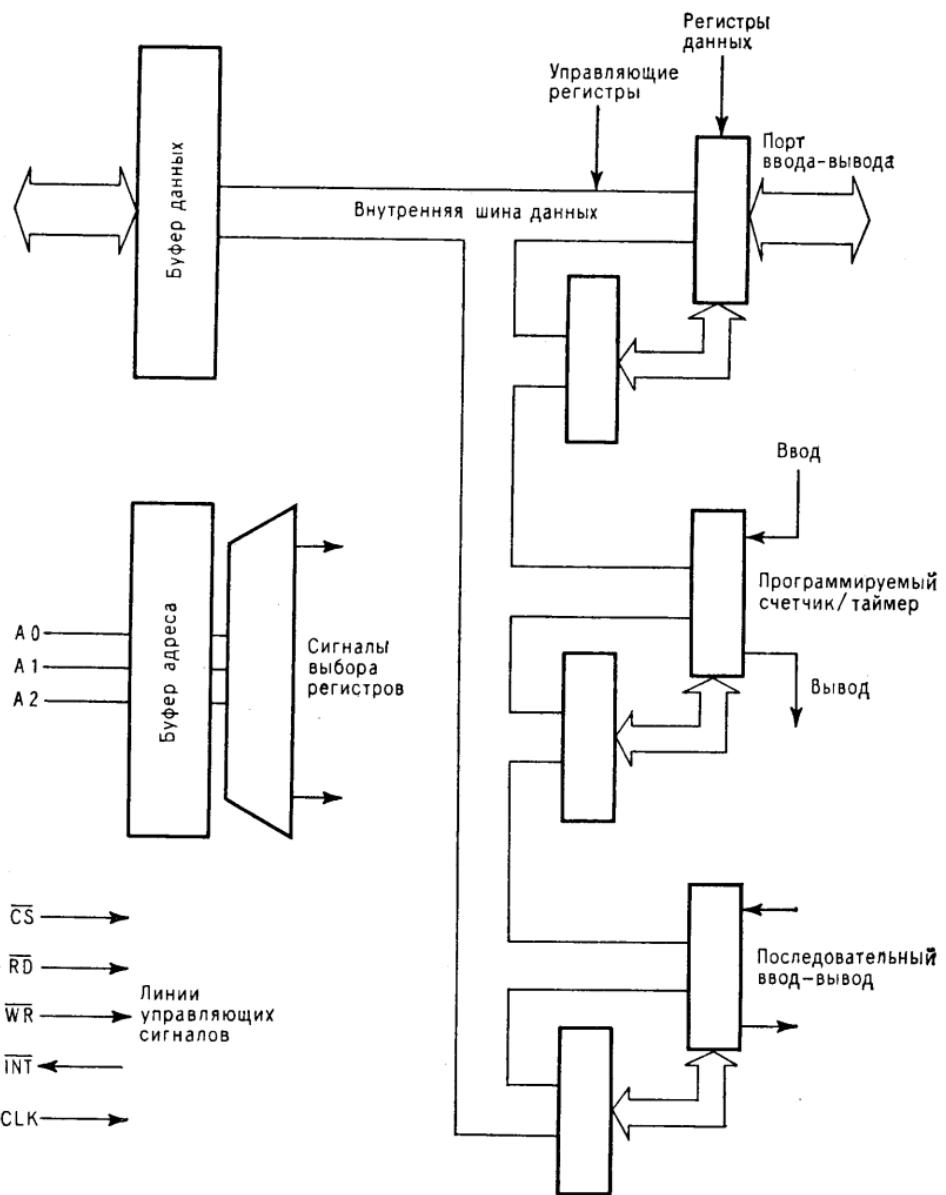


Рис. 2.23. Обобщенная структура устройства ввода-вывода.

ваны на выполнение ввода или вывода путем записи данных в один или несколько управляющих регистров. Иногда оказывается возможным совместить функции ввода и вывода в одном порте с подключенным к нему управляющим регистром. Каждый бит управляющего регистра задает характер использования соответствующего бита порта. Такие регистры иногда называют регистрами направления передачи данных.

В состав устройств ввода-вывода могут также быть включены счетчики и таймеры, служащие для выработки последовательностей импульсов из серий тактовых сигналов, которые могут представлять собой системные тактовые последовательности или какие-то их части. Таймеры обычно являются программируемыми и, будучи запущены, не требуют вмешательства со стороны процессора, в связи с чем отпадает необходимость в программной реализации задержек. Если по окончании какого-либо подсчета процессор должен выполнить определенные действия, схема может инициировать прерывание его работы.

В случае необходимости организации последовательной пересылки данных в состав устройств ввода-вывода могут быть включены программируемые схемы последовательного ввода-вывода. Развитие устройств ввода-вывода по своему характеру аналогично развитию самих микропроцессоров, т. е. новые устройства являются многофункциональными в противоположность более ранним, ориентированным на выполнение какой-то единственной функции. На рис. 2.23 показана типовая структура комбинированной схемы, предназначенней для последовательного и параллельного ввода-вывода и содержащей счетчики-таймеры. Относительно более специализированные схемы подобного типа используются в составе таких устройств, как клавиатуры, дисплеи, контроллеры дисководов.

Микропроцессоры, в которых применяется мультиплексирование шины адресов/данных, обычно комплектуются системными устройствами памяти и ввода-вывода. Для выбора того или иного из этих устройств процессор использует управляющую линию (линии) ВВОДА-ВЫВОДА/ПАМЯТИ. Однокристальные микропроцессоры содержат встроенные устройства ввода-вывода и могут быть использованы в системах, где критичным является минимальное число компонентов.

Заключение

Микропроцессоры являются шинно-ориентированными устройствами, которые обмениваются информацией с другими по параллельным линиям, объединенным в так называемые шины. Подобный характер взаимосвязи с внешним миром требует нового подхода к тестированию. С точки зрения обслуживания микропроцессоров были рассмотрены выше основные моменты его функционирования. Работа регистров, памяти и организация выполнения программы проанализированы как основа для ознакомления с наиболее важными методами обслуживания.

Глава 3

Интерфейс

Поскольку микропроцессор предназначен для выполнения полезных функций в реальном мире, он не может существовать изолированно, т. е. являться лишь неким устройством обработки двоичной информации. Должна существовать возможность вводить в него данные, над которыми может выполняться программа, и выводить любые результаты в форме, позволяющей интерпретировать их или использовать для управления системой, в которую микропроцессор входит в качестве составной части.

В том случае, когда требуется взаимодействие микропроцессора с человеком-оператором, для ввода может служить клавиатура, выдающая в микропроцессор двоичные данные, а для вывода — шестнадцатеричный индикатор, с помощью которого двоичная информация представляется в виде, удобном для восприятия оператором.

В более сложных ситуациях микропроцессор получает данные от нескольких источников, включая клавиатуры, датчики, устройства хранения данных и другие микропроцессоры, а вывод осуществляется на такие устройства, как дисплеи, запоминающие и другие вычислительные блоки, управляющие элементы системы.

Передача данных в микропроцессор и из него может выполняться одним из двух способов — параллельно или последовательно. При параллельной пересылке для каждого передаваемого бита данных имеется отдельная физическая линия, а при последовательной пересылке данные передаются по единственной линии бит за битом.

Так как при обоих видах передачи микропроцессор соединяется с внешними по отношению к нему устройствами, нарушение функционирования линий связи может неблагоприятным образом сказываться на работе системы на обеих сторонах этих линий. Отказы могут, кроме того, иметь место в логических схемах передачи и приема, а также в других составных частях микропроцессорной системы. Чтобы разобраться в последствиях

отказов в таких ситуациях, следует иметь представление о средствах интерфейса.

Все производители микропроцессоров изготавливают различные параллельные интерфейсные устройства, известные под такими названиями, как периферийно-интерфейсный адаптер, устройство параллельного ввода-вывода, универсальный интерфейсный адаптер. Существуют ИС, в состав которых входят устройства памяти и ввода-вывода, а некоторые однокристальные микропроцессоры содержат наряду с ПЗУ и ОЗУ внутренние схемы ввода-вывода.

Отличительным признаком аппаратуры параллельного ввода-вывода является наличие одного или нескольких регистров, называемых портами; причем каждый разряд регистра соединен с внешним выводом ИС, который в свою очередь может быть связан с контактами других устройств. Кроме того, обычно имеется ряд дополнительных регистров, служащих для управления вводом-выводом, например для задания направления передачи данных между процессором и внешним устройством.

Параллельный интерфейс

При использовании этого типа интерфейса данные пересыпаются в микропроцессор и из него аналогично тому, как это происходит при обмене информацией между процессором и памятью. Иногда шина данных снабжена рядом 8-битовых фиксаторов, к которым могут подключаться внешние устройства.

Интерфейс микропроцессорных шин адреса и данных представляет собой специальные аппаратные средства, в которые или из которых можно осуществлять пересылку данных с помощью команд чтения/записи памяти и чтения/записи ввода-вывода. Обычно для задания режима работы интерфейсного устройства используются команды, задающие направление передачи данных и другие параметры пересылки.

На рис. 3.1 приведена типичная структура простого параллельного устройства ввода-вывода, состоящего из управляющего регистра и единственного порта. Данные, загружаемые в управляющий регистр, определяют для каждого внешнего контакта порта, используется этот контакт для ввода или для вывода. С помощью специального входного сигнала указывается, будет ли передача осуществляться между процессором и портом или между процессором и управляющим регистром. Выбор устройства передачи осуществляется выходными сигналами дешифратора, подключенного к адреснойшине или к шине адресов ввода-вывода, причем одновременно не может быть выбрано более одного устройства. Подобные устройства могут применяться для считывания состояния переключателей и для вывода

информации на элементы с двумя состояниями, такие, как световые индикаторы, реле, пускатели.

При более сложных пересылках данных в схеме ввода-вывода обычно используются сигналы, характерные для режимов обмена с квитированием: сигнал готовности данных (ДГ) и сигнал-квитанция (КВ), служащий для подтверждения приема.

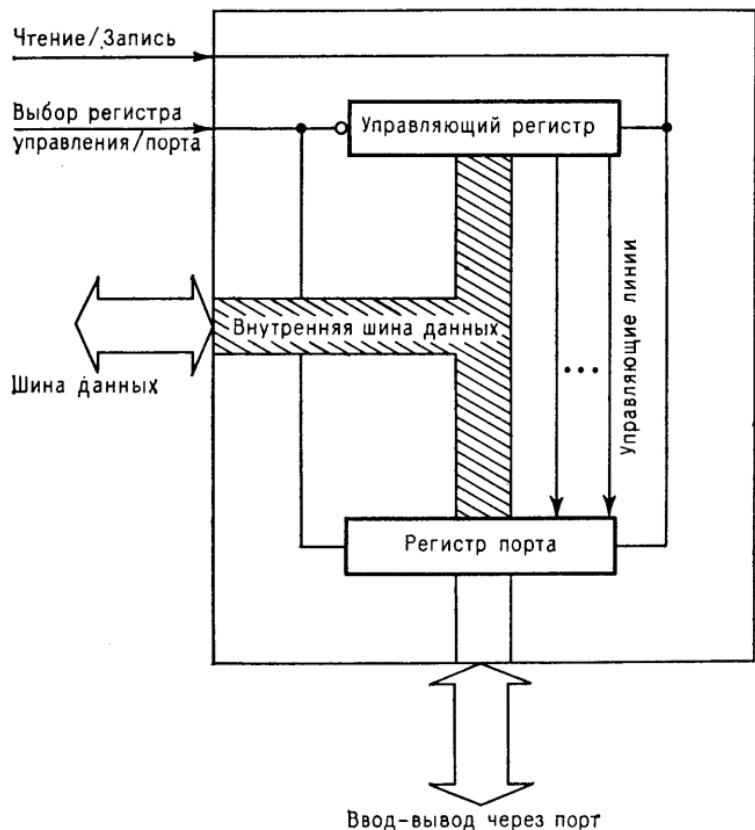


Рис. 3.1. Простое устройство параллельного ввода-вывода.

На рис. 3.2 показано применение этих сигналов для управления передачей данных между двумя системами. Источник передачи изменяет значения данных на выходных линиях и после небольшой задержки, в течение которой происходит установка этих значений, информирует приемник с помощью сигнала ДГ о том, что данные для пересылки готовы. Приемник воспринимает данные и сообщает об этом посредством короткого сигнала положительной полярности КВ. Сигнал КВ может быть использован в источнике передачи для сброса сигнала ДГ, а в некоторых случаях и для выработки сигнала прерывания, сообщающего процессору о возможности вывода очередного элемента данных.

Интерфейс Centronics

Существует стандартный параллельный интерфейс, называемый интерфейсом Centronics, предназначенный для передачи данных из микропроцессорных систем на устройства печати (принтеры). В этом интерфейсе используется обмен с квитированием, аналогичный описанному выше, с той разницей, что активным значениям сигналов соответствуют их низкие уровни.

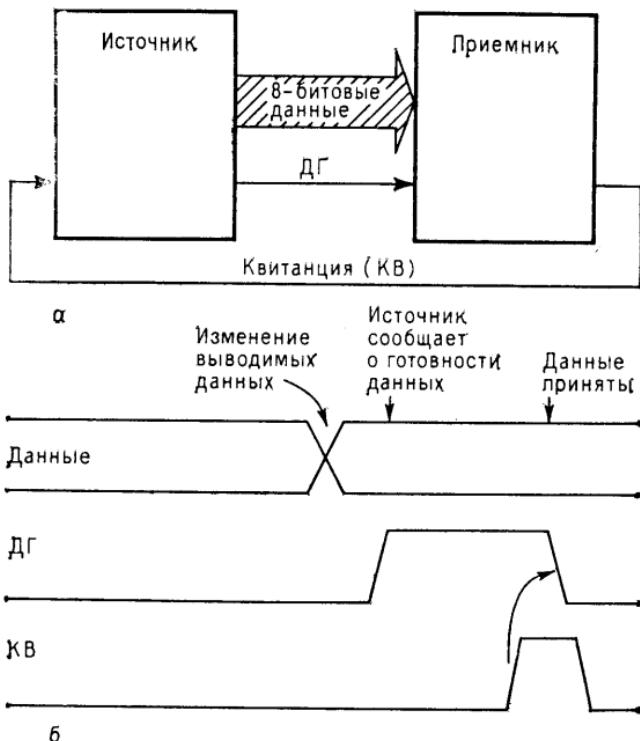


Рис. 3.2. Квитирование при параллельных пересылках: а — блок-схема пересылки; б — временная диаграмма.

Кроме двух сигналов квитирования интерфейс передает другие сигналы для управления принтером.

Однако простейшая разновидность интерфейса Centronics содержит семь или восемь линий данных и две линии квитирования или управления: стробирования (\overline{STR}) и подтверждения приема (\overline{ACK}). Каждая сигнальная линия имеет собственный обратный провод и подключается к определенному контакту стандартного разъема. Организация связей для этого простого случая показана на рис. 3.3. Она позволяет применять кабели, содержащие скрученные пары, для повышения помехоустойчивости.

Сигналам D0—D7 соответствуют обратные провода 20, 21, ..., 27, а сигналам STR и ACK—проводы соответственно 19 и 28. Последовательность передачи данных складывается из следующих этапов:

- 1) источник устанавливает данные на линиях D0—D7;
- 2) после установки стабильного значения данных выдается сигнал STR;
- 3) положительный фронт сигнала STR инициирует передачу данных в приемник (через логические схемы приемника);
- 4) когда приемник готов к получению новых данных, он выдает сигнал ACK, после положительного фронта которого может быть подан новый сигнал STR.

Рис. 3.3. Базовая структура интерфейса Centronics.

Эта последовательность иллюстрируется рис. 3.4. Временные соотношения сигналов различны в разных системах.

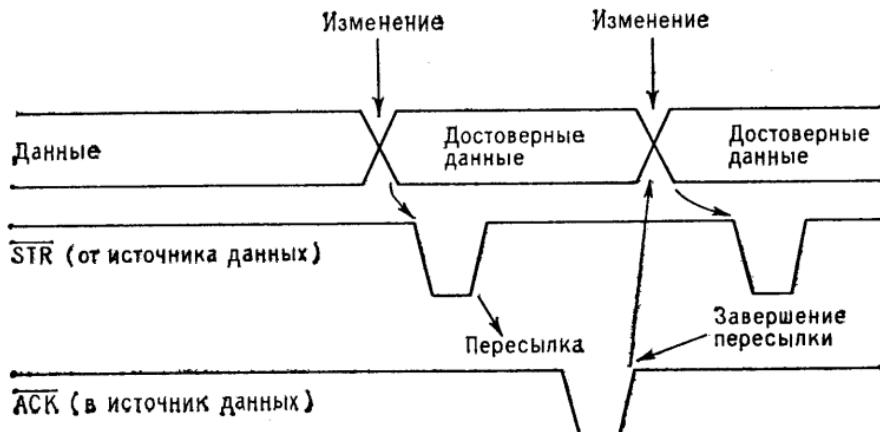


Рис. 3.4. Временные диаграммы интерфейса Centronics.

Описанная простая система передачи не предусматривает многих возможных ситуаций. Так, например, приемное устройство может оказаться отключенным, а в принтере может кончиться бумага. Для обеспечения функционирования интерфейса в подобных случаях интерфейсы Centronics снабжаются следующими дополнительными управляющими сигналами:

1. **BUSY** (Занятость) — сигнал, имеющий в активном состоянии высокий уровень и указывающий, что данные не могут быть приняты. Этот сигнал обычно формируется в системах с буферами символов, когда буфер заполнен. Контакт 11.

2. **PRIME** (Начало) — сигнал, имеющий в активном состоянии низкий уровень и служащий для инициализации логических схем приемника. Контакт 31.

3. **PAPER END** (Конец бумаги) (PE) — выходной сигнал принтера, имеющий в активном состоянии высокий уровень и информирующий о том, что кончилась бумага или что выключатель печатающего устройства находится в положении OFF («Выключено»). Контакт 12.

4. **SELECT** (Выбор) — сигнал с высоким активным уровнем, указывающий на то, что выключатель печатающего устройства находится в положении ON («Включено»); другое название сигнала — **DESEL**. Контакт 13.

5. **FAULT** (Неисправность) — сигнал с низким активным уровнем, приобретающий этот уровень, если кончается бумага, или оказывается выключенным печатающее устройство, или разомкнуты аварийные выключатели. Контакт 32.

Обратные провода сигналов принтера обычно присоединены к контактам 14, 16 и 33, а напряжение питания +5 В иногда подается на контакт 18. Выводы 34, 35 и 36 обычно не задействованы, но могут быть использованы для передачи нетиповых

Таблица 3.1. Сигналы интерфейса Centronics

Номер контакта	Сигнал	Номер контакта	Сигнал
1	<u>DATA STROBE</u> (<u>STR</u>)	19	Земля
2	Бит данных 1	20	»
3	Бит данных 2	21	»
4	Бит данных 3	22	»
5	Бит данных 4	23	»
6	Бит данных 5	24	»
7	Бит данных 6	25	»
8	Бит данных 7	26	»
9	Бит данных 8	27	»
10	<u>ACKNOWLEDGE</u> (<u>ACK</u>)	28	»
11	Занятость	29	»
12	Конец бумаги	30	»
13	Выбор	31	Начало
14	Земля	32	Неисправность
15	Пустой	33	Земля
16	Земля	34	{
17	Корпус	35	Не задействованы
18	+5 В	36	}

сигналов (например, постоянного напряжения 23 В как в одном широко применяемом принтере). В табл. 3.1 представлена разводка интерфейса Centronics.

Интерфейсная шина IEEE 488

IEEE 488 — это стандарт на универсальную интерфейсную шину, предназначенную для сопряжения ЭВМ между собой и для подключения таких устройств, как вольтметры, логические анализаторы, сигнатурные анализаторы, т. е. по сути дела любых приборов, характеристики которых отвечают условиям стандарта. Основную роль в становлении шинного стандарта сыграла фирма Hewlett Packard (этот интерфейс иногда называют ин-

Шина IEEE 488



Рис. 3.5. Пример организации набора абонентов шины IEEE.

терфейсной шиной Hewlett Packard — HPIB); она является обладателем патента на используемый в интерфейсе метод пересылки с квитированием.

Устройства, подключенные к шине, могут выполнять одну или несколько из следующих функций.

1. Контроллер. При включении питания только одно из устройств назначается системным контроллером. Это означает, что оно берет на себя роль активного контроллера, который управляет работой шины и позволяет вести передачу в любой момент времени только одному устройству. Системный контроллер может передавать функцию активного управления любому другому устройству, которое способно ее выполнять. Однако в любое время системный контроллер может привести шину в исходное состояние и вернуть себе активную роль. В схеме, представленной на рис. 3.5, функции системного и активного контроллеров выполняет микроЭВМ.

2. Передатчик. Любое устройство, способное передавать информацию по шине, называется передатчиком. В любой момент

времени только одно устройство может быть активным передатчиком. Из числа устройств, показанных на рис. 3.5, в качестве передатчиков могут использоваться вольтметр, устройство считывания с ленты и микроЭВМ.

3. Приемник. Устройство, которое может получать данные, называется приемником. Среди устройств, показанных на рис. 3.5, такими являются принтер, микроЭВМ и вольтметр.

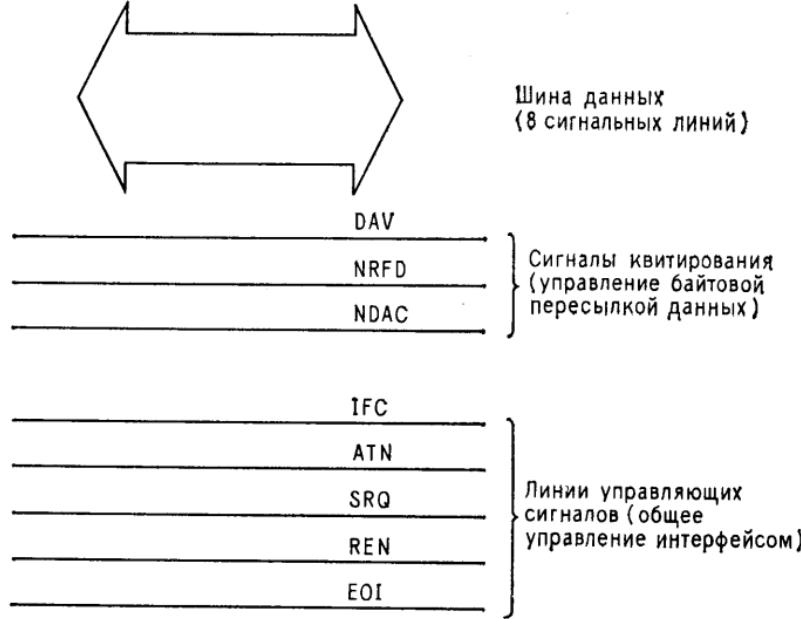


Рис. 3.6. Сигнальные линии универсальной интерфейсной шины IEEE 488.

Каждое устройство, подсоединенное к шине, имеет свой, отличный от других адрес; с помощью этих адресов активный контроллер узнает и выбирает устройства, между которыми осуществляется пересылка данных, т. е. определяет активный передатчик и активные приемники, а также все «непринимающие» устройства, которые не предназначены для приема сообщений. Идея выделения устройств, не принимающих данные, на первый взгляд может показаться странной. Дело в том, что пересылки по шине происходят асинхронно и скорость пересылки определяется быстродействием самого медленного из участвующих в ней устройств. Назначение каких-то устройств, не участвующих в конкретной пересылке, «непринимающими» позволяет избежать снижения скорости пересылки из-за участия в обмене медленных устройств, которым передаваемые данные не нужны. Так, например, и принтер, и дисковая память могут использоваться в качестве приемников, но если все обращения

к дисковой памяти выполнять достаточно медленно, чтобы передаваемые данные успевал принимать и принтер, эффективность системы будет невысока.

Собственно шина (рис. 3.6) состоит из восьми двунаправленных линий данных, трех линий квитирования байтовой пересылки и пяти линий общих управляющих сигналов, с помощью которых указывается состояние информации на линиях данных. По восьми линиям данных передаются как данные, так и информация управляющего и адресного характера в виде последовательности байтов, содержащиеся в которых биты пересылаются параллельно.

Все описанные выше сигналы интерфейса IEEE 488 имеют активные значения при низком уровне напряжения. (Отметим, что общепринято в наименования таких сигналов включать черточки, обозначающие инверсию, но в стандарте IEEE это правило не соблюдено.)

Так как некоторые из сигналов могут вырабатываться не обязательно одним устройством, для подачи их на шину применяются формирователи с открытым коллектором; например, сигналы NDAC, NPED и SRQ всегда должны передаваться на шину с помощью таких схем. Если при ответе на запрос обслуживания (SRQ) используется параллельный опрос, то выходные элементы на шину данных также должны быть схемами с открытым коллектором.

Остальные линии сигналов квитирования: ATN, IFC, REN, EOI, DAV и линии данных, если применяется последовательный опрос, могут управляться схемами либо с открытым коллектором, либо с тремя состояниями.

Для управления пересылкой через интерфейс данных и команд служат три сигнала квитирования:

DAV—DATA VALID (Данные достоверны)

Указывает, что на линиях данных находится достоверная информация.

NRFD—NOT READY FOR DATA (Не готов к приему данных)

Имеет высокий уровень, когда все приемники готовы к получению данных. Устройство, которое первым оказывается «не готово» к приему, придает низкий уровень сигналу на этой линии и запрещает источнику посыпать очередной байт. Снимает сигнал низкого уровня с линии самый медленный из подключенных к шине приемников.

NDAC—NOT DATA ACCEPTED (Данные не приняты)

Служит для указания, что все приемники восприняли информацию, посланную передатчиком. Если

NDAG имеет низкий уровень, значит, хотя бы один из приемников данные не получил. Снимает сигнал низкого уровня с линии самый медленный из приемников.

Последовательность осуществления пересылки с квитированием показана на рис. 3.7. Она включает следующие этапы:

1. Самый медленный из приемников готов к приему данных и прекращает подачу сигнала NRFD.

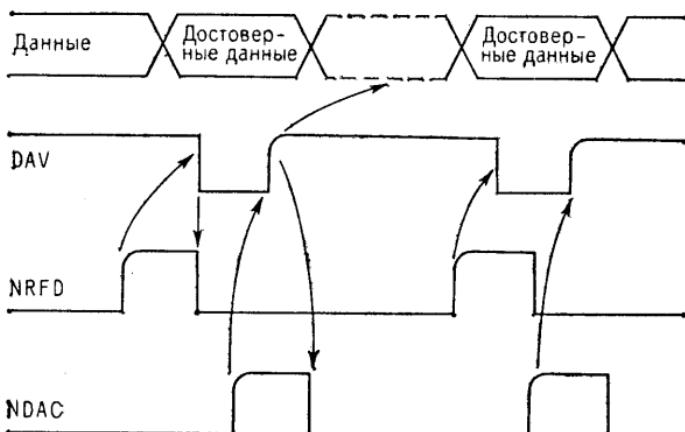


Рис. 3.7. Временные соотношения сигналов DAV, NRFD и NDAC.

2. Источник данных обнаруживает, что сигнал NRFD имеет высокий уровень, и, подав данные на шину, посылает сигнал DAV.

3. Приемники обнаруживают, что сигнал DAV имеет низкий уровень, и приводят в активное состояние сигнал NRFD, т. е. самый быстродействующий из них придает этому сигналу низкий уровень.

4. После получения данных приемники перестают подавать сигнал на линию NDAC. Она освобождается, когда этот сигнал перестает подавать самый медленный из приемников.

5. Источник обнаруживает прекращение сигнала NDAC и перестает подавать DAV.

6. Приемники вырабатывают сигнал NDAC, указывающий, что байт данных принят. Теперь может иметь место пересылка следующего байта.

Пять линий управления определяют и задают характер информации, передаваемой по восьми линиям данных, сообщая, например, представляет ли она собой данные или адрес.

ATN — ATTENTION (Внимание)

Эта линия, управляемая активным контроллером, служит для определения характера информации,

находящейся на линиях данных. Низкий уровень ATN указывает на то, что это адресная или управляющая информация, а высокий — что это данные.

IFC — INTERFACE CLEAR (Начальная установка интерфейса)

Системный контроллер использует эту линию для инициализации шины или возвращения себе управления ею (активный уровень — низкий).

SRQ — SERVICE REQUEST (Запрос на обслуживание)

Любое устройство, чтобы привлечь к себе внимание контроллера, придает сигналу на этой линии низкий уровень.

EOI — END OR IDENTIFY (Конец или идентификация)

Низкий уровень сигнала на этой линии указывает на окончание блока информации, пересылаемого по линиям данных.

REN — REMOTE ENABLE (Разрешение удаленным устройствам)

Этой линией управляет системный контроллер, а опрашивают ее приборы, которые могут управляться дистанционно.

МикроЭВМ HP85 в качестве контроллера

Персональный компьютер HP85, так же как Pet, Apple и многие другие широко распространенные микроЭВМ, может быть легко подключен к шине IEEE 488. Фирма Hewlett Packard выпускает «приставку», включающую интерфейс и набор ПЗУ, содержащих программное обеспечение, необходимое для работы шины. В этом разделе рассмотрено применение HP85 в качестве системного контроллера, управляющего шиной с помощью программы на языке Бейсик.

С помощью управляющей линии REM системный контроллер сообщает какому-либо прибору, что он становится объектом дистанционного управления. При использовании Бейсика это осуществляется с помощью оператора REMOTE. Например, оператор 10 REMOTE 716 назначает объектом дистанционного управления устройство по адресу 16. Интерфейс HPIB обычно размещается в сегменте адресации 7, поэтому и появляется цифра 7 в этой и других командах.

При желании можно заблокировать собственные органы управления удаленного устройства с помощью оператора LOCAL LOCKOUT и предотвратить тем самым любые вмешательства на время дистанционной пересылки. Например,

10 REMOTE 716
20 LOCAL LOCKOUT

Взяв на себя управление прибором, контроллер передает данные для задания режима работы (например, для мультиметра задаются режим измерения напряжения и диапазон измерения). Ясно, что соответствующие коды определяются типом удаленного прибора, но обычно они представляют собой последовательности символов ASCII. Набор кодов для широко распространенного прибора, носящего название Solartron Locator, приведен в табл. 3.2.

Таблица 3.2. Набор команд интерфейса для Solartron Locator

Код команды	Функция	Код команды	Функция
F00	Возврат в состояние готовности	F10	Постоянное напряжение
F01	Счет событий	F11	Постоянный ток
FC2	Счет передач	F12	Сопротивление
F03	Период/длительность	F13	Температура
F04	Задержка	F14	Самопроверка
F05	Частота	F15	Логический уровень
F06	Сигнатуры	F16	Переменное напряжение
F07	Сигнатуры линий	F17	Переменный ток
F08	Данные линий	F18	Запасной код

Оператор OUTPUT языка Бейсик служит для пересылки данных в выбранное устройство; например, оператор 30 OUTPUT 716; "0IF5" пересылает цепочку 0IF5 символов ASCII в устройство по адресу 16. Хотя этот оператор выглядит простым, он инициирует следующую последовательность действий интерфейсной шины:

1. Определяет все устройства как непринимающие (с гарантией исключает прием теми устройствами, которые его осуществлять не должны).
2. Назначает источник передачи (HP85).
3. Назначает приемник (прибор).
4. Пересылает данные.

Аналогичным образом оператор ENTER языка Бейсик позволяет контроллеру взять на себя роль приемника и получать данные. Так, оператор 50 ENTER 716; A\$ дает возможность контроллеру принимать данные из адреса 16 и присваивать их значение списковой переменной A\$. Интерфейсная шина при этом выполняет следующие действия:

1. Определяет все устройства как непринимающие.
2. Назначает приемник (HP85).
3. Назначает передатчик (прибор).
4. Пересылает данные.

Описанные выше простые операции представляют собой лишь небольшую часть возможностей шины IEEE 488. Однако, как мы

вскоре убедимся, этих операций достаточно для реализации дистанционного управления тестовым оборудованием, что позволяет организовать управление контрольной аппаратурой от ЭВМ.

Solartron Locator как приемник/источник

Прибор Locator 720 А подключается к шине IEEE 488 через адаптер внешнего интерфейса. С помощью малогабаритных переключателей интерфейса оператор может задать шинный адрес и статус прибора, т. е. назначить его приемником, источником или приемником/источником.

Все функции прибора, набор которых весьма широк — от возможностей простого мультиметра (тестера) до сигнатурного анализа, — могут быть выбраны и реализованы дистанционно через шину. Для выбора функции и инициирования измерений системный контроллер передает цепочку символов ASCII, представляющую собой последовательность кодов.

Приводимая ниже программа на Бейсике дает представление о том, как HP85 и Locator могут быть совместно использованы для непрерывного измерения частоты.

```

10 REMOTE 716
20 OUTPUT 716; "T1O1F5"
30 ENTER 716; A$
40 PRINT A$
50 GOTO 30

```

Командная цепочка состоит из трех кодов:

T1 — задает постоянную связь с прибором Locator для непрерывного приема;

O1 — разрешает вывод информации из прибора;

F5 — задает функцию прибора — измерение частоты.

Эти коды являются специфичными для прибора, а не для шины. С помощью подобной последовательности кодов можно управлять любым другим прибором, но при этом сами коды могут оказаться совершенно другими.

Результатом выполнения приведенной выше программы будет следующая информация на дисплее контроллера:

FREQUENCY	5000000 HZ
FREQUENCY	5000000 HZ
.	.
.	.
.	.

Если измерение должно быть не непрерывным, а осуществляться лишь в момент нажатия небольшой кнопки HOLD (фиксация),

имеющейся на логическом пробнике, можно использовать следующую программу:

```

10 REMOTE 716
20 OUTPUT 716; "T1O1R2F6"
30 GOSUB 100
40 PRINT B$
50 GOTO 30
60 END

100 REM WAIT FOR HOLD BUTTON ROUTINE
110 ENTER 716; A$
120 IF POS(A$, "TIMOUT") THEN 150
130 B$ = A$
140 GOTO 110
150 RETURN

```

В командную цепочку входит здесь команда R, служащая для формирования сообщения TIME OUT системному контроллеру, если кнопка HOLD нажата в течение 2 с. Стока подпрограммы 120 предназначена для просмотра цепочки символов, получаемой из прибора Locator, и обеспечивает возврат в основную программу только после того, как в результате получения сообщения TIMOUT в B\$ оказывается занесенным последнее измеренное значение частоты.

Применение шинного интерфейса IEEE 488 в различных приборах постоянно расширяется. Некоторые микроЭВМ, например OSBORNE, включают его в качестве стандартной составной части. В другие, такие, например, как IBM PC, Apple II, может быть установлена плата этого интерфейса. Это наиболее распространенный тип шины для соединения приборов в измерительных системах.

Последовательный интерфейс

Передача данных между двумя пунктами в параллельном виде, как это делается в интерфейсе IEEE, экономически эффективна только при небольших расстояниях. При значительном удалении связываемых точек стоимость кабелей и устройств буферизации существенно сказывается на стоимости системы в целом.

Рассмотрим, например, параллельный интерфейс принтера, содержащий три управляющие линии. Для него требуется кабель с не менее чем 12 проводниками; обычно, если принтер размещен вблизи источника данных, для этих целей используется кабель в резиновой изоляции. Если же, однако, принтер должен быть размещен на более или менее значительном расстоянии от

источника, например около 200 м, то используется более сложный кабель: как правило, он состоит при этом из скрученных пар с общим экраном, а если условия окружающей среды достаточно жесткие, то он должен быть армирован. На обоих концах кабеля должны быть установлены формирователи и приемники сигналов, чтобы обеспечить согласование линий и ис-

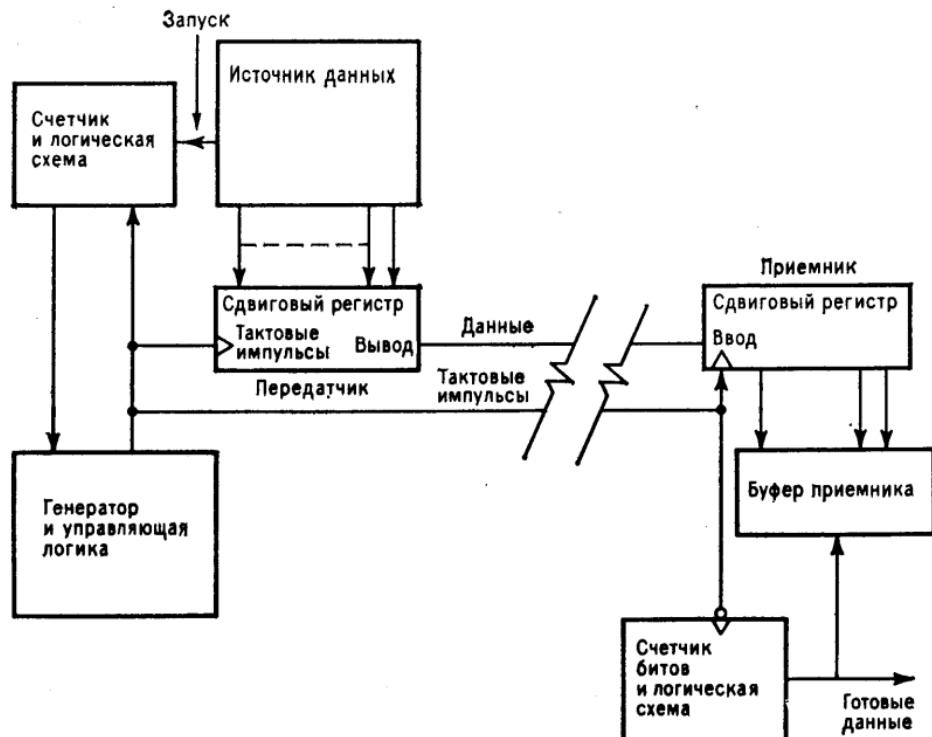


Рис. 3.8. Использование сдвиговых регистров для передачи данных по одной линии.

ключить влияние синфазных помех. Все это существенно усложняет аппаратуру передачи данных, хотя в некоторых случаях указанные меры и применяются.

Последовательный интерфейс позволяет избежать трудностей, связанных с необходимостью иметь отдельную линию для каждого из параллельно передаваемых битов; для передачи данных используется в случае последовательного интерфейса одна линия, по которой биты пересыпаются по одному последовательно (каждая линия данных должна быть также снабжена общим, или обратным, проводом). Чтобы разобраться в принципах этого метода передачи данных, рассмотрим его поэтапно, начиная со структурной схемы, представленной на рис. 3.8.

В начале передачи источник данных загружает информацию в сдвиговый регистр с параллельной записью, запускает схему тактирования и счетчик. Каждым тактовым сигналом данные сдвигаются на одну позицию вправо и поступают при этом на линию данных. Приемник состоит из еще одного сдвигового регистра, счетчика и логической схемы, которые управляются теми же тактовыми сигналами. После того как счетчик зарегистрирует поступление необходимого количества тактовых сигналов, он может инициировать параллельную передачу данных.

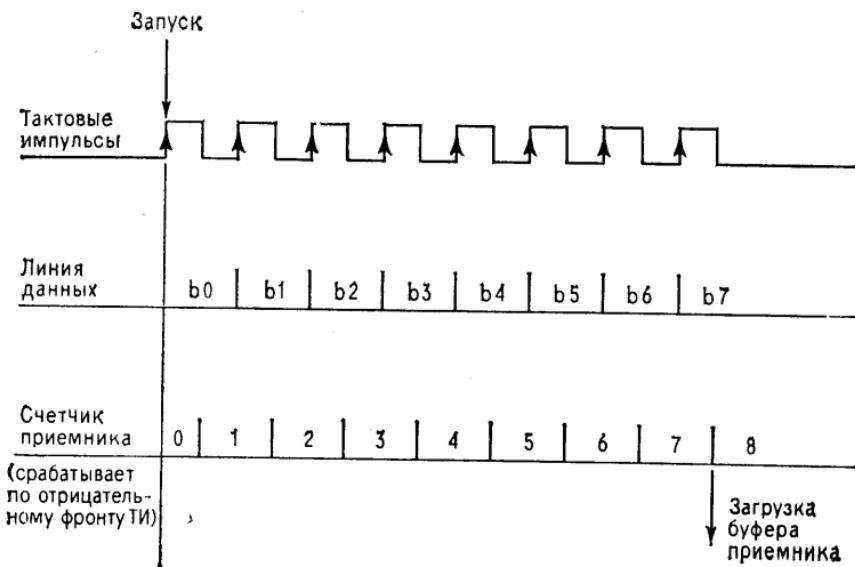


Рис. 3.9. Временные диаграммы простой системы последовательной передачи с использованием сдвиговых регистров.

из сдвигового регистра в буферный. Этот метод последовательной пересылки данных является одним из видов синхронной передачи, при которой вместе с данными требуется передавать тактовые сигналы. Временная диаграмма работы описанной системы приведена на рис. 3.9.

Уменьшив, таким образом, число сигнальных линий и снизив сложность схемы передачи, надо исключить из передаваемой информации тактовые сигналы и перейти к модифицированному варианту рассмотренной системы передачи, который уже не будет синхронным. Структура полученной в результате системы передачи представлена на рис. 3.10. Для ее функционирования необходимо соблюдение следующих условий:

- 1) оба генератора должны вырабатывать одинаковую частоту;
- 2) необходимо каким-то образом информировать приемник о том, что передача началась;

3) перед началом передачи очередного элемента данных должно быть отведено время для начальной установки приемника.

Чтобы выполнить первое из этих условий, целесообразно установить ряд стандартных частот передачи; асинхронная последовательная передача обычно ведется на одной из частот, принадлежащих этому ряду. Такую стандартизованную частоту

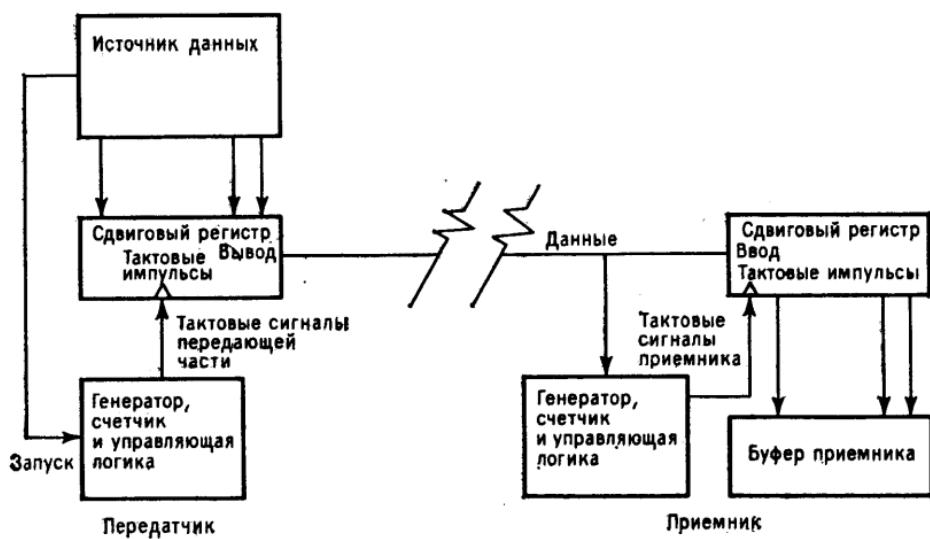


Рис. 3.10. Превращение системы передачи данных в асинхронную путем исключения линии тактирования.

принято называть скоростью передачи; скорость передачи изменяется в бодах, т. е. количеством битов, передаваемых за секунду по линии, когда происходит пересылка данных. В табл. 3.3 приведены наиболее широко применяемые скорости передачи и указаны области их использования.

Низкие скорости передачи (до 110 бод) применялись при работе с электромеханическими телепринтерами, такими, как Teletype 33, который работал со скоростью 110 бод; быстродействие подобных устройств ограничено параметрами их механических узлов. Относительно высокие скорости передачи используются при работе с дисплеями и каналами передачи данных, реализованными на полупроводниковой элементной базе, а средние скорости характерны для принтеров и других периферийных устройств, содержащих движущиеся механические части и сравнительно простые электронные схемы.

Два других условия успешного функционирования асинхронных систем передачи данных обусловливают стандартизацию компоновки блоков передаваемых данных: блок обычно содержит

Таблица 3.3. Стандартные скорости передачи данных и их применение

Скорость передачи, бит/с	Область применения
50	Электромеханические принтеры и устройства чтения
75	
110	
150	
300	Среднескоростные принтеры и устройства ввода данных
600	
1200	
1800	Быстродействующие принтеры, дисплеи, каналы передачи данных
2400	
3600	
4800	
9600	
19200	

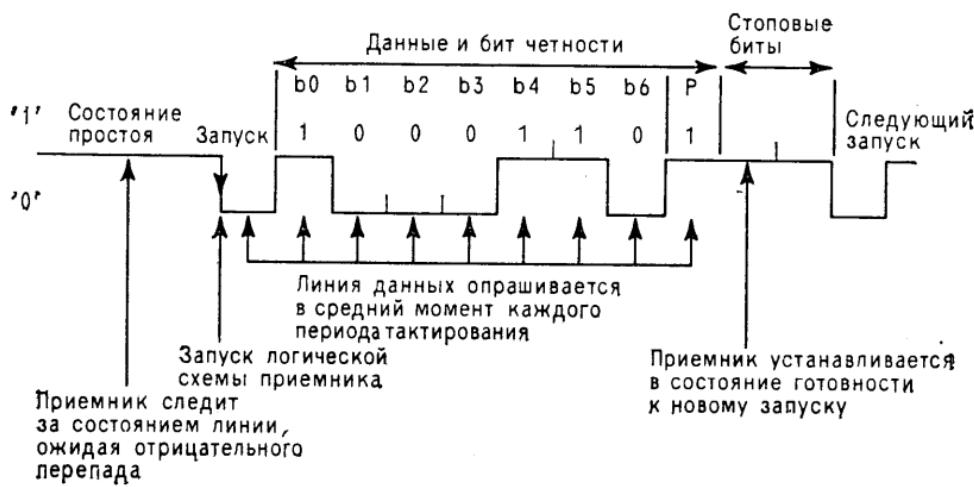
жит 7 или 8 бит, а пересылка каждого элемента данных начинается и заканчивается соответственно стартовыми и стоповыми битами. Более подробно асинхронная передача данных описана в следующем подразделе.

Асинхронная последовательная передача

Для уровней сигналов и схем, применяемых в асинхронной последовательной передаче данных, разработан ряд стандартов, причем ситуация в области этих стандартов весьма динамична, т. е. устаревающие стандарты постоянно заменяются новыми, более совершенными. Поэтому целесообразно рассмотреть асинхронную последовательную передачу вне зависимости от конкретных стандартов. Так как речь идет о передаче двоичных данных необходимо условиться о том, какие уровни электрических сигналов используются для передачи по линии данных логических значений 0 и 1.

Целесообразно считать, что данные передаются в виде их «истинных значений», т. е. если бит, который передается по линии, имеет значение 1, то и сигнал в линии имеет значение 1. (Электрические уровни сигналов, передаваемых на линии, могут не совпадать с уровнями, характерными для схемотехники ИС, на которых построены схемы параллельно-последовательного преобразования, потому что между логическими схемами и линиями передачи обычно имеет место то или иное преобразование уровней.) В соответствии с такой договоренностью, если линия пребывает в состоянии простоя, т. е. по этой линии не передается никаких данных, на ней имеет место сигнал 1.

Начало передачи каждого элемента данных (состоящего обычно из 7—8 бит) обозначается изменением значения сигнала в линии на 0 на один период тактовых сигналов; тем самым приемник информируется о том, что передача началась. Сразу же после этого стартового бита начинают передаваться биты данных, по одному за период тактирования, начиная с младшего бита. Если в процессе передачи производится контроль по четности, для чего должен быть передан бит четности (сумма значения которого со значениями битов данных должна быть либо



Данные 0110001 (число единиц в коде ASCII 1)

Рис. 3.11. Формат данных при асинхронной последовательной передаче.

четной, либо нечетной), то он передается после старшего бита данных.

Теперь линия данных должна в течение некоторого минимального времени, называемого стоповым интервалом, находиться в состоянии простоя. Это необходимо для приведения приемника в исходное состояние перед передачей нового элемента данных. Стоповый интервал обычно длится в разных системах: 1, 1,5 или 2 периода тактирования. Если новый элемент данных не посыпается сразу же по истечении этого времени, линия остается в состоянии простоя дольше длительности стопового интервала. Таким образом, при асинхронной последовательной передаче данных имеет место следующая очередность событий:

- 1) состояние простоя, т. е. значение 1;
- 2) один стартовый бит со значением 0;
- 3) 7 или 8 бит данных, начиная с младшего;
- 4) один бит четности;
- 5) 1, 1,5 или 2 стоповых бита, т. е. биты со значением 1, соответствующим состоянию простоя;
- 6) следующий стартовый бит со значением 0.

Блок данных в том виде, в каком его можно было наблюдать с помощью осциллографа, показан на рис. 3.11. Семь битов данных — это 0110001, вместе с битом четности они имеют вид 10110001. После передачи этих восьми битов следуют два стоповых бита, а сразу после окончания стопового интервала начинается передача следующего элемента данных.

Наиболее часто стоповый интервал состоит из одного бита (два бита использовались при работе с электромеханическими телетайпами, так как для их начальной установки однобитового интервала оказалось недостаточно). Таким образом, полное число битов в блоке, содержащем 8 или 7 бит данных и один бит четности, равно 10, и максимальная скорость пересылки составляет одну десятую от скорости передачи, измеряемой в бодах, т. е. при скорости передачи 1200 бод максимальная скорость пересылки данных 120 элементов/с.

Последовательно-параллельное и параллельно-последовательное преобразование

Последовательности битов данных, аналогичные рассмотренной выше, должны формироваться из данных, которые внутри системы представлены в параллельной форме. Устройство, принимающее данные в последовательной форме, должно, в свою очередь, быть способно вновь преобразовать их в параллельную форму.

Рассмотренные выше принципы последовательной передачи с использованием сдвиговых регистров, счетчиков и логических схем могут быть реализованы на базе ИС малой и средней интеграции (МИС и СИС). Так это практически и делалось до недавнего прошлого, и многие вычислительные системы, находящиеся сейчас в эксплуатации, содержат устройства передачи и приема данных в последовательной форме, конструктивно оформленные в виде плат с установленными на них МИС и СИС. Достижения технологии БИС позволяют реализовать функционально законченное устройство преобразования данных из параллельной формы в последовательную и обратного преобразования в виде одной ИС. Такие устройства иногда называют универсальными асинхронными приемопередатчиками (УАПП).

В микропроцессорных системах находит применение программный подход к решению задачи параллельно-последовательного и обратного преобразования. Можно написать программу, которая обрабатывает находящиеся в регистре данные таким образом, что они поступают на один из внешних контактов порта вывода в последовательной форме, будучи снабженены стартовыми и стоповыми битами. Возможно и обратное преобразование с использованием программы, которая следит за

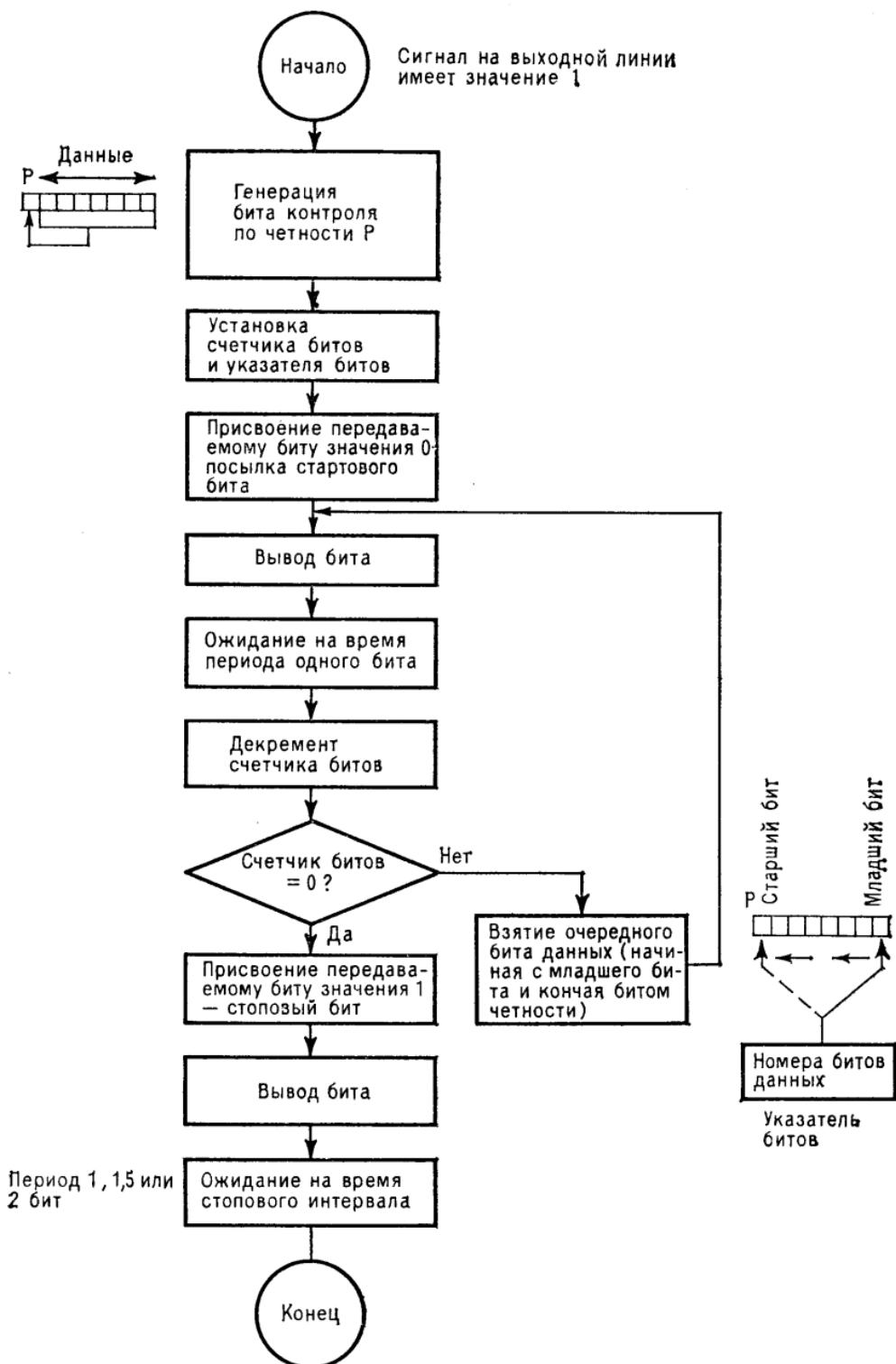


Рис. 3.12. Схема алгоритма передачи данных в последовательной форме.

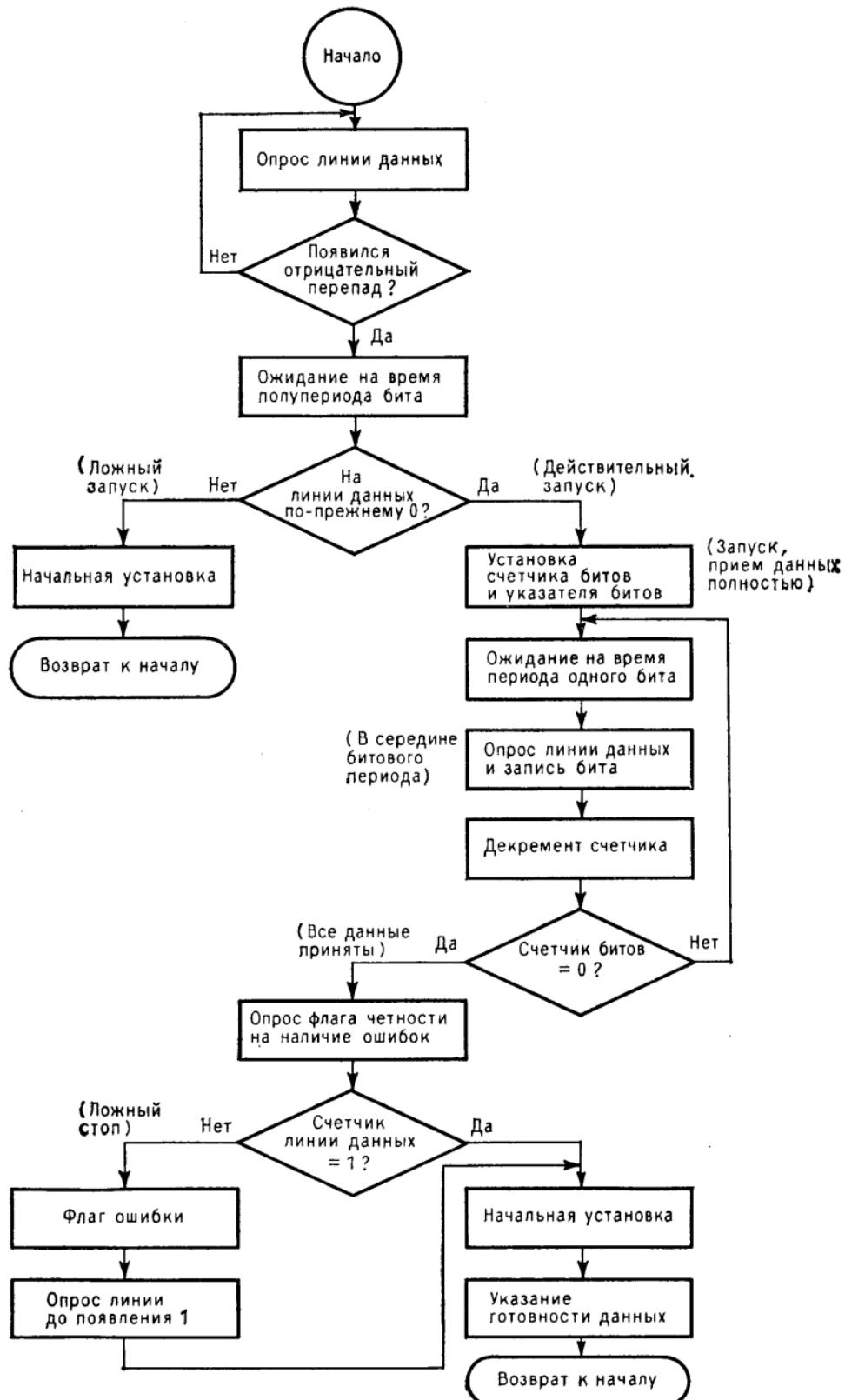


Рис. 3.13. Схема алгоритма приема данных в последовательной форме.

HEWLETT-PACKARD: АССЕМБЛЕР 8085
ИСХОДНЫЙ ТЕКСТ

ОБЛАСТЬ ПАМЯТИ	ОБЪЕКТ. КОД	НОМЕР СТРОКИ	
0000	F3	1	'8085'
0001	C5	2	OBTIM
0002	D5	3	STPTIM
0003	0608	4	
0004	3E40		
0005			
0006			
0007			
0008			
0009			
000A			
000B			
000C			
000D			
000E			
000F			
0010			EQU 0000
0011			EQU 0000
0012			; ЗНАЧЕНИЯ, НЕОБХОДИМЫЕ ДЛЯ
0013			; УДОВЛЕТВОРЕНИЯ ТРЕБОВАНИЙ
0014			; ТАКТОВОЙ ЧАСТОТЫ И СКОРОСТИ
0015			; ПЕРЕДАЧИ
0016		5	; ИМЯ;
0017		6	; ВХОДЫ:
0018		7	; ВЫХОДЫ:
0019		8	; ВЫЗОВЫ:
0020		9	; ИЗМЕНЕНИЯ: А, ФЛАГИ
0021		10	; ОПИСАНИЕ:
0022		11	ПОСЫЛАЕТ СИМВОЛ В ПОСЛЕДОВАТЕЛЬНОЙ ФОРМЕ
0023		12	НА КОНТАКТ SOD, ПРЕДВАРИЯ ЕГО СТАРТОВЫМ
0024		13	БИТОМ И СОПРОВОЖДАЯ СТОПОВЫМ БИТОМ
0025		14	SEROP:
0026		15	DI
0027		16	PUSH B
0028		17	PUSH D
0029		18	MVI B, 08
0030		19	MVI A, 01000000B
0031		20	LOOP:
0032		21	SIM
0033		22	LXI D, OBTIM
0034		23	CALL DELAY
0035		24	
0036		25	MOV A, C
0037		26	RAR
0038			
0039			
0040			
0041			
0042			
0043			
0044			
0045			
0046			
0047			
0048			
0049			
004A			
004B			
004C			
004D			
004E			
004F			
0050			
0051			
0052			
0053			
0054			
0055			
0056			
0057			
0058			
0059			
005A			
005B			
005C			
005D			
005E			
005F			
0060			
0061			
0062			
0063			
0064			
0065			
0066			
0067			
0068			
0069			
006A			
006B			
006C			
006D			
006E			
006F			
0070			
0071			
0072			
0073			
0074			
0075			
0076			
0077			
0078			
0079			
007A			
007B			
007C			
007D			
007E			
007F			
0080			
0081			
0082			
0083			
0084			
0085			
0086			
0087			
0088			
0089			
008A			
008B			
008C			
008D			
008E			
008F			
0090			
0091			
0092			
0093			
0094			
0095			
0096			
0097			
0098			
0099			
009A			
009B			
009C			
009D			
009E			
009F			
00A0			
00A1			
00A2			
00A3			
00A4			
00A5			
00A6			
00A7			
00A8			
00A9			
00AA			
00AB			
00AC			
00AD			
00AE			
00AF			
00B0			
00B1			
00B2			
00B3			
00B4			
00B5			
00B6			
00B7			
00B8			
00B9			
00BA			
00BB			
00BC			
00BD			
00BE			
00BF			
00C0			
00C1			
00C2			
00C3			
00C4			
00C5			
00C6			
00C7			
00C8			
00C9			
00CA			
00CB			
00CC			
00CD			
00CE			
00CF			
00D0			
00D1			
00D2			
00D3			
00D4			
00D5			
00D6			
00D7			
00D8			
00D9			
00DA			
00DB			
00DC			
00DD			
00DE			
00DF			
00E0			
00E1			
00E2			
00E3			
00E4			
00E5			
00E6			
00E7			
00E8			
00E9			
00EA			
00EB			
00EC			
00ED			
00EE			
00EF			
00F0			
00F1			
00F2			
00F3			
00F4			
00F5			
00F6			
00F7			
00F8			
00F9			
00FA			
00FB			
00FC			
00FD			
00FE			
00FF			

0010	4F	27		MOV C,A ; ЗАПОМИНАНИЕ РЕЗУЛЬТАТА
		28	;	;
0011	3E80	29		MVI A,1000000B ; УСТ-КА РЕГ.А ДЛЯ ВЫВОДА,
0013	1F	30		RAR ; СОДЕРЖИМОЕ РЕГ.А ТЕПЕРЬ 1000000B
0014	05	31		DCR B
0015	F20007	32		JP LOOP
0018	3EC0	33	;	
001A	30	34		MVI A,1100000B ; УСТ-КА РЕГ.А ДЛЯ ВЫВОДА,
001B	110000	35		SIM ; СТОП-БИТА
001E	CD0025	36		CALL SOD ; ВЫВОД НА SOD
		37	;	LXI D,STPTIM ; ЗНАЧЕНИЕ СТОПОВОЙ ЗАДЕРЖКИ
		38	;	CALL DELAY ; ОЖИДАНИЕ НА ЭТО ВРЕМЯ
0021	D1	39		POP D ; ВОССТАНОВЛЕНИЕ В,С,Д,Е
0022	C1	40		POP B ; РАЗРЕШЕНИЕ ПРЕРЫВАНИЙ
0023	FB	41		EI
0024	C9	42		RET
		43		DELAY
		44	;	ИМЯ:
		45	;	ВХОДЫ:
		46	;	ВЫХОДЫ:
		47	;	ВЫЗОВЫ:
		48	;	ИЗМЕНЕНИЯ:
		49	;	ОПИСАНИЕ:
		50	;	ДЕКРЕМЕНТ DE И ВОЗВРАТ В ВЫЗВАвшую
0025	1B	51	;	ПРОГРАММУ ПРИ DE = 0
0025	7A	52		DCX D ; ДЕКРЕМЕНТ DE
0026	B3	53		MOV A,D ; ПОДГОТОВКА D К СРАВНЕНИЮ С Е
0027	C20025	54		ORA E ; НУЛЕВОЙ РЕЗУЛЬТАТ, ЕСЛИ А=Е = 0
0028	C9	55		JNZ DELAY ; ВОЗВРАТ, ЕСЛИ НЕ НУЛЬ
0028		56		RET

Программа 3.1. Программа последовательного вывода с 16-битовой задержкой на языке ассемблера 8035.

HEWLETT-PACKARD: АССЕМБЛЕР 8085

ИСХОДНЫЙ ТЕКСТ

ОБЛАСТЬ ОБЪЕКТ-
ПАМЯТИ НЫЙ КОД

СТРОКИ

“8085”

EXT

SERIN

; ИМЯ:

ЛАННЫЕ ОТ SID В 8085

; ВХОДЫ:

ПРЕОБРАЗОВАННЫЕ ДАННЫЕ В РЕГИСТР A

; ВЫХОДЫ:

DELAY

; ВЫЗОВЫ:

DELAY

; ИЗМЕНЕНИЯ: А,ФЛАГИ

; ОПИСАНИЕ:

ПРОГРАММА ОПРАВЛЯЕТ КОНТАКТ SID И, ОБНАРУЖИВ

; ДОСТОВЕРНЫЙ СТАРТОВЫЙ БИТ, ВДВИГАЕТ ДАННЫЕ

; ОТ ЭТОГО КОНТАКТА В РЕГИСТР A, ГДЕ ФОРМИРУЕТСЯ

ПАРАЛЛЕЛЬНОЕ ПРЕДСТАВЛЕНИЕ СИМВОЛА

;

SERIN:

DI

; ПРЕРЫВАНИЯ ДОЛЖНЫ БЫТЬ

; ЗАПРЕЩЕНЫ

PUSH D

PUSH B

MVI B,00

MVI C,08

RDIP:

RIM

RAL

; ЧТЕНИЕ МАСКИ ПРЕРЫВАНИЙ

; В ТРИГГЕР ПЕРЕНОСА С ВХОДНЫМИ

JC RDIP

LXI D,HBIT

; ДАННЫМИ

; ВОЗВРАТ, ЕСЛИ НЕТ СТАРТ-БИТА

; ИНАЧЕ ОЖИДАНИЕ В ТЕЧЕНИЕ

; ПОЛУБИТА

DA0007

23

110000

24

000C

000F	CD0000	25	CALL DELAY	; ПРОВЕРКА НА ДОСТОВЕРНЫЙ
0012	20	26	RIM	; СТАРТ-БИТ
0013	17	27	RAL	; В ТРИГГЕР ПЕРЕНОСА С ДАННЫМИ SID
0014	DA0007	28	JC RDIP	; ВОЗВРАТ, ЕСЛИ НЕДОСТОВЕРНЫЙ
			; ЗАПУСК	
0017		29	LOOP:	
0017	110000	30	LXI D,BTIM	; ОДНОБИТОВАЯ ЗАДЕРЖКА
001A	CD0000	31	CALL DELAY	; ОЖИДАНИЕ ДО СЕРЕДИНЫ СЛЕДУЮЩ.
			; БИТА	
001D	20	32	RIM	; ПОЛУЧИТЬ БИТ
001E	17	33	RAL	; В ПЕРЕНОС С ЭТИМ БИТОМ
001F	78	34	MOV A,B	; ПРОМЕЖУТОЧНЫЙ РЕЗУЛЬТАТ В А
0020	1F	35	RAR	; СДВИГ ПЕРЕНОСА В СТАРШИЙ БИТ
0021	47	36	MOV B,A	; ПЕРЕСЫЛКА ОБРАТНО В В
0022	0D	37	DCR C	; ДЕКРЕМЕНТ СЧЕТЧИКА
0023	C20017	38	JNZ LOOP	
0026	110000	39	LXI D,STPTIM	; ЗНАЧЕНИЕ ЗАДЕРЖКИ СТАРТОВОГО
			; БИТА	
0029	CD0000	40	CALL DELAY	
002C	78	41	MOV A,B	; ДАННЫЕ В А
002D	C1	42	POP B	
002E	D1	43	POP D	
002F	FB	44	EI	; РАЗРЕШЕНИЕ ПРЕРЫВАНИЙ
0030	C9	45	RET	

Программа 3.2 Программа последовательного вывода на языке ассемблера 8085.

HEWLETT-PACKARD: АССЕМБЛЕР Z80

ОБЛАСТЬ ПАМЯТИ	ОБЪЕКТ-ПАМЯТИ	НОМЕР КОД СТРОКИ	ИСХОДНЫЙ ТЕКСТ	
	<code><0000></code>	1	"Z80"	
	<code><000C></code>	2	PORT A	EQU 00 ; ЗДЕСЬ НЕОБХОДИМО ЗНАЧЕНИЕ А
	<code><0018></code>	3	OPTIM	EQU 12 ; ДЛЯ ПОЛУЧЕНИЯ КОРРЕКТНОЙ
		4	STRTIM	EQU 24 ; ВЕЛИЧИНЫ СТОПОВОГО БИТА
		5		; В СООТВЕТСТВИИ СО СКОРОСТЬЮ
				; ПЕРЕДАЧИ
		6	; ИМЯ SEROP	
		7	; ВХОДЫ:	СИМВОЛ ДЛЯ ВВОДА В РЕГИСТР С
		8	; ВЫХОДЫ:	СИМВОЛ ДЛЯ ВО ПОРТА А
		9	; ВЫЗОВЫ:	НЕТ
		10	; ИЗМЕНЕНИЯ: А, ФЛАГИ	
		11	; ОПИСАНИЕ В	
		12	;	
		13	;	ПРОГРАММА ПОСЛЕДОВАТЕЛЬНОГО ВЫВОДА ДЛЯ Z80
		14	;	ИСПОЛЬЗУЕТСЯ БИТ 0 ИЗ ПОРТА 0
		15	;	СОСТОЯНИЕ ПРОСТОЯ = 1
		16	;	СИМВОЛ ДЛЯ ПОМЕЩЕНИЯ В С ВЫЗЫВАЮЩЕЙ ПРОГРАММОЙ
		17	;	ПРИ ВЫЗОВЕ ЛИНИЯ ДОЛЖНА БЫТЬ УСТАНОВЛЕНА В 1
		18	;	СИМВОЛ ПОМЕЩАЕТСЯ В С ВЫЗЫВАЮЩЕЙ ПРОГРАММОЙ
		19	;	
		20	SEROP:	
		21		DI
		22		PUSH BC
		23		PUSH DE
		24	;	
		25		LD B,08
	0000	F3		
	0000	C5		
	0001	D5		
	0002			
	0608			

0005	DB00	26		IN A,[PORTA] ; ЧТЕНИЕ ПОРТА ДЛЯ СОХРАН.
0007	E6FE	27	LOOP:	AND 1111110B ; СОДЕРЖИМОГО
0009	D300	28		OUT [PORTA],A ; ВЫВОД БИТА
0009	11000C	29		LD DE,OBTIM ; ЗАПИСЬ БИТОВОЙ ЗАДЕРЖКИ В DE
000B	CD0036	30		CALL DELAY ; ОЖИДАНИЕ НА ЭТО ВРЕМЯ
000E		31	;	
0011	79	32	;	LD A,C ; ВЗЯТИЕ СИМВОЛА ИЗ С
0012	0F	33		RRCA ; МЛАДШИЙ БИТ В ТРИГГЕР ПЕРЕНОСА
0013	4F	34		LD C,A ; ЗАПОМИНАНИЕ РЕЗУЛЬТАТА
0014	3807	35		SDVIGA В С
0016		36		JR C,SET1
0016	DB00	37	SET0:	
0018	E6FE	38		IN A,[PORTA]
001A	C30022	39		AND 1111110B ; УСТАНОВКА ВО В 0
001D	00	40		JP COUNT
001D		41	SET1:	
001D		42		NOP ; Для выравнивания времени
001E	DB00	43		; в таймерах
0020	F601	44		IN A,[PORTA]
0022	05	45	COUNT:	OR 00000001B ; УСТАНОВКА ВО В 1
0023	F20009	46		DEC B
0026	DB00	47		JP P,LOOP
		48		
		49		IN A,[PORTA]

```

0028 F601 50 OR 00000001B ; УСТАНОВКА СТАРТОВОГО БИТА
002A D300 51 OUT [PORTA],A; ВЫВОД БИТА
002C 110018 52 LD DE,STPTIM
002F CD0036 53 CALL DELAY ; ОЖИДАНИЕ

0032 D1 54 ;
0033 C1 55 POP DE
0034 FB 56 POP BC
0035 C9 57 EI
0036 58 RET
0037 59 ; ИМЯ DELAY
0038 60 ; ВХОДЫ: ЗНАЧЕНИЕ ЗАДЕРЖКИ В DE
0039 61 ; ВЫХОДЫ: НЕТ
003A 62 ; ВЫЗОВЫ: НЕТ
003B 63 ; ИЗМЕНЕНИЯ: A,D,E,ФЛАГИ
003C 64 ; ОПИСАНИЕ: ДЕКРЕМЕНТ DE И ВОЗВРАТ В ВЫЗВАВШУЮ
003D 65 ; ОПИСАНИЕ: ПРОГРАММУ ПРИ DE == 0
003E 66 ;
003F 67 DELAY
0040 68 DEC DE ; ДЕКРЕМЕНТ СЧЕТЧИКА
0041 69 LD A,D ; СРАВНЕНИЕ D С E В A
0042 70 OR E ; НУЛЕВОЙ РЕЗУЛЬТАТ, ЕСЛИ D == E == 0
0043 71 JP NZ,DELAY ; ВОЗВРАТ, ЕСЛИ НЕ НУЛЬ
0044 72 RET

```

Программа 3.3. Программа последовательного вывода с 16-битовой задержкой на языке ассемблера 8085.

сигналом на входном контакте и выявляет изменения его логического значения по мере поступления данных.

На данном этапе изложения материала книги целесообразно рассмотреть схемы алгоритмов преобразования формы представления данных, потому что они могут быть применены как

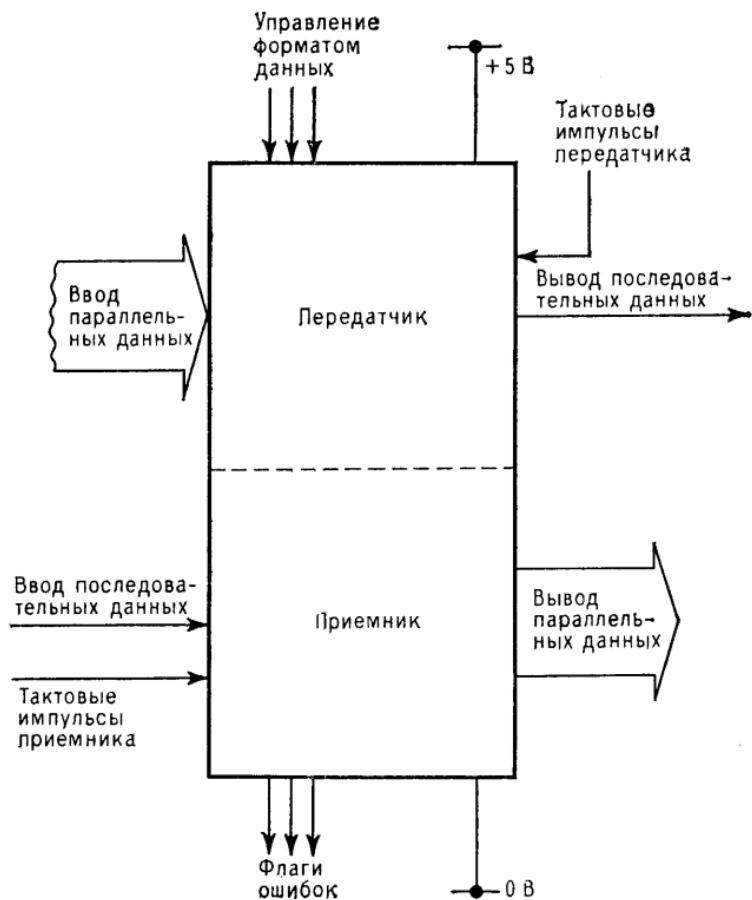


Рис. 3.14. Блок-схема простого универсального асинхронного приемопередатчика.

для построения аппаратных средств, так и для написания программ, выполняющих те же функции. Схемы алгоритмов приведены на рис. 3.11 и 3.12, а программы 3.1—3.3 являются примерами практической реализации этих алгоритмов.

Применение программных методов для выполнения параллельно-последовательного и последовательно-параллельного преобразования данных снижает эффективность использования времени работы процессора и обычно ограничено небольшими системами. В этих случаях задержки, необходимые для того,

чтобы темп преобразования соответствовал определенной скорости передачи, зависят от тактовой частоты процессора. Поэтому необходимо следить, чтобы эта частота не изменилась вследствие каких-либо неисправностей или замены кварцевого кристалла генератора на другой, имеющий иную рабочую частоту.

Для систем, содержащих несколько каналов последовательной передачи данных, более характерно применение упомянутых выше устройств на интегральных схемах, а не программных средств преобразования. Типовая блок-схема такого устройства представлена на рис. 3.13.

Стандарты последовательного обмена

Для определения режимов последовательной передачи данных, т. е. уровней сигналов, имеющих место в линиях, и быстродействия схем, используемых для генерации этих сигналов, существуют и продолжают появляться различные стандарты (см. стандартные скорости передачи и области их применения в табл. 3.3).

(а) Интерфейсы с односторонним источником напряжения RS232, V24 и RS423

Эти наиболее широко применяемые интерфейсные стандарты ориентированы на использование источника сигнального напряжения, который расположен со стороны передатчика. Логические сигналы представляются в этих интерфейсах следующим образом:

- 1 — отрицательное напряжение;
- 0 — положительное напряжение.

Стандарт RS232 определяет не только уровни сигналов, но и применяемый разъем, а также номера контактов, которые должны служить для передачи различных сигналов квитирования при использовании интерфейса с модемом (модулятором-демодулятором) при передаче данных по телефонным каналам.

Стандарт V24, разработанный Международным консультативным комитетом по телеграфии и телефонии (МККТТ), — это аналог американского стандарта RS232, применяемый в оборудовании европейского производства.

Хотя стандарт RS232 определяет функциональное назначение каждого контакта разъема, некоторые изготовители устройств, применяя этот стандарт, используют контакты разъемов не в соответствии с его нормами, что ставит проблемы совместимости при введении устройств в состав систем.

Ниже приводятся уровни сигналов и функциональное назначение контактов разъемов в соответствии со стандартом RS232

(табл. 3.4).

Логическая 1 определяется как напряжение, лежащее в диапазоне от -3 до -25 В.

Таблица 3.4. Назначение контактов разъемов в соответствии со стандартом RS232

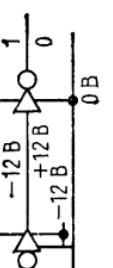
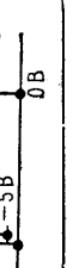
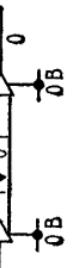
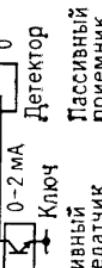
Номера контактов	Описание функций
1	Экран
2	Данные передатчика
3	Данные приемника
4	Запрос на посылку
5	Разрешение посылки
6	Готовность набора данных
7	«Земля» сигналов
8	Указатель сигнала на линии данных приемника
9 }	Контрольные точки
10	
11	Вторичная линия приемника
12	Указатель сигнала
13	Вторичная линия разрешения посылки
14	Сигнал передачи
15	Тактирование (от аппаратуры передачи данных)
16	Вторичная линия данных приемника
17	Элемент сигнала приемника
18	—
19	—
20	Готовность терминала данных
21	Указание качества сигнала
22	Кольцевой указатель
23	Выбор скорости передачи сигналов данных
24	Элемент сигнала передатчика
25	—

Логический 0 определяется как напряжение, лежащее в диапазоне от $+3$ до $+25$ В.

Стандарт предполагает применение 25-контактного разъема, причем назначение каждого контакта также оговорено стандартом. Стандартом предусмотрена возможность небольшого варьирования использования контактов в зависимости от применения интерфейса; конкретное назначение каждого контакта должно быть указано в справочных материалах изготовителя.

Стандарт RS232 не предназначен для связи устройств, удаленных друг от друга на большие расстояния. В качестве максимального обычно указывается расстояние около 20 м. Этот стандарт предусматривает применение модемов (модуляторов — демодуляторов), которые позволяют передавать данные на большие расстояния по телефонным каналам общего пользования или выделенным каналам; скорость передачи обычно составляет 300 или 1200 бод. Стандарты систем передачи данных приведены в табл. 3.5.

Таблица 3.5. Сравнительные характеристики стандартов последовательных интерфейсов

Тип интерфейса	Описание	Конфигурация схемы	Скорость передачи, Кбит/с	Максимальная длина линий, м
RS232C(U24)	Односторонний, драйвер — один приемник		20 (макс.)	16 Может быть увеличена без уменьшения скорости передачи. Интерфейс не предназначен для больших расстояний
RS423	Односторонний, драйвер — до 10 приемников		100 (макс.) 10 1	10 100 1300
RS422	Дифференциальный, один драйвер — до 10 приемников		10 000 (макс.) 1000 100	13 130 1300
Петля 20 мА	Токовая петля может охватывать несколько приемников, но обычно один драйвер — один приемник		10 1	300 2000 При нескольких приемниках длина линии должна уменьшаться с добавлением каждого нового приемника

Изготовители ИС выпускают в интегральном исполнении схемы преобразования уровней сигналов для сопряжения логических схем и линий передачи в соответствии со стандартом RS232. Для выходных каскадов передатчиков, отвечающих этому стандарту, требуется напряжение питания ± 12 В, но их входные сигналы могут иметь стандартные логические уровни. Входными для приемников являются сигналы линий, соответствующие стандарту RS232, а выходные сигналы приемников имеют типовые логические уровни. Таким образом, применение

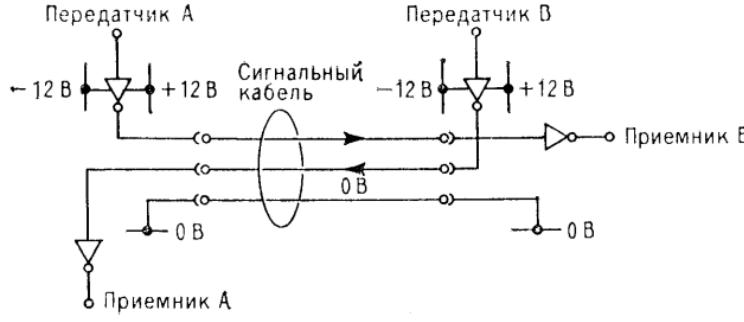


Рис. 3.15. Интерфейс для двухпроводной схемы по стандарту RS232C.

логических узлов, служащих для последовательно-параллельного и обратного преобразования, в составе интерфейсов стандарта RS232 не вызывает затруднений. Включение преобразователей уровней в двухпроводном интерфейсе, соответствующем этому стандарту, показано на рис. 3.15.

Стандарт RS423 предусматривает использование в линиях передачи более низких уровней напряжения (типовые величины ± 5 В) и допускает более высокие скорости передачи (100 Кбит/с при удалении 10 м) и большую удаленность пунктов передачи (1300 м при 1 Кбит/с). Еще одно преимущество этого стандарта перед RS232 заключается в возможности обслуживания нескольких (до 10) приемников посредством одной линии передачи.

Так же как и в случае стандарта RS232, имеются преобразователи для сопряжения уровней сигналов линий и ТТЛ-схем. Первым видом тестирования подобной системы связи является проверка уровня напряжения на двух линиях данных; это напряжение должно быть отрицательным, если линии пребывают в неактивном состоянии. Появление любого другого напряжения является признаком неисправности, которая должна быть устранена до проверки функционирования схем последовательно-параллельного и параллельно-последовательного преобразования.

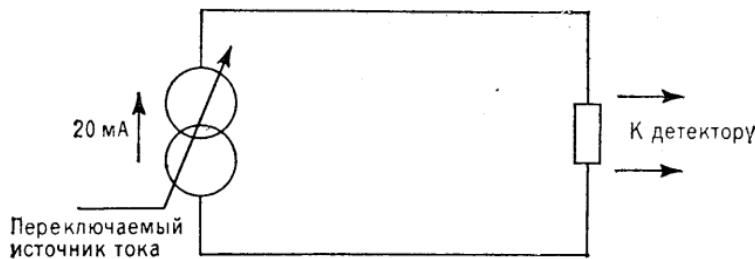


Рис. 3.16. Схематическое представление интерфейса с токовой петлей.

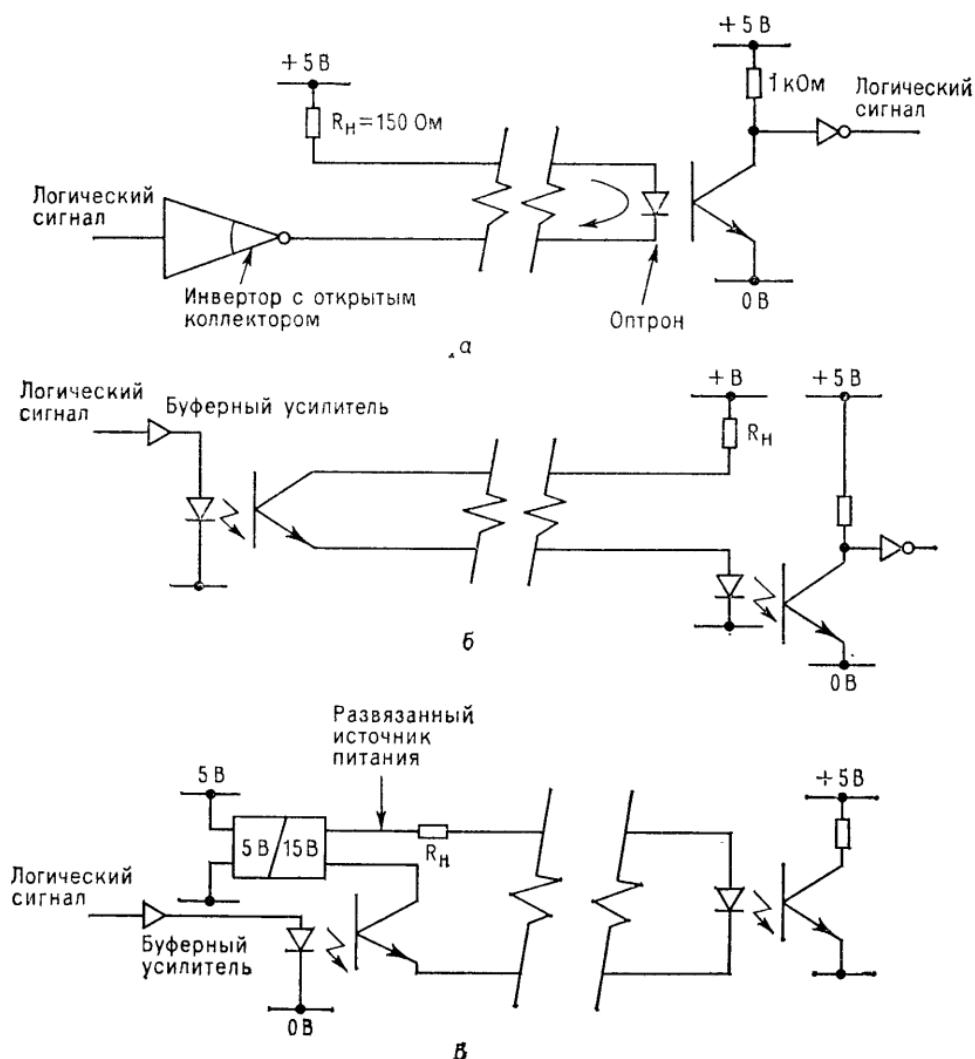


Рис. 3.17. Некоторые практические схемы интерфейса с токовой петлей:
а — источник тока в передатчике; б — источник тока в приемнике; в — изолированная линия передачи.

(б) Интерфейсы с токовой петлей

При этом способе передачи линия связи представляет собой пару проводов, которые образуют цепь, содержащую переключаемый источник тока и приемник; схема этой цепи представлена на рис. 3.16.

Токи, превышающие 17 мА, служат в такой схеме для представления логической 1, а токи, меньшие 2 мА,— для представления логического 0. Интерфейсы такого типа стандартизованы в меньшей степени, чем интерфейсы, соответствующие RS232, но они широко применяются в промышленных системах, так как позволяют осуществлять связь на больших расстояниях без необходимости использования модемов.

Системы передачи с токовой петлей могут быть построены так, что линия передачи в них оказывается развязанной как с источниками питания, так и за счет применения оптронов с передатчиком и приемником. Передатчики и приемники в свою очередь могут быть активными или пассивными. Эти обстоятельства создают предпосылки для самого различного построения систем передачи данных, некоторые примеры конфигурации которых представлены на рис. 3.17. Некоторые недавно разработанные устройства передачи и приема (такие, как HCPL-4100 и HCPL-4200) позволяют строить еще более простые интерфейсы с токовой петлей.

(в) Дифференциальные системы передачи типа RS422 и RS485

Это третья разновидность организации последовательного интерфейса. В ней применяются дифференциальные формирователи и приемники сигналов линий. Использование дифференциального принципа передачи повышает помехоустойчивость системы и, как следствие, позволяет существенно увеличить длину линий связи и скорость передачи по сравнению с интерфейсами обоих типов, описанных выше (1300 м при 100 Кбит/с, 10 Мбит/с при 13 м). Схема дифференциального интерфейса представлена на рис. 3.18.

Стандарт RS422 допускает, кроме того, подключение к одной линии более чем одного приемника, а более поздний стандарт PS485 ориентирован на совместную работу нескольких источников и нескольких приемников, что недопустимо при использовании стандарта RS232, предусматривающего лишь связь одного источника с одним приемником. Если интерфейс с токовой петлей используется для работы с несколькими приемниками, то при подключении каждого дополнительного приемника даль-

Представление печатаемых символов

Во всех вычислительных системах для представления числовых значений служат двоичные числа, хранящиеся в запоминающих устройствах. Необходимо, однако, располагать каким-либо способом двоичного кодирования символов, которые могут быть воспроизведены печатающими устройствами.

Американский стандартный код для обмена информацией (ASCII) был разработан и принят в качестве стандарта представления символов во всех областях, где имеет место печать

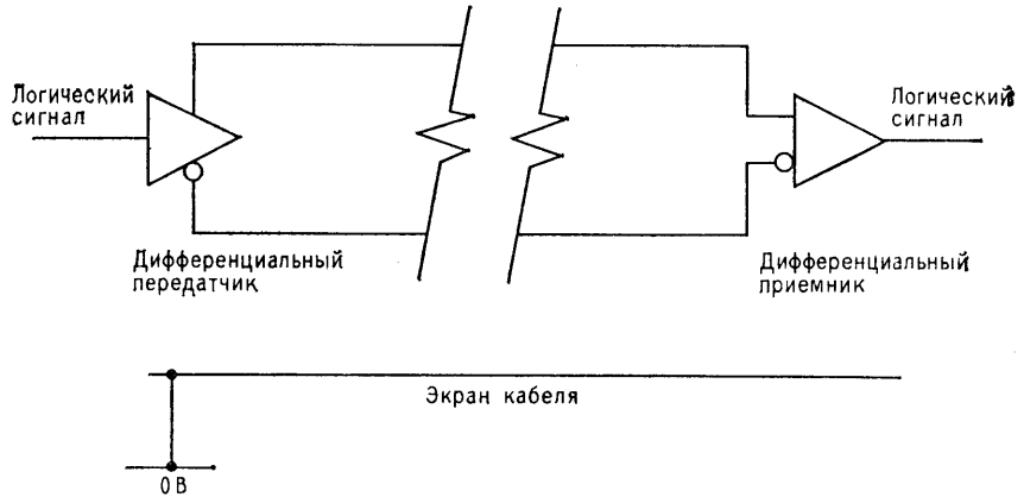


Рис. 3.18. Последовательный интерфейс дифференциального типа.

символьной информации. ASCII охватывает 128 различных символьных и управляющих кодов. Во всех устройствах, служащих для наглядного представления информации, предусмотрена возможность интерпретации этого кода. Для представления 128 двоичных кодовых комбинаций достаточно семи битов; поэтому в 8-разрядных машинах один бит остается свободным. Обычно его используют для обнаружения ошибок. Семь битов ASCII-кода размещаются в семи младших разрядах 8-битового слова, а старший разряд выполняет функции бита четности. Значение бита четности может определяться одним из двух способов:

1) с таким расчетом, чтобы количество единиц в слове было четным;

2) так, чтобы количество единиц в слове было нечетным.

Контроль, реализуемый при использовании первого из этих способов, принято называть проверкой на четность, а второго — проверкой на нечетность. Таким образом, из микроЭВМ в принтер передаются либо параллельно по восьми линиям, либо в последовательной форме семь битов данных и бит четности. В слу-

чае последовательной передачи биты данных всегда пересылаются, начиная с младшего, который передается сразу после стартового бита, а бит четности поступает перед стоповым интервалом. При искажении любого одного бита данных изменяется общее число единиц в 8-битовом слове; значит, любой принятый символ может быть проконтролирован по четности. В процессе передачи могут исказиться значения более чем одного бита, и контроль по четности может не выявить при этом наличия ошибок; для обнаружения множественных ошибок используются более сложные методы контроля.

Иногда возникают сомнения, какой из внешних контактов устройства (2 или 3) используется для передачи и правильно ли выполнена коммутация в обеих частях разъема, используемого для связи двух устройств. В этих случаях необходимо выяснить, на каком из внешних контактов в обоих устройствах имеет место отрицательный уровень напряжения, и таким образом прояснить ситуацию. Если, например, отрицательный уровень существует на контактах 2 обоих устройств, то прямое попарное соединение этих контактов 2—2 и 3—3 приведет к закорачиванию выходов двух устройств (в данном случае подсоединенных к внешним контактам 2). В этом случае корректное соединение устройств может быть достигнуто за счет соответствующей коммутации частей разъема.

Заключение

Передача данных внутри микропроцессорных систем, а также между системами и периферийными устройствами может осуществляться в параллельной или последовательной форме. Выше были рассмотрены три типа стандартов передачи: Centronics, IEEE 488 и RS232/V24. Знание характеристик систем передачи, построенных в соответствии с этими стандартами, позволяет конкретизировать решение проблем обслуживания.

Часть II

Методы обслуживания

Глава 4

Методы поиска неисправностей

На фоне развития автоматизированных средств поиска неисправностей в аппаратуре, построенной на основе микропроцессоров, для их отладки широко используются методы, применяемые при работе с обычными электронными устройствами. Специалист по обслуживанию микропроцессорной системы по-прежнему должен знать ее устройство и принципы функционирования. Существенную помощь при этом могут оказать электрические схемы и другая документация, включающая, например, таблицы размещения информации в памяти и даже листинги программ.

Многие проблемы, встречающиеся при обслуживании микропроцессорных систем и микроЭВМ, характерны для большинства электрических устройств. К их числу относятся

- неисправности источников питания;
- некачественность соединений;
- загрязненность контактов переключателей;
- неисправности конденсаторов, резисторов и т. д.

Однако кроме этих проблем, имеющих общий характер, обслуживание микропроцессорной аппаратуры затрудняется целым рядом ее специфических особенностей, примерами которых являются следующие:

— Сигналы внутри микроЭВМ передаются по шинам, обычно содержащим до 8, 16 или 32 линий связи.

— Информация на этих шинах быстро сменяется и зависит в каждый момент времени от конкретной функции, реализуемой в микроЭВМ; таким образом, сигналы на шинах не обязательно являются повторяющимися.

— Многие устройства, входящие в систему, параллельно подсоединены к системным шинам. При обнаружении некорректной информации на какой-то линии шины или в соединительном узле необходимо еще выяснить, какое из устройств является причиной некорректности.

— На шинах не всегда имеет место осмыщенная информация. Время от времени происходит смена передаваемой по шине

информации, или она оказывается в неактивном состоянии, когда выходы всех подключенных к ней устройств находятся в состоянии высокого импеданса.

Перечень подобных проблем достаточно большой, и решение их с использованием лишь традиционного контрольного оборудования не представляется возможным. В этой главе будут кратко рассмотрены некоторые новые методы поиска неисправностей, разработанные с целью преодоления этих проблем: логический анализ, сигнатурный анализ и внутрисхемная эмуляция. Ни один из этих методов, будучи применен в отдельности, не решает всех проблем, но каждый из них позволяет сформировать сведения об источнике неисправности.

Следует, кроме того, отметить, что во многих случаях причины некорректного функционирования лежат за пределами микропроцессорной части системы, например в источниках питания или в интерфейсных устройствах. Не стоит пренебрегать здравым смыслом, который заставляет считать, что знание традиционных цифровых и аналоговых устройств может оказать существенную помощь при решении проблем, связанных с обслуживанием устройств, не являющихся микропроцессорными. Вместо того чтобы пытаться рассмотреть здесь всеобъемлющий перечень возможных их неисправностей, пользование которым может само по себе оказаться затруднительным, ниже будет представлен некоторый набор практических рекомендаций, которые отражают опыт многолетней работы техников и инженеров — специалистов по обслуживанию электронной аппаратуры. Применение этих рекомендаций связано с относительно небольшими затратами времени и средств, и они позволяют снять многие вопросы, источником возникновения которых могла считаться микропроцессорная аппаратура.

Для каждого нового типа аппаратуры опытный инженер — специалист по обслуживанию — может выработать соответствующий набор приемов и методов, опирающийся на общие рекомендации и учитывающий конкретную специфику аппаратуры данного типа.

Простейшие виды контроля системы

Сначала рассмотрим некоторые тесты, выполнение которых по отношению к микропроцессорным системам требует весьма несложного оборудования. Простота этих тестов определяет целесообразность их рассмотрения в данной вводной главе.

Системный генератор тактовых импульсов

Некачественное тактирование может быть причиной многих неприятностей, от полной неработоспособности системы, когда она совершенно не способна функционировать, до работы

с явными нарушениями. Многие микропроцессоры, в том числе 6502, критичны к тактовой частоте и перестают работать, когда она выходит за допускаемые пределы. Работоспособность динамических запоминающих устройств, в частности сохранность информации в их ячейках, во многом определяется качеством регенерации. Если частота регенерации слишком снижается, с динамической памятью возникают проблемы. Частота тактовых сигналов и их прохождение в системе могут быть проконтролированы с помощью частотомеров и осциллографов.

Начальная установка

После того как сигнал на входе начальной установки меняется с логического 0 на логическую 1, микропроцессор, как правило, начинает выполнение программы с определенного адреса. Обычно специальная схема, предусмотренная для этого в составе микропроцессорной системы, задерживает поступление этого перепада на вход процессора до тех пор, пока на все устройства системы не будет подано питание и не будет выполнена необходимая начальная установка. Если сигнал начальной установки поступает на вход микропроцессора слишком рано или его форму искажают помехи, которые могут привести к многократному запуску системы, возможна некорректная начальная установка. Если на линии сигнала начальной установки оказывается низкий уровень, который не изменяется при подаче сигнала, то, очевидно, система вообще не сможет начать работать. Состояние входа начальной установки опять в определенной мере может быть выяснено с помощью обычного осциллографа, хотя одиночный сигнал начальной установки лучше наблюдать, пользуясь осциллографом с запоминанием.

Другие линии управляющих сигналов

Наличие шумов и неисправность схем, подключаемых к входам процессора, служащим для подачи таких сигналов, как запросы на прерывание, ожидание или на пользование шинами, также могут быть причинами неправильной работы или полного отказа системы. Для наблюдения сигналов на этих входах можно воспользоваться осциллографом или даже логическим пробником. Если какой-то из данных сигналов покажется подозрительным, возможно, найдется средство на время тестирования заблокировать его подачу.

Тестирование с входными воздействиями

Основной принцип, используемый при контроле любого устройства, является ли оно электрическим или механическим,

состоит в подаче входных воздействий и наблюдении реакции устройства с целью установления, правильно ли оно функционирует. Так водитель, подозревающий, что не работает стоп-сигнал, нажимает на тормозную педаль, а его товарищ смотрит в это время сзади на стоп-сигналы. Такой же подход вполне правомерен по отношению к микропроцессорной системе. Если оператору кажется, что в интерфейсе принтера имеют место неполадки, надо проверить функционирование интерфейса с помощью тестовой программы, которая «прогоняет его по всем режимам», посыпая в принтер все возможные символы ASCII, в то время как работа принтера контролируется либо визуально, либо с использованием тестовой аппаратуры.

```

2020 LDA 0000 ; In first ROM
LDA 0800 ; In second ROM
LDA 2000 ; In first RAM
LDA 2800 ; In second RAM
JMP 2020 ; back to start

```

Программа 4.1. Тестовая программа для дешифратора адреса для ЭВМ SDK 85

Программа 4.1 представляет собой пример тестовой программы задания входных воздействий, предназначеннной для перебора всех возможных сигналов выбора кристалла, являющихся выходными сигналами проверяемой ИС дешифратора адреса. Программа написана для одноплатной ЭВМ SDK-85 (см. приложение 2), но может быть применена без существенных изменений для почти любой микропроцессорной системы. Программа поочередно приводит в активное состояние выходные сигналы дешифратора, что дает оператору возможность проверить с помощью осциллографа правильность их выработки.

Свободный прогон микропроцессора

Простой, но эффективный метод, с помощью которого можно привести в действие значительную часть микропроцессорной системы, состоит в так называемом свободном прогоне микропроцессора. Рис. 4.1 дает представление о том, как реализуется этот метод. Процессор извлекается из смонтированного на печатной плате гнезда, и между ним и гнездом устанавливается колодка, обеспечивающая свободный прогон. Для этого на входы данных микропроцессора подается от колодки код NOP — Нет операции (00 для микропроцессора Z80, EA для 6502). Все другие связи

процессора с соответствующими проводниками печатной платы при наличии колодки сохраняются. После начальной установки микропроцессор обращается к памяти за первой командой и

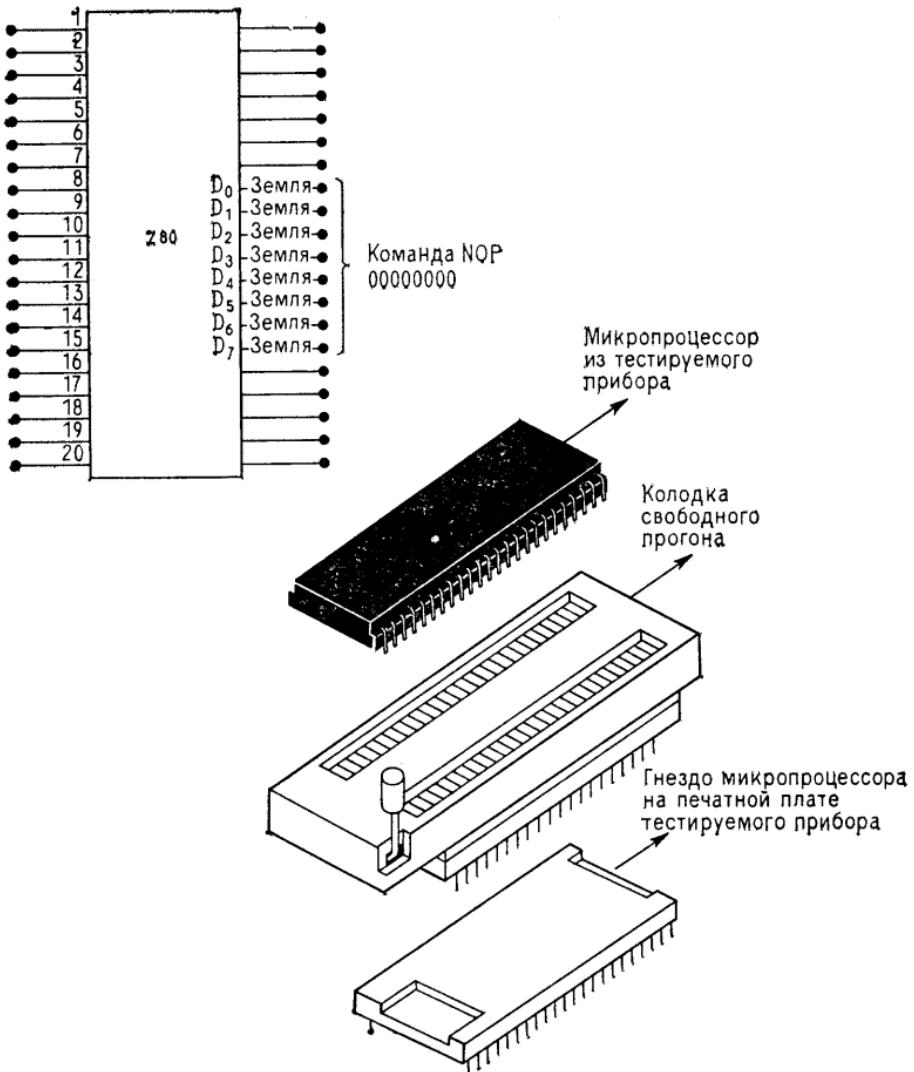


Рис. 4.1. Установка колодки свободного прогона между микропроцессором и платой тестируемого прибора.

в качестве таковой принимает NOP. Выполнив инкремент адреса, он обращается к следующей области памяти, пытаясь выбрать более содержательную команду. Однако поиск таких команд оказывается тщетным, так как при обращении к любой ячейке в него поступает все тот же код NOP. В результате процессор вынужден перебрать все адреса, входящие в его диапазон адресации.

При свободном прогоне адресная шина реализует функцию двоичного счета, причем на каждой из ее линий имеет место сигнал определенной частоты (рис. 4.2). Кроме того, в связи с тем, что процессор вырабатывает все возможные адреса от 0000 до FFFF, оказываются перебранными все возможные состояния логической схемы декодирования адреса, имеющиеся в составе системы.

Мультиплексирование шины, применяемое в 8085 и в некоторых новых 16-разрядных процессорах, таких, например, как

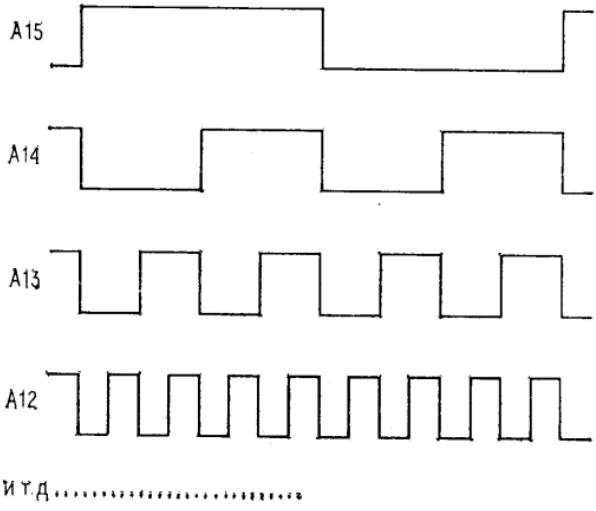


Рис. 4.2. Свободный прогон микропроцессора: адресная шина реализует функцию двоичного счета.

8086, обуславливает некоторое усложнение колодки свободного прогона (рис. 4.3). Двунаправленный буфер обеспечивает поставку задаваемого монтажом кода NOP на контакты адреса/данных только в те моменты, когда они используются для передачи данных.

Колодка свободного прогона позволяет просто, но весьма эффективно выполнить контроль значительной части микропроцессорной системы. Каждая адресная линия проверяется на отсутствие обрывов и коротких замыканий с соседними линиями путем контроля частоты сигналов на ней с помощью осциллографа. Точно так же могут быть проконтролированы выходные сигналы логической схемы выбора кристаллов памяти, а также проверено отсутствие обрывов в линиях, соединяющих дешифратоны с другими компонентами системы.

Источники питания

Проверка источников питания не должна ограничиваться измерением напряжений с помощью вольтметра. Такие измерения

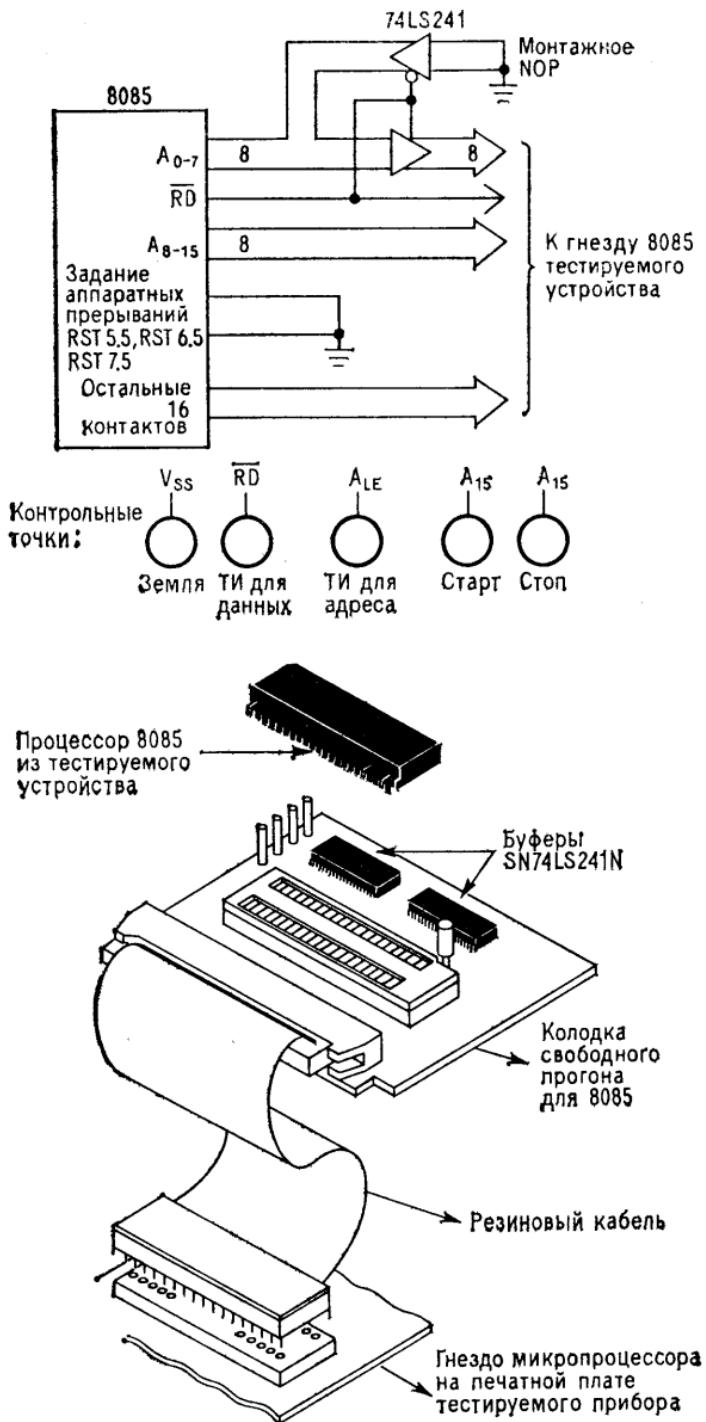


Рис. 4.3. Свободный прогон микропроцессора 8085 (аналогичные схемы могут быть применены для других процессоров с мультиплексируемой шиной, в том числе для 16-разрядного 8086).

могут показать отсутствие напряжения, но не дают представления о пульсациях и выбросах. Даже контроль питающих напряжений с помощью осциллографа должен выполняться очень осмотрительно. Любые пульсации питания обнаруживаются легко, а отклонения, превышающие несколько милливольт, являются для линии стабилизированного питания весьма существенными. Амплитуда колебаний уровня питания на фильтрующем конденсаторе источника может достигать 1—2 В. Хорошо спроектированный регулятор должен уменьшить ее примерно в тысячу раз, если не имеет места критический уровень, когда стабилизатор работает на пределе своих возможностей. В этом критическом режиме даже небольшое увеличение тока нагрузки (или падение напряжения на линии) приводит к непропорционально большому увеличению пульсаций на регулируемом выходе.

Если возникает предположение, что источник питания работает в критическом режиме, целесообразно прибегнуть к следующему простому тесту. Выясните, каково нормальное потребление тока (найдите его величину в руководстве по эксплуатации или измерьте), и подберите резистор, который увеличивал бы это потребление примерно на 10 %. Тщательно проверьте свои расчеты, чтобы избежать ошибок, и на несколько секунд подсоедините дополнительное сопротивление. Увеличение нагрузки в таких пределах является вполне допустимым и не должно существенно сказаться на изменении пульсации напряжения. Возрастание пульсации на величину, большую чем несколько милливольт, может означать, что стабилизатор работает на пределе своих возможностей.

Любые небольшие изменения потребления тока и напряжения питания, вызываемые и другими причинами, не приводя к повреждениям аппаратуры, могут стать причиной существенных искажений информации. Так, например, пиковые токи, возникающие при переходных процессах в выходных каскадах ТТЛ-схем или переключении внешних нагрузок, могут привести к соответствующим броскам напряжения в линиях стабилизированного питания, которые в свою очередь вызовут самые разнообразные изменения содержимого счетчиков и нарушения последовательности выполнения программы. Причины подобных случайных переходов, хотя и не являются абсолютно произвольными (для них всегда существуют какие-то внутренние предпосылки), очень трудно поддаются выявлению.

Аналогичным образом необходимо исследовать любые высокочастотные колебания напряжения питания, которые, хотя и кажутся незначительными в местах их обнаружения, могут быть следствием более существенных колебаний напряжения питания в других частях системы.

Влияние температуры

С проблемами обеспечения работоспособности аппаратуры тесно связан вопрос о влиянии повышенной температуры. Довольно типична для микропроцессорных систем возможность установки дополнительных плат и введения дополнительных устройств памяти по мере того, как у пользователя возникает необходимость расширения функций системы. Некоторым микро-ЭВМ, как, например, «классической» Apple, возможность наращивания технических средств внутренне присуща, но даже при самых оптимальных предпосылках для этого слишком заметное увеличение количества оборудования может привести к неприятным последствиям. В качестве первого из них укажем опасность приближения режимов работы имеющегося в системе источника питания к предельным режимам стабилизации, как это было только что рассмотрено. Кроме того, повышение температуры внутри корпуса может оказаться на надежности системы, что может, например, проявиться в виде прерывистости соединений между устройствами. Хорошо проверить в этом отношении предполагаемые опасные точки системы, и простое дотрагивание пальцем до корпусов ИС зачастую помогает обнаружить места явного перегрева, хотя и не следует забывать, что сложные устройства неизбежно потребляют большие токи и могут быть весьма горячими на ощупь. Подобная проверка, конечно, субъективна, но при наличии опыта, получаемого при работе со многими образцами определенного изделия, она позволяет выявить ненормальное повышение температуры.

Если возникает подозрение, что источником перегрева является какое-то конкретное устройство, можно попробовать оросить его бытовым «замораживающим» аэрозолем. Если оно действительно содержит неисправность, вызывающую перегрев, то понижение температуры под действием аэрозоля может устраниТЬ или изменить проявления этой неисправности. Этот метод применим также к непропаянным соединениям, подвергающимся расширению и сокращению под действием источников местного нагрева.

Когда источник перегрева не столь точно локализован, его уточнению могут помочь открывание дверей стоек и другие способы улучшения циркуляции воздушных потоков. В таких случаях для проявления неисправностей требуется больше времени, и происходит это, как правило, по прошествии части рабочего дня. Может понадобиться несколько часов после включения питания, чтобы с приближением полудня температура в лаборатории или цехе поднялась в степени, достаточной для возникновения нарушений в работе системы.

Помехи

Анализ влияния помех дает еще один повод для размышлений о перемежающихся сбоях и их возникновении в зависимости от времени суток. Попытайтесь найти закономерности в появлении этих неисправностей. То, что на первый взгляд кажется случайным, совершенно не связанным с окружающей обстановкой, может фактически происходить синхронно с событиями, имеющими место за пределами помещения, где находится проверяемая аппаратура. Здесь в первую очередь имеются в виду помехи, возникающие при переключении где-то на территории предприятия энергоемких нагрузок. Установить такие причины некорректной работы обычно непросто; однако при появлении частых сбоев может оказаться необходимым переместить неустойчиво работающее устройство в другое помещение или хотя бы подать на него питание с другого выхода источника. В качестве еще одного простого средства решения данной проблемы укажем возможность установки фильтра в цепи питания, положительный эффект воздействия которого не только подтвердит правильность локализации причины неустойчивой работы, но и устранит эту причину.

Ни одно из этих средств не может заменить грамотного построения источника питания, в котором предусмотрено эффективное подавление выбросов, возникающих при переходных процессах, и, конечно, всегда существует риск, что дополнительные меры подавления, которых едва хватает при проведении тестирования, окажутся недостаточными при работе аппаратуры в наполненной электрическими шумами «реальной» обстановке.

Неисправности интерфейса

В качестве первого шага по выявлению неисправностей этого типа необходимо сопоставить характеристики аппаратных и программных интерфейсных средств с их паспортными данными, чтобы выявить возможные отклонения. На печатных платах могут быть навесные монтажные перемычки, а руководство по программному обеспечению может содержать сведения о каких-то дополнительных стандартных программах. Когда речь идет об интерфейсах, то наибольшей неясностью отличается обычно документация, описывающая шины системы. Назначение некоторых контактов разъемов вполне однозначно, а решение об использовании других предоставляетяется принимать изготовителю. Допустимая длина соединительного кабеля, при которой обеспечивается достоверная передача, связана со скоростью передачи сигналов. Полезно убедиться, что скорости передачи как на стороне передачи, так и на приемной стороне установлены одинаково.

ковые, а длина кабеля такова, что не будет причиной неожиданных задержек. Для большинства систем скорости передачи от 25 до 1200 бод не должны вызывать проблем, но при более высоких скоростях передача может становиться ненадежной.

Интерфейс между микропроцессорной системой и ее окружением представляет собой питательную почву для всякого рода проблем, так как к проблемам, связанным с установленными на платах электронными схемами, добавляются неприятности, обусловленные применением разъемов, и трудности согласования соединяемых устройств. Сами периферийные устройства могут быть источниками ошибок, которыеказываются на работоспособности микропроцессорной системы в целом. Типичным примером является в этом отношении принтер с ограниченной буферной памятью. При небольших объемах текстов и низких скоростях передачи принтер ведет себя безупречно. Если какой-либо из этих показателей существенно возрастает, принтер не успевает печатать текст, заносимый в буфер, и происходит потеря или неправильная печать символов. Короткая тестовая последовательность позволяет убедиться в отсутствии ошибок печати, а впоследствии полный прогон выявит дефекты печати, которые могут быть отнесены за счет неисправностей в других частях системы по принципу «этот принтер до этого работал прекрасно, а значит, виновата электроника».

Периферийные устройства

Характерными неисправностями дисковой памяти являются неправильная ориентация и дефекты дисков. Пользователь может просто неаккуратно вставить диск. Дефекты дисков должны проявляться в виде отчетливых и недвусмысленных сообщений об ошибках. Необходимо всегда иметь резервные дубликаты дисков, и не стоит использовать основной экземпляр, пока не сняты сомнения об исправности дисковода. Чистка головок обычно не составляет труда, но юстировка дисковода и ремонт его электронных схем требуют гораздо большей квалификации.

С мониторами и дисплеями связаны проблемы другого плана. Хотя это довольно сложные устройства, технология их изготовления хорошо проработана, а принципы построения и функционирования вполне доступны специалистам по обслуживанию радио- и телевизионной аппаратуры, имеющим некоторый опыт практической работы. Несколько новыми для них могут оказаться дешифрирующие устройства клавиатур, но неисправности здесь обычно имеют явно выраженный механический характер. Экраны и схемы формирования изображений, имея относительно более высокую разрешающую способность, в основном аналогичны соответствующим узлам телевизионных приемников.

Виды ошибок

Причины некорректной работы микропроцессорных систем можно разделить на несколько категорий, например, следующим образом: выход из строя или изменение характеристик компонентов, помехи от внешних источников, некачественность разработки, ошибки пользователя. Причины двух первых типов были рассмотрены выше. Некачественность разработки должна проявляться лишь на стадиях исследования и проектирования изделий.

Исключениями в этом аспекте являются опытные и малосерийные образцы, контроль которых может не охватить всего набора проверок и перепроверок. Характерными примерами такой ситуации являются перспективные разработки, выполняемые в учебных и исследовательских лабораториях в условиях сжатых сроков научными сотрудниками или инженерами, которые могут не иметь опыта работы с микропроцессорами. При этом может случиться так, что опытный образец, нормально функционирующий в благоприятных лабораторных условиях, окажется неработоспособным в реальной обстановке. В качестве некоторых наиболее вероятных причин некорректной работы назовем нечеткость тактирования, пологость фронтов и дребезги сигналов начальной установки, плохую буферизацию и согласование в схемах интерфейса. Эти дефекты могут вкладываться и в системы, разрабатываемые опытными проектировщиками, но в этих случаях они, как правило, выявляются и устраняются задолго до начала серийного производства.

«Ошибки пользователей» — это сокращенное название множества причин некорректной работы, ответственность за которые нельзя возложить на разработчиков и изготовителей аппаратуры или отнести за счет качества компонентов, хотя часто неграмотные действия пользователя являются следствием несовершенства предоставляемой в его распоряжение документации. К числу наиболее распространенных ошибок пользователей относятся следующие:

- неправильная организация интерфейса, когда пользователь не учитывает нагрузочных возможностей, применяет несовместимые периферийные устройства или неправильно использует контакты разъемов;

- неквалифицированное составление программ, без учета особенностей операционной системы, применение «приемов», повышающих эффективность программы, но увеличивающих вероятность возникновения ошибок впоследствии, когда потребуется внесение изменений в программу;

- неправильное обращение с аппаратными средствами, когда пользователь вносит изменения в схему с целью расширения

ее функциональных возможностей, что приводит к дополнительным нагрузкам и временными задержками.

Учет неисправностей

Сказанное выше наводит на мысль о необходимости систематизации сведений и выявления причинно-следственных связей в комплексе мер по обеспечению работоспособности микропроцессорных систем. Если описание связей может вызвать определенные трудности, то сбор данных о неисправностях может оказаться наилучшим методом формирования подобного комплекса. Если неисправность какого-то типа возникает либо в одно и то же время суток, либо при размещении аппаратуры в одном и том же месте, либо после выполнения определенной последовательности действий, то она уже не является случайной. Мы можем еще не знать ее причин, но паника, вызываемая появлением случайных, несистематически проявляющихся неисправностей, уже не является оправданной. Все случаи некорректного функционирования надо тщательно регистрировать, причем форма регистрации должна по возможности учитывать условия работы системы. Это может оказаться полезным в следующих отношениях:

- разработчики и поставщики оборудования могут получить сведения по «обратной связи» о слабых местах системы и принять меры к их устраниению;
- на ранних этапах эксплуатации устройства получаются данные, которые могут быть приняты во внимание при работе с другими устройствами того же типа;
- выявляются внешние источники ошибок, которые могут воздействовать и на устройства других типов;
- формируются банки данных о неисправностях и закономерности, описывающие их возникновение, которые представляют огромную ценность для специалистов по обслуживанию при последующей эксплуатации;
- в процессе фиксации определяются слабые места конкретных типов оборудования, выпускаемого определенными изготовителями; часто случается, что во многих микроЭВМ наблюдаются одни и те же проявления некорректного функционирования, являющиеся следствием работы устройств на предельных режимах в критических ситуациях.

Перечни проверок и алгоритмы их выполнения

Опытный человек, приступая к решению задач обслуживания, зачастую пробирается к их сути интуитивно. Это нехорошо, так как, действуя подобным образом, он (или в наш просвещенный век она) подсознательно отбрасывает некоторые на-

правления поиска, делает ряд поверхностных наблюдений и целиком полагается на имеющийся опыт. Для начинающих и менее уверенных в себе следует рекомендовать более систематизированную методику. Надо заранее составить список необходимых проверок и в дальнейшем придерживаться его. Такие списки должны составлять опытные специалисты по обслуживанию, имеющиеся в данной организации, обращая особое внимание на описание тех действий, которые они сами обычно выполняют автоматически. Подобный перечень будет содержать много три-виальных пунктов:

- проверка, подключена ли система к источнику питания;
- проверка, включено ли питание;
- снятие показаний всех дисплеев;
- регистрация состояния всех индикаторов;
- проверка стабилизируемых напряжений.

На каждом шаге работы по перечню может оказаться достаточно получить ответы на соответствующие вопросы по принципу «работает/не работает». Если же перечень должен обеспечить более детальный анализ ошибок, то ответов «да/нет» и действий, базирующихся на этих ответах, становится недостаточно. Для этих целей может потребоваться разработка соответствующего алгоритма, предусматривающего переходы к вспомогательным тестам, которые реализуются в зависимости от того, просто выявлены или устраниены причины конкретных ошибок. Может оказаться возможным построение универсального алгоритма или хотя бы такого алгоритма, который был бы применим к широкому набору устройств, но не исключено, что его сложность заставит отдать предпочтение разработке более простых частных алгоритмов для отдельных устройств.

Опытный пользователь будет по-прежнему стремиться при работе опустить некоторые фрагменты полной схемы проверки и положиться на свой опыт и интуицию, но если эта схема составлена квалифицированно, ее использование позволяет сберечь время. Начальные проверки могут быть простыми и быстрыми, но их применение зачастую позволяет сразу же исключить из рассмотрения значительные части системы. Если, например, источник питания неисправен, но не был проверен, то многие неисправности различных устройств системы могут быть проявлениями плохой работы источника. Одна-две минуты работы с вольтметром и проверки критических точек с помощью осциллографа позволяют исключить подозрения в некачественности напряжения питания.

Заключение

Микропроцессорные системы имеют шинную архитектуру, и некоторые их неисправности трудно обнаружить, не пользуясь

шинно-ориентированной контрольной аппаратурой. Принудительно заставляя систему проходить определенную последовательность состояний, можно выявить явные неисправности с помощью осциллографа и логического пробника. Причинами некорректного функционирования системы могут быть неисправности источников питания, повышение внутренней или окружающей температуры и внешние помехи. Они могут, кроме того, быть вызваны ошибками разработчиков и пользователей, а также отказами компонентов. Все случаи отказов должны фиксироваться с целью выявления закономерностей и причин их возникновения. Для поиска неисправностей при отсутствии достаточного опыта могут быть применены перечни проверок и алгоритмы их выполнения.

Глава 5

Принципы самоконтроля

Микропроцессорные системы и микроЭВМ могут содержать встроенные тестовые программы. Они служат для занесения и считывания в память и через устройства ввода-вывода стандартных наборов данных. Тестовые программы содержат примеры функционирования реальных работающих систем.

Во многих микропроцессорных системах для проведения тестирования используются их собственные процессоры. Тестовые программы, записанные в ПЗУ и инициируемые при включении питания или по запросу, обеспечивают контроль ПЗУ, ЗУПВ самого процессора и в определенной степени всех устройств ввода-вывода. Результаты контроля обычно доводятся до оператора через дисплей системы или с помощью специального индикатора, который сообщает об успешном завершении проверки. Существуют, однако, некоторые тестовые программы, предназначенные для работы с внешними приборами, такими, например, как сигнатурный анализатор.

Тестовые программы не обязательно должны находиться в ПЗУ. Они могут загружаться с диска или ленты и ЗУПВ и затем исполняться. Понятно, что при проведении самопроверки предполагается, что большая часть системы функционирует нормально или хотя бы в достаточной степени удовлетворительно для загрузки и прогона тестовых программ.

Тестирование ЗУПВ

Типовой подход к контролю ЗУПВ состоит в записи в него набора данных, последующего считывания и проверки, что обе операции были выполнены успешно. Для обнаружения возможных неисправностей ЗУПВ используется несколько подходов, которые рассматриваются ниже.

(A) *Тест «шахматная доска».* В ячейки памяти заносятся значения 1 и 0, размещаемые так же, как располагаются черные и белые клетки на шахматной доске, после чего проверяется

правильность записанной информации. Затем в каждую ячейку заносится противоположное значение и снова выполняется проверка. При этом выясняется, может ли каждый элемент памяти хранить значения как логической 1, так и логического 0 (рис. 5.1). Наборы данных, аналогичные по структуре шахматной доске, в отличие от наборов, состоящих целиком из единиц и нулей, делают тест чувствительным к возможным паразитным

заязкам между соседними ячейками.

(Б) Тест с заполнением нулями и единицами. При данном виде проверки сначала вся память заполняется нулями. Затем считывается содержимое первой ячейки и в нее записывается 1. Считывание нулей и замещение их единицами последовательно выполняется для всех ячеек, пока вся память не будет заполнена единицами. Затем эти действия повторяются в обратном порядке, т. е. все ячейки, начиная с последней, снова заполняются нулями. Теперь вся

Рис. 5.1. Проверка ЗУПВ с помощью теста «шахматная доска». 1 — битам присваиваются поочередно значения 1 и 0 и проверяется содержимое памяти. 2 — содержимое памяти инвертируется и проверяется каждый раз перед переходом к следующему ряду ячеек.

последовательность действий воспроизводится вновь с той разницей, что при инициализации вся память заполняется единицами (рис. 5.2).

(В) Тест с перебором дополнительных адресов. Этот вид тестирования служит для выявления некорректного функционирования внутренних дешифраторов адреса ИС ЗУПВ. Оно начинается с записи в память чередующихся столбцов нулей и единиц. Считывается содержимое первой ячейки, затем содержимое ячейки с адресом и потом снова содержимое исходной ячейки. Тест продолжается аналогичным образом по отношению к следующей ячейке и т. д. до тех пор, пока не будет считано содержимое всех ячеек. Затем инвертируется исходное содержимое всей памяти и вся процедура повторяется снова (рис. 5.3).

Последовательный перебор всех адресов памяти, которая размещена на ИС, осуществляемый с использованием инвертирования их разрядов, представляет собой серьезное испытание для внутренних дешифраторов и позволяет выявить все задержки, выходящие за допустимые пределы.

	00	01	10	11
00	0	1	0	1
01	1	0	1	0
10	x	x	x	x
11	x	x	x	x

	00	01	10	11
00	0	0	0	0
01	0	0	0	0
10	0	0	0	0
11	0	0	0	0

Сначала память
заполняется нулями

1	2	3	4
1	1	1	1
0	0	0	0
0	0	0	0

Считывается содержимое первой ячейки,
проверяется его равенство 0
и заменяется на 1. Процедура
повторяется, пока память
не заполнится единицами

1	1	1	1
1	1	1	1
1	1	0	5
0	0	0	0

Аналогичным образом
начиная с последней ячейки
память заполняется нулями

1	1	1	1
1	1	1	1
1	1	0	5
0	0	0	0

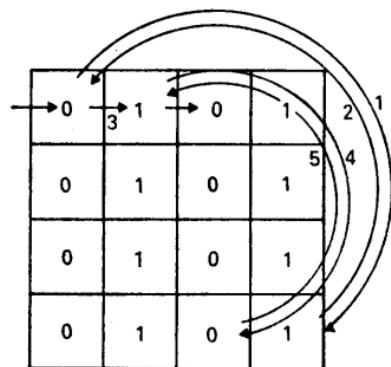
Все перечисленные действия
повторяются, начиная с заполнения
памяти единицами и последователь-
ной замены их на нули

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

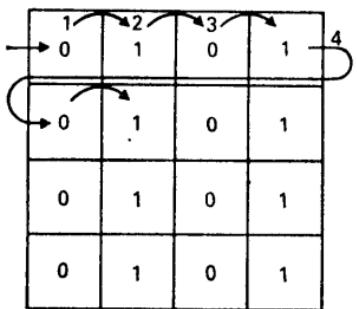
Рис. 5.2. Последовательность тестирования ЗУПВ с заполнением нулями и единицами.

	00	01	10	11
00	0	1	0	1
01	0	1	0	1
10	0	1	0	1
11	0	1	0	1

Сначала память заполняется
перемежающимися столбцами
единиц и нулей



Затем считывается и проверяется
содержимое первой ячейки,
а потом ячейки с обратным адресом.
Те же действия повторяются
для второй ячейки



Описанная процедура
повторяется
для всей памяти

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

Весь тест повторяется
для инвертированного
исходного содержимого
памяти

Рис. 5.3. Тестирование ЗУПВ с последовательным перебором дополнительных адресов.

Некоторые методы тестирования памяти связаны с большими затратами времени; на полную реализацию их может уходить несколько часов. Поэтому обычно для самоконтроля применяются лишь простые алгоритмы, такие, например, как «шахматная доска», а более детальные тесты, чувствительные по отношению к более сложным сочетаниям данных, используются для выявления определенных, уже подозреваемых неисправностей ЗУПВ. Ни один тест не гарантирует полной проверки исправности памяти. Если какой-то тест ЗУПВ выполняется успешно, то довольно высока вероятность того, что оно исправно; если же тест не выполняется, то наличие неисправности не вызывает сомнений.

Программа 5.1 представляет собой пример типичной программы контроля ЗУПВ методом «шахматной доски». Как и в большинстве программ, предназначенных для быстрой проверки функционирования памяти, здесь выполняются поочередно запись и считывание шестнадцатеричных чисел АА и 55 в восьмибитовые области памяти. При обнаружении ошибки управление может быть возвращено программе-монитору или предусмотренной пользователем программе обработки ошибки; содержимое регистровой пары Н указывает при этом адрес памяти, по которому была обнаружена ошибка.

Тестирование ПЗУ

Простейший метод тестирования ПЗУ включает формирование контрольной суммы. Содержимое всех областей ПЗУ суммируется, а полученная сумма обрезается до восьми разрядов контрольного байта. Программа, инициируемая в ходе самопроверки, вычисляет контрольную сумму и сравнивает ее с хранящимся достоверным значением. Несовпадение свидетельствует о неисправности ПЗУ. При наличии совпадения ПЗУ может быть исправно, но остается непроверенной возможность взаимной компенсации нескольких имеющихся в нем неисправностей.

Более точный метод, при использовании которого существует меньшая вероятность взаимного маскирования ошибок, предусматривает формирование контрольных циклических избыточных кодов (CRC—Cyclic Redundancy Check). Первоначально метод CRC возник как средство контроля передачи данных между ЭВМ и их периферийными устройствами, например такими, как накопители на магнитных дисках и лентах. Чтобы понять, каким образом он используется для тестирования ПЗУ, допустим, что все содержимое кристалла ПЗУ выводится из него слово за словом так, что образуется одна большая линейная цепочка битов. Эта цепочка вводится в сдвиговый регистр, снабженный цепями обратной связи, благодаря которым осуществляется

сложение по модулю 2 данных обратной связи и новых данных, поступающих в виде битового потока. Результат, остающийся в регистре после ввода всех битов содержимого ПЗУ, представляет собой циклический код контроля по избыточности (рис. 5.4)¹⁾. Как и большинство вычислительных задач, описанная процедура может быть реализована программным путем.

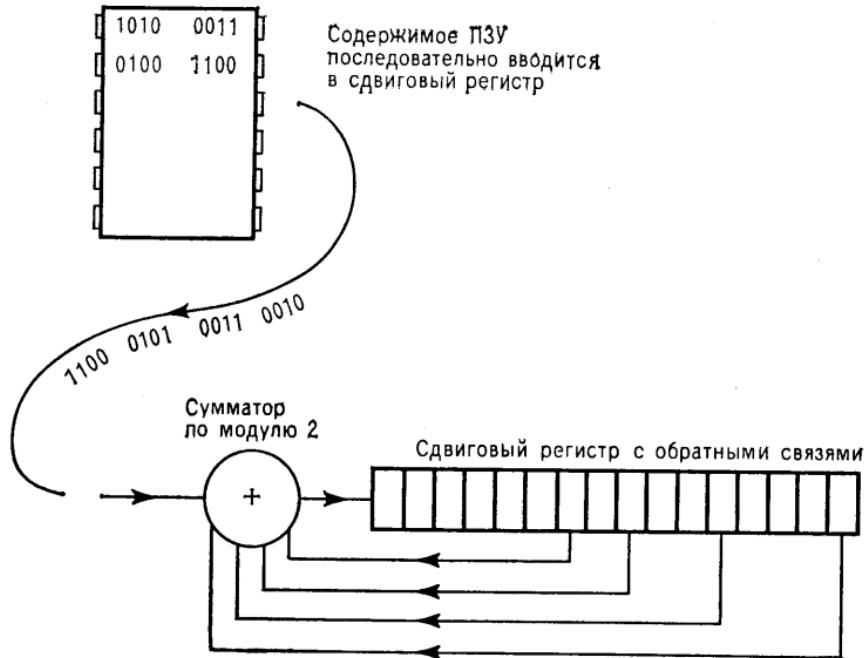


Рис. 5.4. Формирование сигнатуры ПЗУ.

Программа 11.3 представляет собой программу на языке ассемблера, служащую для вычисления циклического контрольного кода блока данных объемом 2 Кбайт.

Чувствительность данного метода контроля обусловливается конфигурацией цепей обратной связи, которые обеспечивают влияние каждого бита, вводимого в регистр, на окончательную сигнатуру. Если хотя бы один бит пропущен, высока вероятность того, что окончательный CRC будет существенно отличаться от исходного.

Устройства ввода-вывода

Разнообразие устройств ввода-вывода и форм их взаимодействия с окружающим миром определяет сложность решения проблем их самоконтроля. Обычно устройство ввода-вывода

¹⁾ Этот метод составляет основу сигнатурного анализа, который будет рассмотрен в гл. 9.

0028	ПРИ ВОЗНИКНОВЕНИИ ОШИБКИ УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ERROR,
0029	ГДЕ ТОЖЕ МОЖЕТ БЫТЬ РАЗМЕЩЕНА КОМАНДА ПЕРЕХОДА
0030	К МОНИТОРУ ИЛИ ПРЕРЫВАНИЮ
0031	A = РЕАЛЬНЫЕ ДАННЫЕ
0032	B = ОЖИДАЕМЫЕ ДАННЫЕ
0033	HL = АДРЕС ОШИБКИ
0034	ЕСЛИ ПРИ ОБНАРУЖЕНИИ ОШИБКИ ОСУЩЕСТВЛЯЕТСЯ ПЕРЕХОД
0035	К МОНИТОРУ, РЕКОМЕНДУЕТСЯ СОХРАНЬ В ПАМЯТИ
0036	СОДЕРЖИМОЕ РЕГИСТРОВ ВО ИЗБЕЖАНИЕ ИСКАЖЕНИЯ ЕГО
0037	МОНИТОРОМ.
0038	С ЭТОЙ ЦЕЛЬЮ АДРЕС ERROR ЯВЛЯЕТСЯ ПОСЛЕДНИМ
0039	В ПРОГРАММЕ, ЧТО ОБЕСПЕЧИВАЕТ ВОЗМОЖНОСТЬ ЕЕ РАСШИРЕНИЯ
0040	ДЛЯ ПРОГРАММЫ НЕОБХОДИМ СТЕК НЕ МЕНЕЕ 10 БАИТОВ.
0041	УКАЗАТЕЛЬ СТЕКА Д. Б. УСТАНОВЛЕН ПЕРЕД НАЧАЛОМ
0042	ВЫПОЛНЕНИЯ ПРОГРАММЫ.
0043	ОБРАТИТЕ ВНИМАНИЕ, ЧТО ПРИ ВЫХОДЕ ИЗ ERROR СТЕК НЕ ВОССТАНАВЛИВАЕТСЯ
0044	ТЕСТ ОБНАРУЖИВАЕТ ВСЕ НЕКОРРЕКТНЫЕ БИТЫ И НЕКОРРЕКТНЫЕ
0045	ОШИБКИ АДРЕСАЦИИ
0046	START DS 2 ; НАЧАЛЬНЫЙ АДРЕС
0047	END DS 2 ; КОНЕЧНЫЙ АДРЕС
0048	QUIKTST LD B,0AAH ; ЗАПИСЬ АА ВО ВСЕ ЯЧЕЙКИ ПАМЯТИ
0049	CALL TEST ; СЧИТЫВАНИЕ И ПРОВЕРКА ЭТОЙ
0050	CD1100' LD B,055H ; ЗАПИСЬ И ПРОВЕРКА ЧИСЛА 55
0051	TESTEND CALL TEST ; Т.О. КАЖДЫЙ БИТ ПРОВЕРЕН НА 1 И 0
0052	R JP \$; КОНЕЦ ТЕСТА
0053	CD1100' 0009'
0054	C30E00' 000B'
0055	R 000E'

; ПРОВЕРКА ЗАПОЛНЕНИЯ ПАМЯТИ И ЧТЕНИЯ ИЗ В
 0056
 0057 ;
 0058 TEST CALL GETEND ; ЗАПИСЬ КОНЕЧНОГО АДРЕСА В DE
 0059 CALL FILL ; ЗАПОЛНЕНИЕ ПАМЯТИ ИЗ В
 0060 LD HL,(START) ; ВЗЯТИЕ НАЧ.АДРЕСА ИЗ ЯЧЕЙКИ 100
 0061 LD A,(HL) ; ПОЛУЧЕНИЕ ОБРАТНО СЛЕДУЮЩЕГО
 ; БАЙТА
 0062 CP A,B ; ТО,ЧТО БЫЛО ЗАПИСАНО?
 0063 JR NZ,ERROR ; НЕТ,ЗНАЧИТ НЕИСПРАВНОСТЬ!
 0064 CALL CHLDE ; ПРОВЕРКА,НЕ ДОСТИГНУТ ЛИ КОНЕЦ
 0065 RET Z ; ДА,ЗНАЧИТ КОНЕЦ
 0066 INC HL ; ИНКРЕМЕНТ УКАЗАТЕЛЯ ОЧЕРЕДНОГО
 ; БАЙТА
 0067 JR TEST100 ; ЦИКЛ ДЛЯ ВСЕЙ ПРОВЕРЯЕМОЙ
 ; ПАМЯТИ
 0068 GETEND LD,HL(END)
 0069 PUSH HL
 POP DE
 0070 RET ; ПЕРЕСЫЛКА В DE И ВОЗВРАТ
 0071 LD HL,(START) ; ЗАПИСЬ НАЧ.АДРЕСА В HL
 0072 LD (HL),B ; ЗАПИСЬ СОДЕРЖИМОГО В
 0073 FILL100 ; ПРОВЕРЯЕМУЮ ПАМЯТЬ
 0074 CALL CHLDE ; НЕ КОНЕЦ?
 0075 RET Z
 0076 INC HL
 0077 JR FILL100
 0078 ;
 0079 ; СРАВНЕНИЕ HL И DE НА РАВЕНСТВО
 0080 ;
 0081 CHLDE LD A,H ; ВЗЯТИЕ Н
 CP A,D ; СРАВНЕНИЕ С D

0038'	CO	0083	RET NZ	; НЕ РАВНЫ?
0039'	7D	0084	LD A,L	; СРАВНЕНИЕ МЛАДШИХ БАЙТОВ
003A'	BB	0085	CP A,E	
003B'	C9	0086	RET	
		0087		
		0088		; ЗАПИСЬ И ПРОВЕРКА АДРЕСА КАЖДОЙ ЯЧЕЙКИ
		0089		
	CD2500'	0090	ADTEST	CALL GETEND ; ЗАПИСЬ КОНЕЧНОГО АДРЕСА В DE
	2A0000'	0091		LD HL,(START) ; ЗАПИСЬ НАЧАЛЬНОГО АДРЕСА В HL
	74	0092	ADT100	LD (HL),H ; ЗАПИСЬ СТАРШЕГО БАЙТА АДРЕСА
	CD360J'	0093		CALL CHLDE ; ПРОВЕРКА НА ОКОНЧАНИЕ
	2808	0094		JR Z,ADT200 ; ОКОНЧАНИЕ НАЙДЕНО
	23	0095		INC HL ; ИНКРЕМЕНТ УКАЗАТЕЛЯ
	75	0096		LD(HL),L ; И ЗАПИСЬ МЛАДШЕГО БАЙТА АДРЕСА
	CD3600'	0097		CALL CHLDE ; ПРОВЕРКА НА ОКОНЧАНИЕ
	23	0098		INC HL ; ПОВТОРНАЯ ЗАГРУЗКА В HL НАЧ.
	20F2	0099		JR NZ,ADT100 ; НЕ КОНЕЦ, ЗНАЧИТ ЦИКЛ
	2A0000'	0100	ADT200	LD HL,(START) ; АДРЕСА
		0101	ADT300	; ВЗЯТИЕ ОЧЕРЕДНОГО БАЙТА
	7E	0102		CP A,H ; ПРОВЕРКА СТАРШЕГО БАЙТА АДРЕСА
	BC	0103		200uF ; НЕПРАВИЛЬНО!
	0055'	0104		CALL CHLDE ; ПРОВЕРКА НА ОКОНЧАНИЕ
	0057'	0105		RET Z ; ИНКРЕМЕНТ УКАЗАТЕЛЯ
	CD3600'	0106		INC HL ; ВЗЯТИЕ ОЧЕРЕДНОГО БАЙТА
	C8	0107		LD A,(HL) ; ПРОВЕРКА СТАРШЕГО БАЙТА АДРЕСА
	23	0108		CP A,L ; НЕПРАВИЛЬНО!
	005A'	0109		CALL CHLDE ; ПРОВЕРКА НА ОКОНЧАНИЕ
	005B'	0110		RET Z ; ЦИКЛ ДЛЯ ВСЕИ ПАМЯТИ
	005C'	0111		JR ADT300 ; JP \$
	005D'	0112		END
	005E'	0113	ERROR	
	CD3600'	0114		
	C8			
	18ED			
	C36600'			
	(0000)			
	0066'			
	0069'			

Программа 5.1. Программа QUIKTST для тестирования ЗУПВ методом «шахматной доски» (перепечатано с разрешения фирмы Comart Ltd.).

включает несколько функциональных узлов; так, интерфейсный адаптер 6522 содержит два параллельных порта с четырьмя линиями квитирования, двумя счетчиками-таймерами и последовательно-параллельно-последовательным сдвиговым регистром. Один из способов частичного тестирования, которые могут быть применены для этого устройства, состоит в контроле выполнения чтения и записи содержимого тех регистров, которые снабжены данными функциями. Второй подход, часто



Рис. 5.5. Тестовая колодка для проверки портов (соединение PA0 с PB0, PA1 с PB1 и т. д.).

используемый для контроля ИС, содержащих параллельные порты или универсальные асинхронные приемопередатчики (УАПП), требует установки специальной колодки, которая соединяет соседние порты (рис. 5.5). Одному из портов придаются при этом функции входного, а другому — выходного. Черезпорт вывода передается тестовый набор данных, после чего контролируется его прием портом ввода. Затем порты меняются ролями и тест повторяется.

Заключение

Если центральная часть системы функционирует должным образом и, в частности располагает возможностями индикации информации, можно использовать ее для генерации тестовых наборов данных и контроля работоспособности памяти и средств ввода-вывода.

Глава 6

Логический анализ

Один из подходов к поиску неисправностей устройств с программным управлением состоит в контроле текущей информации на системных шинах в процессе выполнения известных документированных программ. Информация, имеющая место на шинах,

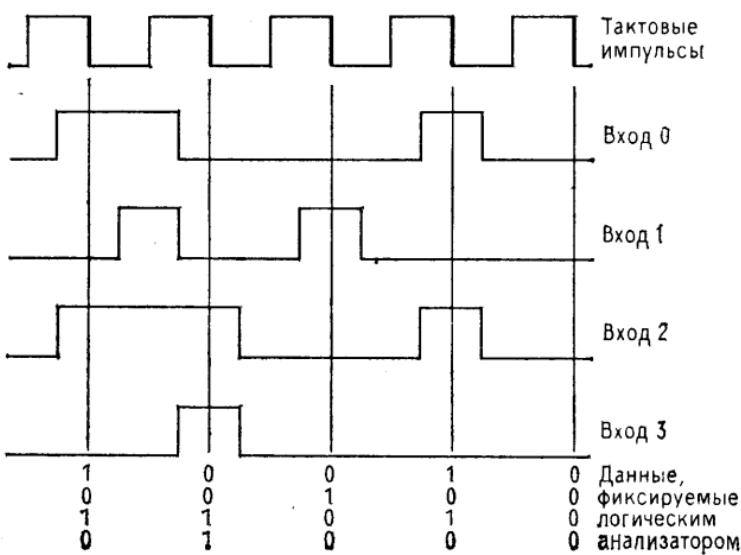


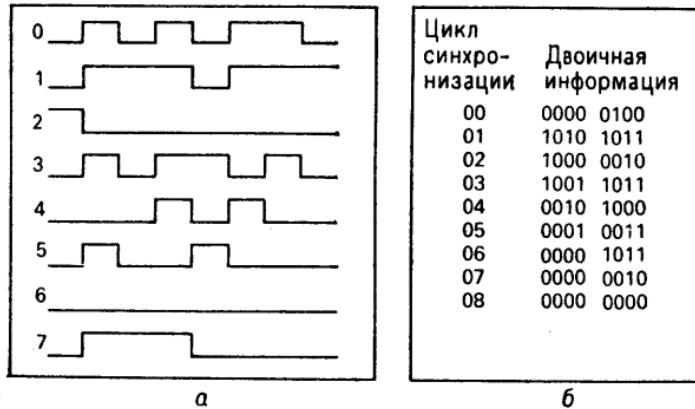
Рис. 6.1. Прием информации логическим анализатором при поступлении активных фронтов тактовых сигналов.

на каждом шаге работы системы сравнивается с данными, взятыми из листинга программы, что позволяет выяснить, правильно ли система выполняет программу. Эта процедура называется трассировкой программы.

Для наблюдения информации на шинах могут быть использованы разные приборы, но наиболее удобными являются логические анализаторы. Логический анализатор обычно имеет 8, 16 или 32 входных канала, с помощью которых можно записать поведение во времени любого из сигналов, имеющих место в микропроцессорной системе, например:

- на шине данных;
- на адреснойшине;
- на шине управляющих сигналов;
- в портах ввода-вывода.

В отличие от осциллографа с электронно-лучевой трубкой логический анализатор не отслеживает входные сигналы непре-



a

б

Адреса, данные				
Такты регист- раций	0006H	INC	HL	
+001	0007H	CP	L	
+002	0008H	JP	NZ, ****	
+003	0005H	LD	[HL], B	
+004	0006H	INC	HL	
+005	0007H	CP	L	
+006	0008H	JP	NZ, ****	
+007	0005H	LD	[HL], B	
+008	0006H	INC	HL	
+009	0007H	CP	L	
+010	0008H	JP	NZ, ****.	
+011	0005H	LD	[HL], B	
+012	0006H	INC	HL	
+013	0007H	CP	L	
+014	0008H	JP	NZ, ****	
+015	0005H	LD	[HL], B	

в

Рис. 6.2. Различные формы представления информации в логических анализаторах. а — в виде временных диаграмм; б — в двоичной форме; в — в мемонической форме.

рывно, а воспринимает их значения в момент получения активного фронта тактового сигнала (рис. 6.1). Фиксируемая при этом информация не выводится сразу же на экран, а заносится в собственную память анализатора, откуда она может быть извлечена и рассмотрена на досуге.

В большинстве анализаторов предусмотрена возможность выбора режима индикации наблюдаемых данных. На рис. 6.2 показаны три наиболее широко применяемых способа их пред-

ставления. Способ, иллюстрируемый рис. 6.2, а, аналогичен применению многоканального осциллографа. На рис. 6.2, б информация дана в виде таблицы состояний, данные которой могут быть представлены в шестнадцатеричной, восьмеричной или двоичной системе счисления. Некоторые анализаторы снабжаются программами-дисассемблерами, ориентированными на один или несколько типов микропроцессоров. С помощью дисассемблера информация, взятая с шин адресов и данных, может быть представлена в удобной мнемонической форме, как показано на рис. 6.2, в.

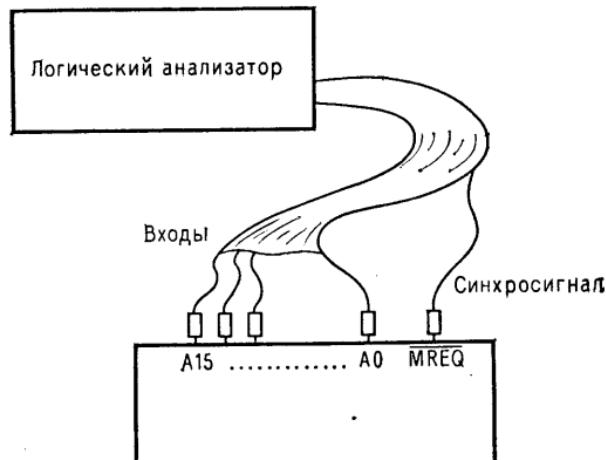


Рис. 6.3. Регистрация информации, снимаемой с адресной шины Z80.

Выпускаемые промышленностью анализаторы бывают двух видов:

- анализаторы временных диаграмм;
- анализаторы доменов данных или состояний.

В анализаторе временных диаграммрабатываются собственные сигналы записи, формируемые от внутреннего генератора, рабочая частота которого выше, чем частота сигналов исследуемой микропроцессорной системы. Так как запись данных в анализатор происходит только по активным фронтам ТИ, любые изменения значений сигналов фиксируются только при поступлении очередного такого фронта. Таким образом, максимальное время, в течение которого изменение входного сигнала может оставаться незамеченным (это явление называется ошибкой квантования по времени), составляет один период тактирования. Чтобы эта ошибка уменьшалась, частота взятия значений информации должна в несколько раз превышать частоту исследуемых сигналов. Это достигается за счет использования тактовых генераторов, рабочие частоты которых достигают величин порядка 200 МГц.

В анализаторах состояний используется другой принцип отбора данных. Вместо того чтобы брать данные по сигналам внутреннего генератора, их ввод в анализатор осуществляется по тактовым сигналам, получаемым из исследуемой системы. Этот подход иллюстрируется рис. 6.3. Анализатор состояний подключен к адресной шине системы, построенной на основе Z80, и используется для наблюдения функционирования шины данных. Вход синхронизации анализатора соединен со стробирующими сигналом MREQ процессора, причем анализатор настроен таким образом, что воспринимает отрицательный фронт этого сигнала в качестве синхронизирующего сигнала. Рассмотрение временных диаграмм Z80 показывает, что выбор этого момента цикла для взятия информации с адресной шины обеспечивает прием анализатором установленногося значения любого адреса (включая адреса регенерации).

Основные функции логических анализаторов

Диапазон возможностей логических анализаторов, как и всех изделий, построенных на основе микропроцессоров, продолжает расти. В этом подразделе будут рассмотрены лишь функции, присущие самым разнообразным приборам.

На рис. 6.4 показан типичный вид лицевой панели 16-канального анализатора состояний. Анализатор соединяется с проводящей системой посредством кабеля, проводники которого оканчиваются либо отдельными контактными щупами, либо тестовыми зажимами корпуса ИС. Наряду с 16 входами данных предусмотрены соединения для синхросигналов, «земли» и других управляющих сигналов, называемых квалифициаторами (их назначение будет пояснено ниже). Внутри анализатора имеется быстродействующее ЗУПВ, служащее для запоминания значений сигналов. В 16-канальном анализаторе эта память организована в виде 16-разрядных слов, количество которых обычно составляет 128, что позволяет хранить информацию о 128 циклах синхронизации (рис. 6.5). Память выполняет две функции: первая — запись данных, когда прибор работает в режиме приема данных, вторая — воспроизведение зарегистрированных данных на дисплее анализатора.

Для повышения гибкости и предоставления пользователю возможности наблюдать выполнение конкретных участков программ, а также просматривать определенные последовательности данных анализатор содержит схемы запуска, т. е. идентификации событий, при появлении которых включается индикация. Находясь в режиме приема, анализатор непрерывно воспринимает информацию на своих входных каналах, а при появлении запускающего слова начинается индикация данных.

В зависимости от сложности анализатора на запускающее слово могут быть наложены те или иные дополнительные условия. Например, запуск может происходить после

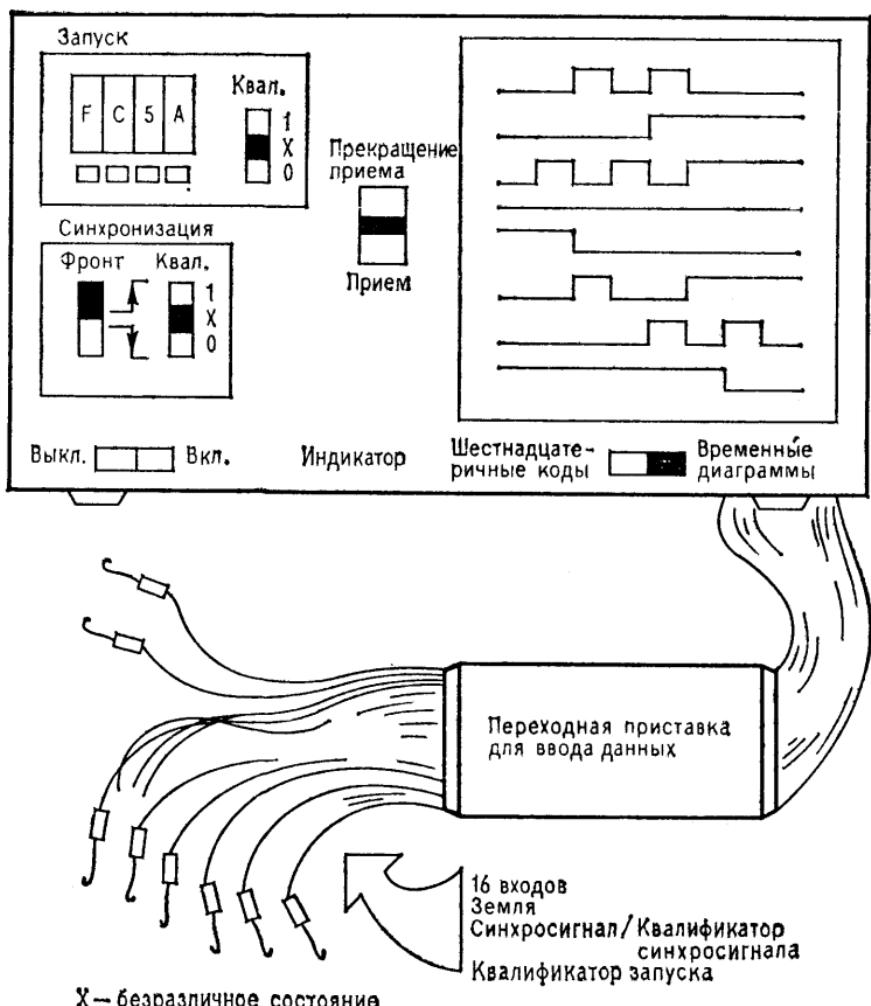


Рис. 6.4. Логический анализатор.

— регистрации заданного числа N появлений запускающего слова,

— прохождения заданного числа N циклов синхронизации вслед за появлением запускающего слова.

Как и осциллографы, логические анализаторы могут выводить на индикацию данные после возникновения запускающего события. Но в отличие от большинства осциллографов анализаторы позволяют представлять на экране дисплея информацию, имевшую место до запускающего события. Этот режим работы

анализатора, называемый регистрацией с обратным отсчетом времени, может быть использован в процессе восстановления

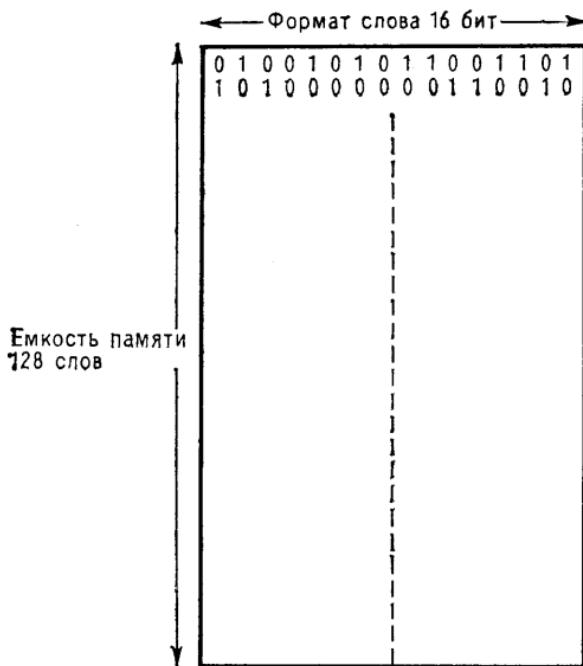


Рис. 6.5. Формат памяти анализатора.



Рис. 6.6. Простая модель, поясняющая «отрицательный отсчет времени».

работоспособности неисправной системы для задания неправильно выполняемой операции в качестве запускающего события и последующего наблюдения событий, приводящих к этому некорректному событию. Концепцию отрицательного отсчета

времени легко понять, представив 128 областей памяти анализатора расположеными по кругу, как показано на рис. 6.6. Находясь в режиме приема данных, анализатор записывает данные, а также «следит», когда произойдет запускающее событие. По каждому синхросигналу данные заносятся в очередную свободную область памяти. Как только происходит запускающее событие, анализатор помечает положение текущей области памяти и продолжает записывать информацию еще по 63 синхросигналам, перед тем как перейти в режим индикации. Выводимые при этом на индикацию данные содержат информацию о 64 состояниях до запускающего события и о 63 состояниях после него.

Органы управления, расположенные на лицевой панели анализатора, можно разделить на три группы:

- управление началом и прекращением приема;
- управление синхросигналом и его квалифицированием;
- управление запуском и квалифицированием запуска.

Начало и прекращение приема

Когда переключатель приема устанавливается в активное положение, анализатор приходит в режим сбора данных. По каждому синхросигналу в память заносится слово данных, составленное из значений 16 логических входных сигналов. Это происходит до момента распознавания запускающего события, после чего регистрация продолжается еще в течение 63 циклов синхронизации, а затем анализатор возвращается в режим индикации. Если запускающее событие не обнаруживается, можно вернуть анализатор в этот режим, установив переключатель в положение «Прекращение приема».

Синхросигнал и квалифицированный синхросигнал

С помощью переключателя фронта синхросигнала оператор может выбрать для управления записью данных в память анализатора либо положительный, либо отрицательный фронт внешнего сигнала синхронизации. Работая с анализатором состояний, пользователь должен выбрать для синхронизации регистрируемых данных подходящий сигнал. При выборе следует учитывать два важных параметра анализатора: время установки данных и время удерживания данных. Первая из этих характеристик — это время, в течение которого данные должны иметь установленные значения до поступления записывающего фронта (обычно 35 нс), а вторая определяет минимальное время сохранения неизменными значений данных после поступления

записывающего фронта (во многих случаях эта величина может иметь нулевое значение).

Возможны ситуации, когда данные необходимо запоминать при поступлении не каждого синхросигнала. Для указания тех синхросигналов, по которым необходима регистрация данных, может быть использован дополнительный входной сигнал, называемый квалифициатором синхросигнала. Этот входной сигнал реализуется в соответствии с положением переключателя на лицевой панели, позволяющего оператору указать состояние

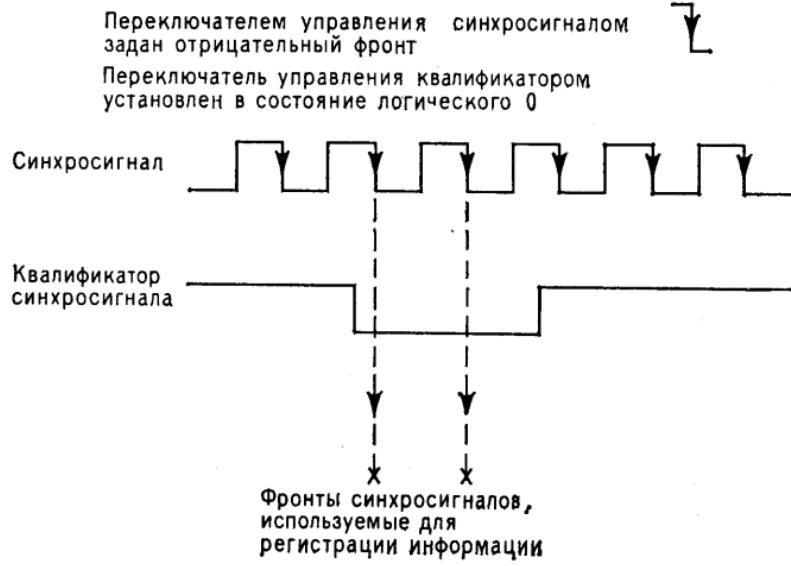


Рис. 6.7. Действие квалифициатора синхросигнала.

квалифициаторного входного сигнала, при котором происходит прием данных (рис. 6.7). В рассмотренном выше примере, где анализатор использовался для регистрации информации адресной шины типовой микропроцессорной системы на базе Z80, синхросигналом служил отрицательный фронт сигнала MREQ. В память анализатора заносились все адреса, т. е. адреса обращений к памяти и адреса регенерации, вырабатываемые микропроцессором Z80. Если бы понадобилось запоминать только адреса обращений к памяти, можно было бы исключить из регистрации адреса регенерации, подав на вход квалифициатора синхросигнала выходной сигнал процессора RFSH и установив переключатель управления квалифициатором синхросигнала в состояние логической 1. При этом только отрицательный фронт сигнала MREQ (соответствующий адресу обращения к памяти) будет использоваться для ввода данных в анализатор. Если переключатель квалифициатора синхросигнала оставлен

в «безразличном» состоянии, запись адресов в память анализатора осуществляется по каждому синхросигналу.

Запуск и квалификатор запуска

Органы управления запуском позволяют оператору задавать набор единиц и нулей, который должен появиться на входных каналах анализатора, чтобы произошел запуск. Для расширения этого набора, т. е. формата запускающего слова, до 17 бит можно использовать дополнительный входной сигнал, называемый квалификатором запуска. На лицевой панели имеется переключатель управления квалификатором, с помощью которого оператор задает значение дополнительного входного сигнала, при котором должен происходить запуск.

Тестирование матричной клавиатуры

Применение логических анализаторов не обязательно должно ограничиваться прослеживанием изменений шинных сигналов ЭВМ. Рассматриваемый ниже пример иллюстрирует использование анализатора для наблюдения некоторых сигналов, которыми ЭВМ обменивается с одним из основных периферийных устройств, а именно — клавиатурой.

Большинство клавиатур состоит из набора клавиш, организованных в виде матрицы (см. рис. П2.3). Для сопряжения матрицы переключателей с ЭВМ обычно применяется один из двух методов.

(A) *Аппаратное кодирование*. В этом случае используется специализированная ИС шифратора клавиатуры, выполняющая следующие функции:

- обнаружение нажатой клавиши и выработка стробирующего сигнала для ЭВМ;
- генерация ASCII-кода, соответствующего нажатой клавише, и выдача его в ЭВМ в параллельной форме;
- устранение дребезга контактов клавиатуры при их замыкании и размыкании;
- другие действия, возможность реализации которых предусмотрена во многих современных клавиатурах, например одновременное нажатие нескольких клавиш и автоматическое повторение посылаемых символов.

(B) *Программное кодирование*. При данном подходе, который находит применение во многих недорогих микроЭВМ, функции специализированной кодирующей ИС реализуются микропроцессором системы в сочетании со стандартными программами.

Устройство клавиатуры с программным кодированием показано на рис. 6.8. Матрица переключателей сопрягается с процессором через комбинированное устройство, содержащее ЗУПВ, два порта ввода-вывода, таймер и некоторые другие узлы и оформленное конструктивно в виде ИС с 40 внешними выводами. Оба порта используются для обслуживания клавиатуры: порт А выполняет роль порта вывода и связан со столбцами переключательной матрицы, а порт В является портом ввода и воспринимает информацию от строк матрицы. Набор сопротивлений смещения обеспечивает считывание портом В слова FF_{16} , когда не нажата ни одна клавиша.

Чтобы обнаружить нажатую клавишу и определить ее положение, микропроцессор выполняет программу просмотра переключательной матрицы. Просмотр складывается из двух частей. Сначала процессор выдает набор битов через порт А на столбцы матрицы. Затем он выполняет чтение информационного слова из порта В и сравнивает его с FF_{16} , чтобы обнаружить, нажата ли какая-либо клавиша. Если бы, например, процессор выводил через порт А слово 00_{16} (при этом на контакты всех переключателей, показанные на схеме справа, т. е. соединенные со столбцами матрицы, подаются логические нули), нажатие любой клавиши приводило бы к изменению информации, поступающей в порт ввода В. Комбинация битов, устанавливаемая в порте В, зависит от того, в какой строке матрицы находится нажатая клавиша, и не зависит от того, в каком она находится столбце. Поэтому микропроцессор не выводит в порт А слово 00_{16} , а осуществляет перебор выводимых слов с таким расчетом, чтобы на все столбцы поочередно подавались нули. В этом случае по положению логического 0 во входном слове, поступающем через порт В, процессор определяет строку, в которой находится нажатая клавиша, а анализируя положение 0 в слове, которое он вывел в ходе перебора, выясняет, к какому столбцу принадлежит эта клавиша. Программные средства обслуживания клавиатуры включают также программы для устранения дребезга переключателей, обработки ситуаций, когда одновременно нажимаются несколько клавиш, реализации функций клавишей сдвига и управления.

Подключение логического анализатора

Для слежения за перебором слов клавиатурной матрицы восемь входов анализатора подсоединены к порту В. Управляющие входы анализатора подключены следующим образом:

1. Вход синхронизации соединен с контактом CS2 (выбор кристалла) устройства, и переключателем задается формирование синхросигналов по отрицательным фронтам сигналов CS2.

Таким образом, каждый раз, когда процессор обменивается информацией с терминалом, анализатор получает синхросигнал.

2. Вход квалификатора синхросигнала соединен с контактом \overline{RS} (выбор ЗУПВ) устройства, переключатель квалификатора установлен в положение, соответствующее логической 1. В комбинированном устройстве не все обращения адресованы к портам. Сигнал \overline{RS} имеет низкий уровень, когда выбрано ЗУПВ, и высокий, когда обращения происходят к портам. Благодаря установке переключателя управления квалификатором

Двоичный код
выходного слова
процессора

7F	0111 1111	
7F	0111 1111	
BF	1011 1111	Восприятие работы порта А
BF	1011 1111	логическим анализатором
DF	1101 1111	
DF	1101 1111	
:	:	
FE	1111 1110	
FE	1111 1110	

Перемещение нуля в слове порта А
с целью поиска нажатой клавиши

Рис. 6.8. Применение логического анализатора для просмотра набора входных кодов клавиатуры.

синхросигнала в состояние логической 1 анализатор регистрирует данные только при обращениях к портам комбинированного устройства.

3. В качестве слова запуска установлено число $7F_{16}$, позволяющее выявить начало перебора выходных слов процессора, содержащих «рабочий нуль». Состояние 8 старших битов запускающего слова не имеет значения, переключатель квалификатора запускающего слова также устанавливается в «безразличное» положение.

После приема данных происходит запуск анализатора и на экране индицируется трасса, показанная на рис. 6.8. Видно, что логический 0 перемещается по всем восьми столбцам по мере того, как микропроцессор выполняет полный просмотр клавиатуры. Каждая комбинация битов воспроизводится два раза: при выводе ее процессором и при выполнении операции чтения содержимого порта В.

Заключение

Логический анализатор воспринимает информацию с групп линий шин. Существуют два основных типа логических анали-

заторов: анализаторы состояний и анализаторы временных диаграмм. Приборы первого типа предоставляют возможность прослеживать функционирование шины или порта с временными интервалами, определяемыми любым сигналом исследуемой системы. С помощью анализаторов второго типа можно анализировать критические временные соотношения шинных сигналов; они используются главным образом в качестве средств отладки аппаратуры.

Глава 7

Поиск неисправностей в последовательных интерфейсах

Отыскание неисправностей в системах с последовательной передачей информации усложняется мультиплексированием данных и использованием в интерфейсных схемах уровней напряжения, которые отличаются от стандартных, применяемых в логических схемах. Некорректное функционирование, обнаруженное в приемной части системы, может вызвать неисправность схем параллельно-последовательного и последовательно-параллельного преобразования. Источником ошибок при передаче данных могут быть помехи, влияющие на работу линий передачи и источников питания интерфейсных схем. Знание возможных неисправностей обусловливает применение для тестирования последовательного интерфейса определенных методов, позволяющих свести поиск этих неисправностей к ограниченным частям схемы интерфейса.

Общие методы

Так как пользователь имеет возможность в широких пределах варьировать номенклатуру устройств, связываемых последовательным интерфейсом, могут иметь место явные отказы, имеющие причиной несовместимость устройств, находящихся по разные стороны линии связи. Поэтому особенно важно выполнить простую предварительную проверку конфигурации системы перед тем, как строить предположения о неисправностях схем и пытаться искать эти неисправности с помощью методов, которые будут описаны ниже.

Нередки случаи, когда система, только что удовлетворительно функционировавшая, по непонятным причинам перестает работать, и лишь после многочасовых безрезультатных попыток отыскания несуществующей неисправности удается обнаружить, что кто-то нечаянно изменил положение переключателя скорости передачи. Кроме скорости передачи с помощью переключателя или программно задается формат передаваемых символов. В небольших системах, где для последовательно-параллельного и

0001 ;
0002 ;
0003 ;
0004 ;
0005 ;
0006 ;
0007 ;
0008 ;
0009 ;
0010 ;
0011 ;
0012 ;
0013 ;
0014 ;
0015 ;
0016 ;
0017 ;
0018 ;
0019 ;
0020 ;
0021 ;
0022 ;
0023 ;
0024 ;
0025 ;
0026 ;
0027 ;

COMART LTD.
8 LITTLE END ROAD
EATON SOCON
ST.NEOTS
CAMBRIDGESHIRE
0008
СЕНТЯБРЬ 1980

ЭТО ПРОГРАММА БЫСТРОГО ТЕСТИРОВАНИЯ ПЛАТ ПАМЯТИ
СОМАРТ.ОНА ПОСТРОЕНА Т.О., ЧТОБЫ ОПЕРАТОР ВВОДИЛ
ОБЪЕКТНЫЙ КОД, ЗАДАВАЛ НАЧАЛЬНЫЙ И КОНЕЧНЫЙ АДРЕСА,
И ВЪЗПОЛНЯЕТСЯ ПОД УПРАВЛЕНИЕМ МОНИТОРА, НАПРИМЕР
NORTH ИЛИ ОТЛАДЧИК СРОМЕМСО.

ПРОГРАММА НАЧИНАЕТСЯ С АДРЕСА 10C(16), НО МОГУТ БЫТЬ
ЗАДАНЫ И ДРУГИЕ НАЧАЛЬНЫЕ ТОЧКИ, КОТОРЫЕ
УСТАНАВЛИВАЮТСЯ ЛИБО ОПЕРАТОРОМ, ЛИБО ПУТЕМ
ОБРАЩЕНИЯ К ЛИСТИНГУ СОМАРТ ПО УКАЗАННОМУ НАЧ.АДРЕСУ.
ПРОГРАММА ПРЕДПОЛАГАЕТ, ЧТО НАЧАЛЬНЫЙ И КОНЕЧНЫЙ АДРЕСА
НАХОДЯТСЯ СООТВЕТСТВЕННО В ОБЛАСТИХ ORIGN И ORIGN + 2
(ЭТИ АДРЕСА ДОЛЖНЫ ХРАНИТЬСЯ В ИНТЕЛЛОВСКОМ ОБРАТНОМ
ФОРМАТЕ).ПРОВЕРКА ДОСТОВЕРНОСТИ АДРЕСОВ НЕ ПРОИЗВОДИТСЯ.
НЕОБХОДИМО ВЫДЕРЖИВАНИЕ УСЛОВИЯ: START + i < END
ОБЛАСТЬ ПРОГРАММЫ НЕ ДОЛЖНА НАКЛАДЕВАТЬСЯ НА
ТЕСТОВУЮ ОБЛАСТЬ!!!

УСПЕШНЫЙ ТЕСТ ЗАВЕРШАЕТСЯ АДРЕСОМ TESTEND, ПО КОТОРОМУ
МОЖНО РАЗРЕШИТЬ ПЕРЕХОД К МОНИТОРУ ИЛИ ПРЕРЫВАНИЕ, ЕСЛИ
ПРЕДУСМОТРЕНА ЕГО ВОЗМОЖНОСТЬ.

обратного преобразования УАПП не используются, скорость передачи данных определяется тактовой частотой процессора; поэтому изменение частоты тактирования, являющееся, например, частью процедуры отыскания какой-то другой неисправности, может вызвать отказ последовательного канала связи.

Приведем перечень задаваемых параметров, которые целесообразно проверить, чтобы избежать потерь времени и сил в поисках несуществующих неисправностей:

- 1) скорость передачи;
- 2) число битов данных;
- 3) вид контроля по четности;
- 4) количество стоповых битов.

Эти четыре характеристики в общем случае должны быть заданы одинаковыми на обоих концах линии связи, хотя форматы данных и могут несколько различаться без ущерба для качества передачи. Если, например, передатчик посыпает с каждым символом два стоповых бита, а приемнику требуется всего один, искажений передаваемой информации не произойдет; передатчик, генерирующий бит четности, будет успешно работать в сочетании с приемником, который не осуществляет контроля по четности, если длина слова в приемнике установлена с учетом бита четности в передаваемых данных.

Перечисленные проверки в системах, в которых характеристики передачи задаются с помощью переключателей, необходимо выполнять даже до включения питания. В системах с программным заданием параметров передачи их значения на обоих концах канала связи должны быть проверены сразу после подачи питания, а в случае внесения изменений необходимо убедиться после отключения и повторной подачи питания, что эти изменения отражены в памяти (например, путем занесения новых значений в память на дисках). Когда в систему подано питание, с помощью простых приборов можно выполнить еще некоторые проверки. В системах передачи с односторонним размещением источника питания, таких, как RS232, V24 и RS423, линия передачи должна иметь отрицательный потенциал по отношению к земле во время отсутствия пересылки данных. Это напряжение можно проверить с помощью вольтметра или осциллографа, и если оно не соответствует норме, значит, имеют место неисправности либо в заземлении, либо в интерфейсной схеме передатчика, либо, что наиболее вероятно, в источнике питания интерфейсной схемы передатчика. В системах типа RS232 напряжение питания 12 В иногда используется только для последовательного интерфейса и неисправности источника проявляются лишь при попытке осуществить передачу данных. В системах с токовой петлей сведения о наличии неисправности можно получить с помощью миллиамперметра, включенного в петлю (при

отсутствии передачи в ней должен протекать ток 20 мА). Для быстрой проверки системы такого типа целесообразно несколько раз подряд отключить и снова включить разъем линии, в результате чего происходит случайное замыкание и размыкание цепи связи и, соответственно, генерация произвольных символов (это удобный способ контроля при наличии устройства визуального отображения информации).

Если в ходе простых проверок целостности кабелей и разъемов никаких неполадок не выявляется, следует воспользоваться более узко специализированными приемами контроля проверок.

Специальные методы контроля последовательного интерфейса

Если с помощью описанных выше методов предварительного тестирования восстановить работоспособность последовательной линии связи не удается, следует использовать одно или несколько из следующих средств:

1. Анализаторы последовательных данных.
2. Специализированные генераторы данных.
3. Программы контроля с возвратом данных.

Анализаторы последовательных данных

Это приборы, по принципам работы аналогичные логическим анализаторам (некоторые логические анализаторы располагают возможностью анализа последовательных данных как основной или дополнительно реализуемой характеристикой). Так как анализируемые данные представляются в виде либо логических сигналов, либо сигналов интерфейсного характера, анализатор должен быть снабжен пробником, обеспечивающим возможность работы с самыми разнообразными уровнями сигналов и задания положительной или отрицательной логики кодирования воспринимаемых сигналов. Последовательные данные обычно преобразуются в параллельную форму и заносятся в память, причем сигнал окончания преобразования каждого полученного блока данных используется в качестве синхросигнала для анализирующей части прибора. Схема анализатора последовательных данных представлена на рис. 7.1. Конечно, для того чтобы корректно выполнялось последовательно-параллельное преобразование и были достоверны сохраняемые в памяти данные, необходимо должным образом выбрать скорость передачи и формат данных. Данные обычно непрерывно заносятся в память до тех пор, пока не появится заданное запускающее слово, после чего происходит запоминание еще некоторого определенного количества данных и запись прекращается.

Последовательные анализаторы с экранами на ЭЛТ обычно снабжены возможностью индикации данных в двоичной и шестнадцатеричной форме, а также в коде ASCII. На рис. 7.2 показан

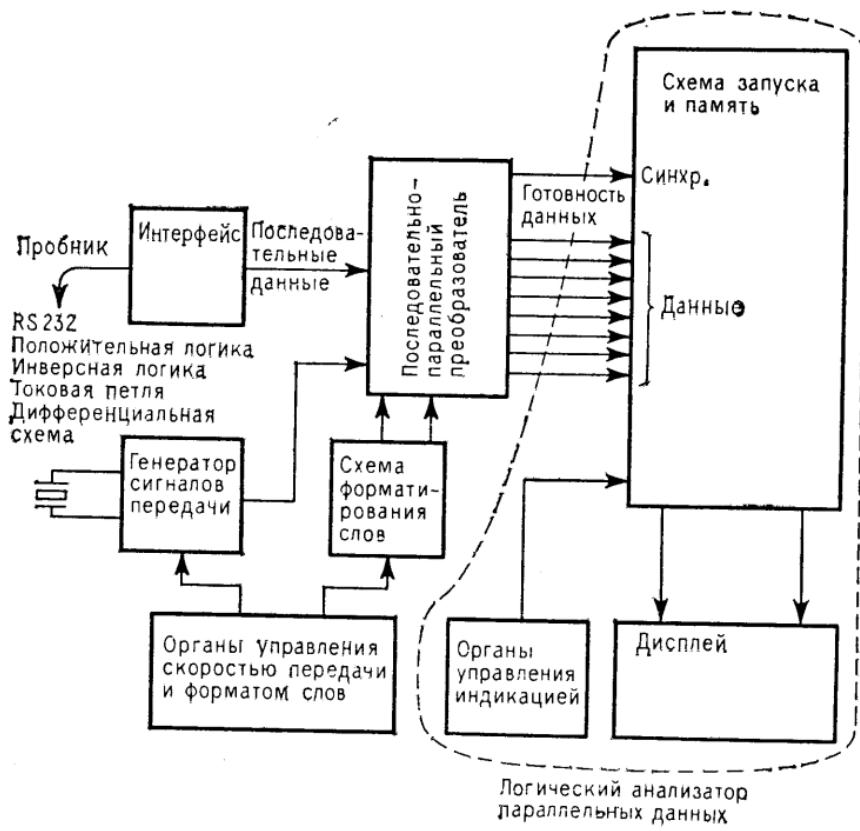


Рис. 7.1. Анализатор последовательных данных.

зано, каким образом анализатор последовательных данных может быть использован для отыскания неисправностей.

Специализированные генераторы данных

Для снижения доли неопределенности, имеющей место в процессе отыскания неисправностей в системе последовательной передачи данных, целесообразно иметь отдельный прибор, который генерирует известные потоки данных с управляемыми скоростью и форматом передачи. Такие генераторы символов удобно использовать для проверки связей с периферийными устройствами, так как они позволяют установить, где имеет место неисправность: либо в канале связи или периферийном устройстве, либо в самой микропроцессорной системе. С помощью генератора можно, кроме того, подавать на вход системы определенные последовательности данных и проверять ее реакцию на заранее

известные входные воздействия. Если специализированного генератора данных нет, для указанных целей можно применить дисплей с клавиатурой. В генераторе символов обычно предусмотрена возможность задания пользователем не только скорости передачи, вида контроля по четности и формата символов, но и выбора одного из тестовых режимов работы. К числу типичных тестов проверки работы системы относятся следующие:

Если имеется доступ
к параллельной
информации, здесь
может быть применен
логический
анализатор

Нормальный уровень
сигнала на линии — низкий

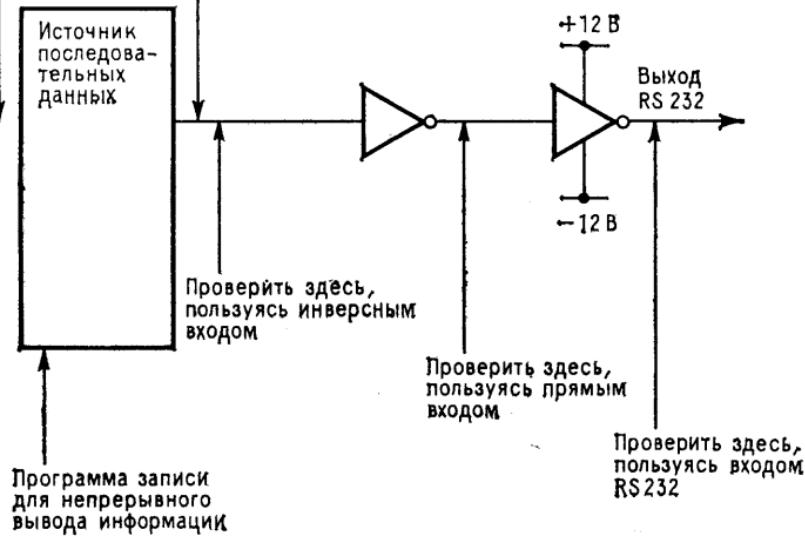


Рис. 7.2. Трассировка последовательных данных в системе.

1. Печатные символы набора ASCII, как правило, отображаются на экране в виде строк по 80 символов, а иногда имеется возможность использования формата, содержащего 132 столбца. Для тестирования устройств наглядного отображения, таких, например, как принтеры, используется вывод результатов работы системы в виде печатных символов.

2. Тест типа U*U*. Эти два символа кода ASCII имеют дополнительные коды, причем значения соседних битов в этих кодах чередуются. При попаренной последовательной передаче двух этих символов образуется последовательный набор данных, в котором количество изменений значений передаваемых битов больше, чем при использовании любых других печатных символов. Такой набор позволяет хорошо проверить работу аппаратных и программных средств ввода информации, в которых

могут иметь место неисправности, связанные с синхронизацией, так как он представляет собой худший случай сочетания битов, который может иметь место при последовательной передаче. Еще один способ достижения того же эффекта смены передаваемых последовательно битов состоит в попеременной передаче слов AA₁₆ и 55₁₆.

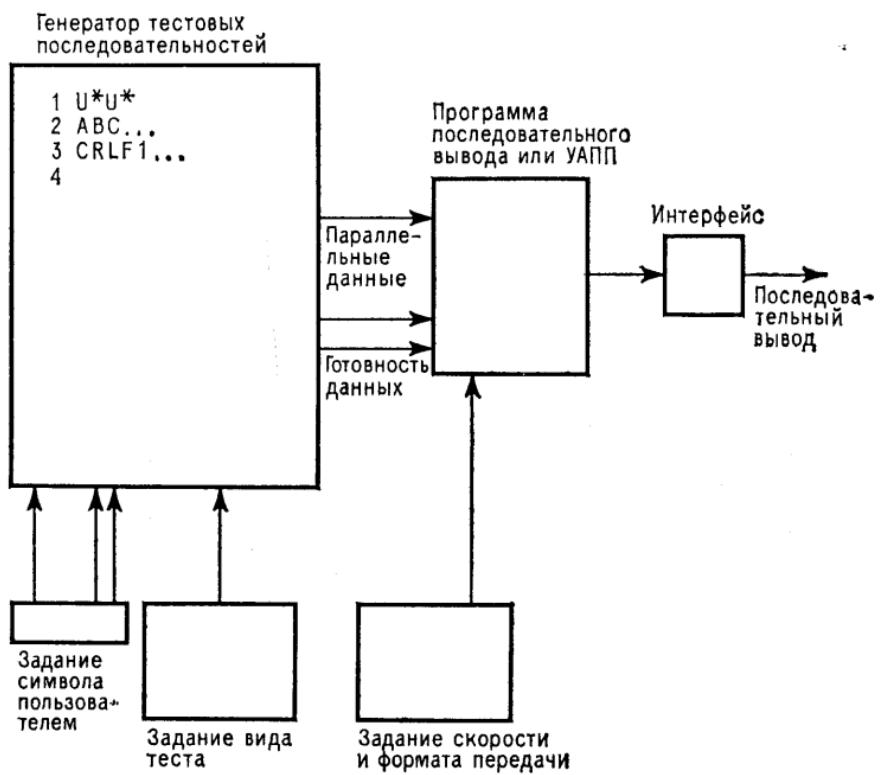


Рис. 7.3. Генератор последовательных символов.

3. Передача конкретных символов или двоичных наборов. Если известно, что проблемы возникают при передаче некоторого набора данных, для отыскания неисправности можно прибегнуть к одноразовой или повторяющейся посылке определенного байта.

4. Если в системе передачи предусмотрены входные буферы и линии квитирования, по которым передаются сообщения о заполнении буфера, целесообразно иметь в распоряжении какой-либо тест для быстрого заполнения этого буфера. Так как принтеру требуется, например, значительно большее время для перевода строки, чем для печати символа, буфер будет заполняться быстрее, когда количество символов перевода строки составляет значительную часть от общего числа передаваемых символов (то же самое справедливо и для символа возврата каретки).

в принтерах с односторонней печатью). Повторяющаяся тестовая последовательность, состоящая из двух печатных символов, сопровождаемых символами CR и LF, обеспечивает заполнение буфера, и если принтер неисправен, то в распечатке обнаружится произвольное пропадание символов. Данный тест позволяет локализовать неисправности, которые могут иметь

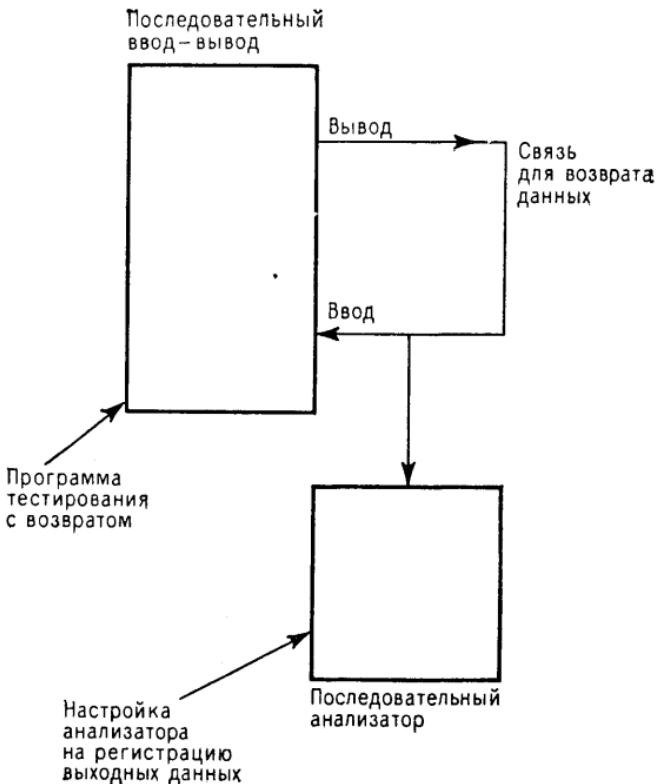


Рис. 7.4. Тестирование с возвратом данных.

место в средствах вывода информации машины или в самом принтере. (Он является, кроме того, хорошим средством проверки механизма перевода строки.) Источник символов должен, конечно, быть построен таким образом, чтобы при поступлении сигнала заполнения буфера передача символов прекращалась и выдавалось сообщение о прекращении передачи путем включения сигнальной лампы. Для тестирования входного порта, имеющего небольшой буфер, можно использовать полный набор символов. На рис. 7.3 приведена типовая схема генератора последовательных данных.

Программные средства тестирования с возвратом данных

При данном подходе средства ввода-вывода микроЭВМ подвергаются испытаниям с использованием программ самоконтроля, с помощью которых различные наборы данных сначала

выводятся, а затем принимаются и проверяются с портами вывода той же микропроцессорной системы. Этот метод может быть применен лишь для контроля систем, в которых возможна одновременная передача и прием информации, и не подходит для систем с программной реализацией последовательно-параллельных преобразований. Основной целью тестов этого типа является пересылка таких последовательностей данных, которые либо с наибольшей вероятностью приводят к возникновению ошибок передачи и приема, либо позволяют локализовать неисправность с точностью до конкретного участка системы ввода-вывода. Так как последовательный ввод-вывод обычно выполняется с использованием того или иного УАПП, тест с возвратом данных позволяет, кроме всего прочего, осуществить самоконтроль УАПП. В системах, содержащих множество линий ввода-вывода, один УАПП может быть использован для вывода информации на любой другой УАПП системы.

При данном виде тестирования для локализации неисправностей может быть дополнительно применен анализатор последовательных данных, который, например, оказывается полезным, когда тестирование с возвратом не позволяет установить место отказа из-за неисправности передатчика. Подсоединение анализатора к выходу системы может помочь установить, не в этом ли причина неправильной передачи данных. Схема подключения анализатора для этих целей показана на рис. 7.4.

Заключение

К числу возможных причин некорректной работы систем последовательной передачи данных относятся ошибки пользователя, которые могут иметь место при внесении изменений в аппаратные программные средства. Перед проведением тестирования с применением генераторов и анализаторов данных необходимо проверить правильность подключения кабелей и разъемов. Используемые наборы данных должны обеспечивать генерацию печатных символов набора ASCII для проверки правильности функционирования дисплеев и принтеров, а также проверку временных соотношений с целью выявления проблем синхронизации.

Глава 8

Внутрисхемная эмуляция

Внутрисхемная эмуляция (ВСЭ) — это мощное средство отладки, позволяющее инженеру, осуществляющему контроль микропроцессорной системы, получить к системе доступ, невозможный при обычном управлении ею с клавиатуры. Внутрисхемная эмуляция первоначально возникла как один из методов проектирования и является одним из наиболее широко используемых



Рис. 8.1. Внутрисхемная эмуляция.

компонент организаций систем проектирования микропроцессорных систем (СПМС — MDS — Microcomputer Development Systems). Однако впоследствии его развитие привело к появлению нескольких видов устройств, предназначенных в первую очередь для инженеров по обслуживанию. В данной главе рассмотрены основные принципы внутрисхемной эмуляции, а затем ее возможности проиллюстрированы примерами использования как при разработке, так и для обслуживания микропроцессорных систем. Отметим, что, как показано в гл. 11, для осуществления внутрисхемной эмуляции в условиях лаборатории, специализирующейся в обслуживании, могут быть применены существующие микроЭВМ.

К установочной колодке микропроцессора обычно подведены линии всех сигналов адреса, данных и управления, посредством которых осуществляется обращение к большинству устройств системы. Это обстоятельство используется для внутрисхемной эмуляции, когда из тестируемой системы удаляется процессор, а его функции имитируются (эмулируются) с помощью другой ЭВМ. Этот принцип иллюстрируется на рис. 8.1. Главная система подключается к проверяемой (целевой) системе посредством разъемной колодки с двухрядным расположением контактов и резинового кабеля. Обычно в главной системе используется микропроцессор, аналогичный тому, который изъят из целевой, чтобы иметь возможность передачи всех сигналов, необходимых для работы целевой системы. На первый взгляд удаление микропроцессора из целевой системы, с тем чтобы сразу заменить его таким же, имеющимся в главной, может показаться бессмысленным. Однако внутрисхемная эмуляция позволяет главной системе включиться в работу между целевой системой и ее микропроцессором, что дает возможность наблюдать работу шин целевой системы и подавать в нее необходимые сигналы.

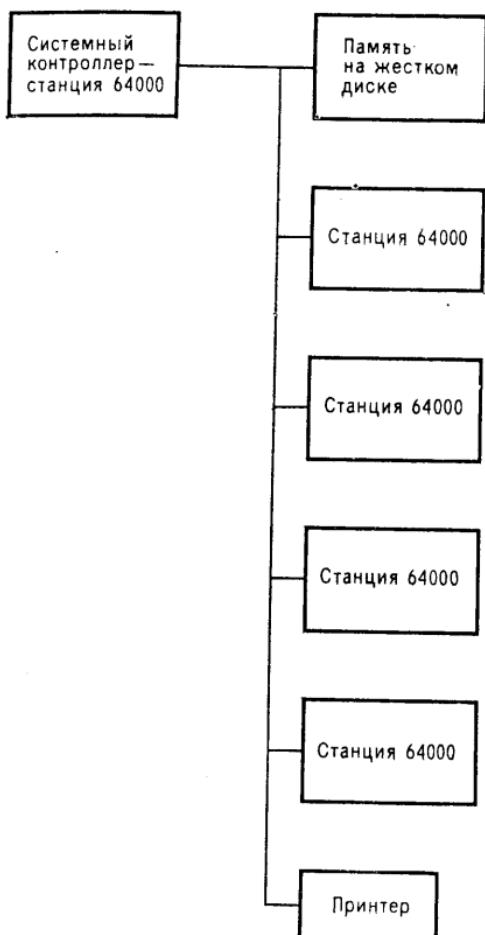
ВСЭ как инструмент проектирования

Метод внутрисхемной эмуляции был впервые применен фирмой Intel в ее системах проектирования микропроцессорных систем (СПМС) в качестве средства отладки аппаратуры и программ в ходе проектирования. Первые системы проектирования были программно ориентированными, предоставляя в распоряжение разработчика редакторы, ассемблеры, редакторы связей, а также некоторые достаточно ограниченные средства выполнения и отладки результирующего машинного кода. Однако для эффективного тестирования аппаратуры и программ необходимо было записать этот машинный код в СППЗУ перед тем, как переслать его в разрабатываемое устройство. Вся последующая необходимая отладка производилась затем с использованием логического анализатора. При обнаружении фактов некорректного функционирования инженер вновь обращался к системе проектирования, чтобы внести в программу изменения перед повторным занесением ее в СППЗУ.

Введение в состав систем проектирования внутрисхемных эмуляторов позволило уменьшить затраты времени на устранение неисправностей, дав разработчику возможность управлять работой проектируемой системы непосредственно от СПМС и избавив его от необходимости использования СППЗУ для передачи программ в целевую систему. Кроме того, так как система проектирования в ходе выполнения программы имеет доступ к шинам целевой системы, она может быть использована в качестве инструмента отладки. Для реализации этих действий боль-

шинство СПМС снабжено встроенными функциями логического анализа.

Описание «типичной» СПМС оказалось терминологически достаточно противоречивым. Диапазон их возможностей широк, и



термины, используемые для их описания различными разработчиками, могут существенным образом различаться. Поэтому в данной главе взята конкретная СПМ и на ее примере проиллюстрированы виды функций и характеристики систем, используемые для внутрисхемной эмуляции. В качестве такой конкретной системы выбрана СПМС 64000 фирмы Hewlett Packard, которая может быть применена для работы со многими 8- и 16-разрядными микропроцессорами. Крупные фирмы-изготовители микропроцессорных средств выпускают свои СПМС.

Эмуляция с использованием СПМС

Система проектирования HP64000 представляет собой с точки зрения инженера-разработчика многопользовательскую систему, базирующуюся на памяти на жестких дисках и применимую для работы с широким диапазоном микропроцессорных средств (рис. 8.2). Многопользовательская работа существенно облегчает объединение программных

Рис. 8.2. Многопользовательская система проектирования HP64000 (память на жестких дисках и аппаратно-программные станции связаны между собой интерфейсной шиной НРІВ).

средств, разрабатываемых коллективом инженеров, позволяет независимо разрабатывать отдельные фрагменты проекта и в то же время совместно использовать полученные результаты всеми членами группы разработчиков.

Как и большинство других систем проектирования, HP64000 содержит программные средства, в число которых входят следующие:

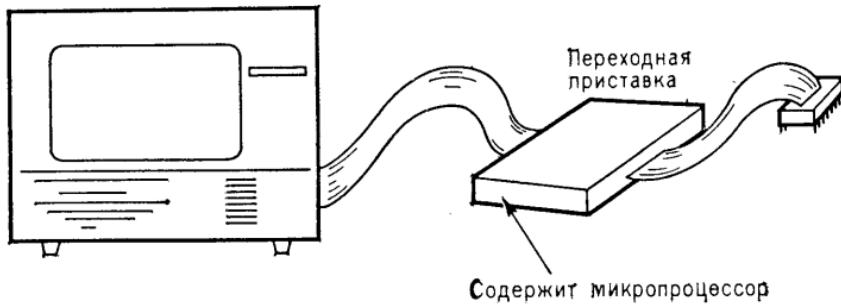


Рис. 8.3. Типичная аппаратная станция. Аппаратная станция может содержать плату управления эмуляцией, плату анализа, плату управления эмуляцией памяти, эмулирующую память.

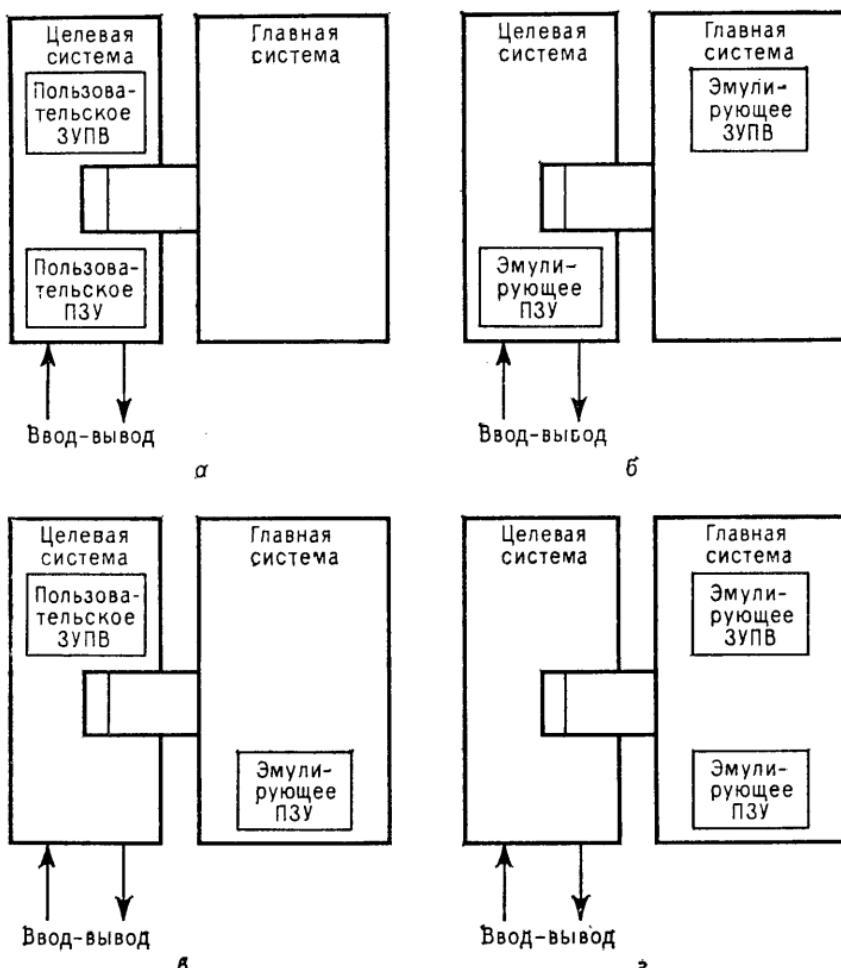


Рис. 8.4. Возможные конфигурации при эмуляции.

- редактор для написания исходных программ на языке ассемблера или на языках высокого уровня;
- ассемблеры для генерации объектного кода из программ, написанных на языке ассемблера;
- компиляторы для генерации объектного кода из программ, написанных на языках высокого уровня (в данном случае в качестве такого языка используется Паскаль);

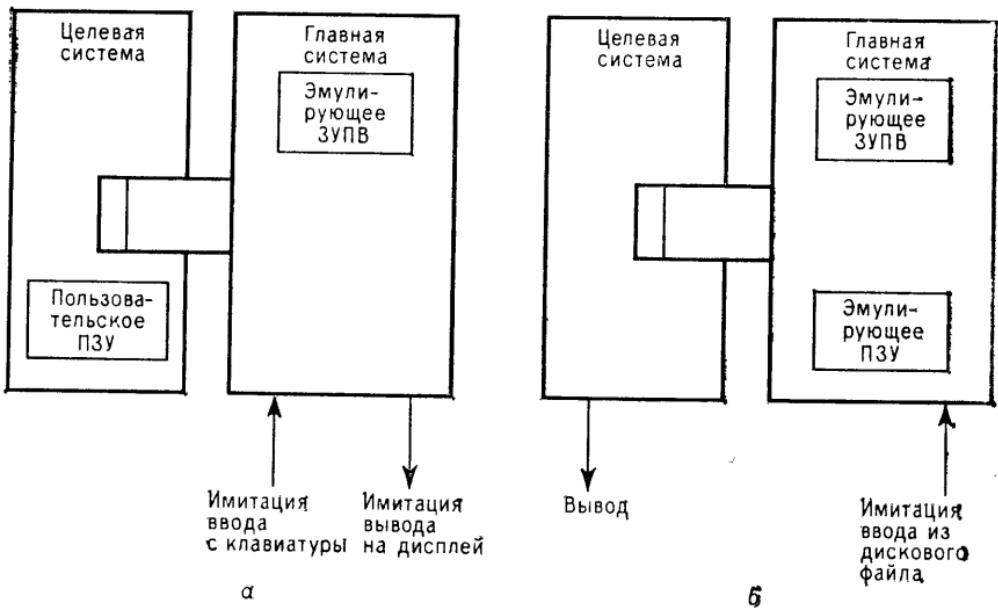


Рис. 8.5. Конфигурации при эмуляции с использованием имитации ввода-вывода.

- редактор связей для связывания программных модулей.
- Аппаратные средства НР64000 включают
- внутрисхемные эмуляторы;
- программаторы СППЗУ;
- логические анализаторы (как состояний, так и временных соотношений).

На рис. 8.3 представлена конфигурация типичной аппаратной станции, предназначеннной для выполнения внутрисхемной эмуляции, а также трассировки или логического анализа целевой системы.

Возможности эмуляции весьма многообразны и позволяют СПМ не только брать на себя роль микропроцессора, но также

- осуществлять системное тактирование целевой системы;
- выполнять функции всех или отдельных ПЗУ и ЗУПВ, входящих в состав целевой системы;
- имитировать функционирование устройств ввода-вывода.

На рис. 8.4 показаны возможные конфигурации, реализуемые при соединении главной и целевой систем. В примере на рис. 8.4, а главная система берет на себя функцию имитации только микропроцессора. Вся память, используемая целевой системой (называемая пользовательской памятью), размещается на плате целевой системы. На рис. 8.4, в показано, каким образом память целевой системы может быть скомпонована из ее собственных аппаратных средств, пользовательского ЗУПВ и памяти главной системы — эмулирующего ПЗУ.

На рис. 8.5 иллюстрируется использование периферийных устройств 64000, а именно дисковой памяти, клавиатуры, принтера и др. для имитации входных и выходных сигналов целевой системы.

Подобная гибкость организации взаимодействия главной и целевой систем обусловила появление понятия «мягкая машина», поскольку их совместная конфигурация легко приспосабливается для имитации любой части целевой системы или всей этой системы в целом.

Применение эмуляторов

Для проведения эмуляции к целевой системе подключается соответствующая переходная приставка. Эмуляционный пробник имеет такое же расположение выводов, как процессор Z80, и осуществляет передачу всех сигналов адресов/данных и управления между системой проектирования и целевой системой.

После осуществления аппаратных соединений оператор выполняет конфигурацию системы, в процессе которой главная система задает ряд вопросов о целевой системе, чтобы выяснить, какая часть последней должна эмулироваться, например:

- Хотите ли вы использовать тактовый генератор процессора, находящийся в целевой системе, или тактовые сигналы должны поступать от меня?
- Есть ли необходимость вести прогон целевой системы все время на предельной скорости или можно иногда снизить эту скорость, чтобы в ходе прогона совершить ряд таких разумных действий, как проверка содержимого регистров и пользовательской памяти?
- Должна ли я устанавливать флаг ошибки при обнаружении недопустимого кода операции?
- Укажите требуемый объем эмулирующих ПЗУ и ЗУПВ.
- Укажите объем пользовательских ПЗУ и ЗУПВ, которые будут функционировать в целевой системе.
- Необходима ли имитация ввода-вывода? Если да, то какой вид имитации вы хотите применить?

На рис. 8.6 представлен пример составления эмуляционной конфигурации для описанной в приложении 2 платы контроллера, реализованного на базе микропроцессора Z80. Эмулятор организован таким образом, чтобы использовалось ЗУПВ, находящееся на плате целевой системы (8000_{16} — $83FF_{16}$) и эмулирующее ПЗУ в главной системе (0000_{16} — $0FFF_{16}$).

После завершения конфигурации оператор может загрузить машинную программу либо в эмулирующую, либо в пользовательскую память и приступить к прогону программы или ее по-

Назначение эмулирующей и пользовательской памяти

—000 —400 —800 —C00	—000 —400 —800 —C00
---------------------	---------------------

0 — — —	Эмулирую-	8 — — —	Пользова-
1 — — —	щее ПЗУ	9 — — —	тельское ЗУПВ
2 — — —		A — — —	
3 — — —		B — — —	
4 — — —		C — — —	
5 — — —		D — — —	
6 — — —		E — — —	
7 — — —		F — — —	

Запись № 2, размер = 128

Конфигурация: область процессора № 9, область памяти № 7,
область анализа № 8

Тактирование процессора: внешнее

Темп тактирования выше порога ожидания: нет

Ограничение процессора прогоном в реальном времени: нет

Останов процессора при недопустимых кодах: нет

Имитация ввода-вывода: нет

Рис. 8.6. Пример конфигурации при эмуляции платы контроллера Z80.

шаговому выполнению. Программа 8.1 представляет собой пример программы с начальным адресом 0000_{16} в карте распределения памяти контроллерной платы Z80. При прогоне программы она использует ЗУПВ емкостью 1 Кбайт, находящееся на плате, последовательно изменяя содержимое всех ячеек памяти, в порядке $00, 01, 02, \dots, FF_{16}$ и т. д.

В процессе выполнения программы оператор может просматривать содержимое либо памяти, либо внутренних регистров процессора 80.

Например, с помощью команды

«вывести на индикацию содержимое памяти 8000_{16} »

на экран выводятся мгновенные значения ячеек используемого ЗУПВ (рис. 8.7). Если заменить эту команду на

«вывести на индикацию содержимое памяти 8000_{16} динамически», на экране появится «движущееся изображение» содержимого памяти (рис. 8.8). В ходе вывода этой информации глав-

) ФАЙЛ: CHANGE : HONEY NEWLETT-PACKARD: АССЕМБЛЕР Z80

ОБЛАСТЬ ОБЪЕКТ- НОМЕР ИСХОДНЫЙ ТЕКСТ
ПАМЯТИ НЫЙ КОД СТРОКИ

```
1      "Z80"
2 *****
3      * CHANGE          ПРИМЕР ПРОГРАММЫ ЭМУЛЯЦИИ
4      *
5      *   ПРОГРАММА ЗАПИСЫВАЕТ 00,01,02,...,FF
6      *   ВО ВСЕ ЯЧЕЙКИ ОТ 8000 ДО 83FF
7
8 *****
9
10     ORG 000H           ;
11     LD B,00H           ; РЕГИСТР В СОДЕРЖИТ ЧИСЛО
12     LD HL,8000H         ; HL СЛУЖИТ ДЛЯ УКАЗАНИЯ ЯЧЕЙКИ
13     LD [HL],B           ; ЗАПИСЬ В ЯЧЕЙКУ СОДЕРЖИМОГО В
14     LD A,0FFH           ; А СОДЕРЖИТ ПОСЛЕДНЕЕ
15     INC HL              ; ЗНАЧЕНИЕ L
16     CP L                ; СЛЕДУЮЩАЯ ЯЧЕЙКА
17     JP NZ, AGAIN        ; МЛАДШИЙ БАЙТ АДРЕСА ДОСТИГ FF?
18     LD A,83H             ; ЕСЛИ НЕТ, ТО СЛЕДУЮЩАЯ ЯЧЕЙКА
19     BC                  ; А СОДЕРЖИТ ПОСЛЕДНЕЕ
20     JP C20005            ; ЗНАЧЕНИЕ Н
21     LD [HL],B           ; МЛАДШИЙ БАЙТ АДРЕСА ДОСТИЛ 83??
22     INC B               ; ЕСЛИ НЕТ, ТО СЛЕДУЮЩАЯ ЯЧЕЙКА
23     JP C30002            ; ПОСЛЕДНЯЯ ЯЧЕЙКА
24
25     INC B               ; ОЧЕРЕДНОЕ ЗНАЧЕНИЕ ДЛЯ
26
27     JP NEXT              ; ЗАПОМИНАНИЯ
28
29     JP NEXT              ; ПОВТОРЕНИЕ ДЛЯ НОВОГО ЗНАЧЕНИЯ
```

Программа 8.1. Пример программы CHANGE, проверяющей все ячейки ЗУПВ с адресами от 8000₁₆ до 83FF₁₆.

ПАМЯТЬ									
Адрес	Данные (шестнадцатеричные)								
8000	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8008	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8010	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8018	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8020	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8028	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8030	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8038	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8040	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8048	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8050	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8058	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8060	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8068	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8070	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk
8078	6B	6B	6B	6B	6B	6B	6B	kkkk	kkkk

Рис. 8.7. Пример вывода на дисплей содержимого памяти (вывод содержимого памяти 8000₁₆).

ПАМЯТЬ - ДИНАМИЧЕСКИЙ									
Адрес	Данные (шестнадцатеричные)								
8000	00	00	00	00	00	00	00	00	00
8008	00	00	00	00	00	00	00	00	00
8010	00	00	00	00	00	00	00	00	00
8018	00	00	00	00	00	00	00	00	00
8020	00	00	00	00	00	00	00	FF	FF
8028	FF	FF	FF	FF	FF	FF	FF	FF	FF
8030	FF	FF	FF	FF	FF	FF	FF	FF	FF
8038	FF	FF	FF	FF	FF	FF	FF	FF	FF
8040	FF	FF	FF	FF	FF	FF	FF	FF	FF
8048	FF	FF	FF	FF	FF	FF	FF	FF	FF
8050	FF	FF	FF	FF	FF	FF	FF	FF	FF
8058	FF	FF	FF	FF	FF	FF	FF	FF	FF
8060	FF	FF	FF	FF	FF	FF	FF	FF	FF
8068	FF	FF	FF	FF	FF	FF	FF	FF	FF
8070	FF	FF	FF	FF	FF	FF	FF	FF	FF
8078	FF	FF	FF	FF	FF	FF	FF	FF	FF

Рис. 8.8. Пример вывода на дисплей содержимого памяти (вывод содержимого памяти 8000₁₆ динамически).

ПАМЯТЬ									
0000H	LD	B, 00H							
0002H	LD	HL, 8000H							
0005H	LD	[HL], B							
0006H	LD	A, FFH							
0008H	JNC	HL							
0009H	CP	1							
000AH	JP	NZ, 0005H							
000DH	LD	A, 83H							
000FH	CP	H							
0010H	JP	.NZ, 0005H							
0013H	LD	[HL], B							
0014H	JNC	B							
0015H	JP	0002H							
0018H	LD	E, [HL]							
0019H	LD	A, L							
001AH	LD	B, C							

Рис. 8.9. Вывод на дисплей в мнемоническом формате (вывод содержимого памяти 0006₁₆ мнемонически).

ная система самопроизвольно отклоняется от прогона на полной скорости к псевдопрогону, т. е. периодически останавливает процессор, чтобы вывести и просмотреть содержимое пользовательской памяти. Наконец, на рис. 8.9 показано, каким образом команда вывода информации на индикацию может быть использована для представления содержимого памяти в дизассемблированном, или мнемоническом, формате.

РЕГИСТРЫ (шестадцатеричные данные)

рс	код опер.		a	b	c	d	e	h	l	i	ix	iy	sp	szxhxpnc	nxt-pc
0000	06	LD	B,	00H		FF	00FF	FFF	82C7	00	FFFF	7FFF	00FF	00000010	0002
0002	21	LD	HL,	8000H		FF	00FF	FFF	8000	00	FFFF	7FFF	00FF	00000010	0005
0005	70	LD	[HL],	B		FF	00FF	FFF	8000	00	FFFF	7FFF	00FF	00000010	0006
0006	3E	LD	A,	FFH		FF	00FF	FFF	8000	00	FFFF	7FFF	00FF	00000010	0008
0008	23	INC	HL			FF	00FF	FFF	8001	00	FFFF	7FFF	00FF	00000010	0009
0009	BD	CP	L			FF	00FF	FFF	8001	00	FFFF	7FFF	00FF	10000010	000A
000A	C2	JP	NZ,	0005H		FF	00FF	FFF	8001	00	FFFF	7FFF	00FF	10000010	0005
0005	70	LD	[HL],	B		FF	00FF	FFF	8001	00	FFFF	7FFF	00FF	10000010	0006
0006	3E	LD	A,	FFH		FF	00FF	FFF	8001	00	FFFF	7FFF	00FF	10000010	0008
0008	23	INC	HL			FF	00FF	FFF	8002	00	FFFF	7FFF	00FF	10000010	0009
0009	BD	CP	L			FF	00FF	FFF	8002	00	FFFF	7FFF	00FF	10000010	000A
000A	C2	JP	NZ,	0005H		FF	00FF	FFF	8002	00	FFFF	7FFF	00FF	10000010	0005
0005	70	LD	[HL],	B		FF	00FF	FFF	8002	00	FFFF	7FFF	00FF	10000010	0006
0006	3E	LD	A,	FFH		FF	00FF	FFF	8002	00	FFFF	7FFF	00FF	10000010	0008
0008	23	INC	HL			FF	00FF	FFF	8003	00	FFFF	7FFF	00FF	10000010	0009
0009	BD	CP	L			FF	00FF	FFF	8003	00	FFFF	7FFF	00FF	10000010	000A

Рис. 8.10. Вывод на дисплей содержимого регистров процессора.

Возможности использования команды вывода на индикацию при отладке программ иллюстрируются рис. 8.10, где показано, как на каждом шаге выполнения программы на дисплей выводится содержимое всех внутренних регистров процессора.

Трассировка

Рассматриваемая система проектирования включает множество средств анализа, начиная с чисто программного логического анализатора, который следит за информацией на шинах адресов, данных и управления эмулятора, вплоть до вставляемых дополнительно плат, с помощью которых обеспечивается возможность анализа состояний и временных соотношений в реальном времени.

Добавление платы анализа придает аппаратной станции свойства анализатора состояний в реальном времени, с помощью которого возможна 256-кратная регистрация информации на 8 линиях шины данных, 16 линиях адресной шины, 8 линиях сигналов состояния, или управления (рис. 8.11). Как правило, для заполнения буфера трассы используется один из трех следующих методов (рис. 8.12):

1. «Трассировка около» запускающего события, при которой в буфер заносятся 128 событий, предшествующих запускающему, и 128 событий после него.
2. «Трассировка до» запускающего события.
3. «Трассировка после» запускающего события.

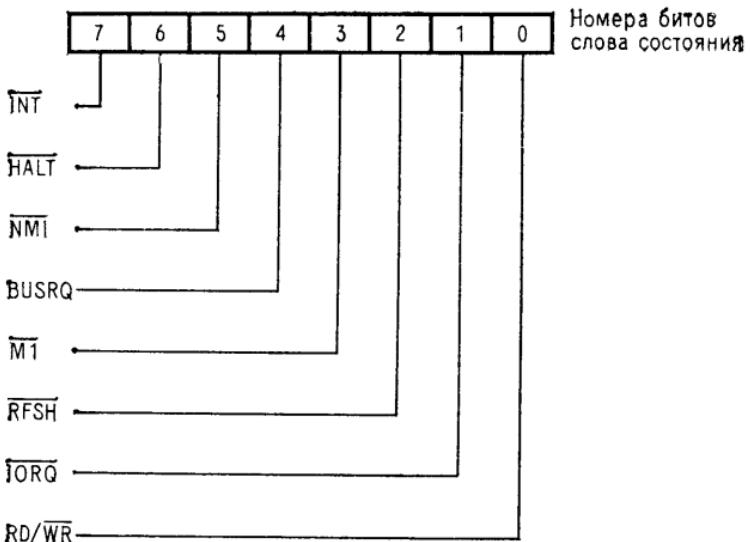
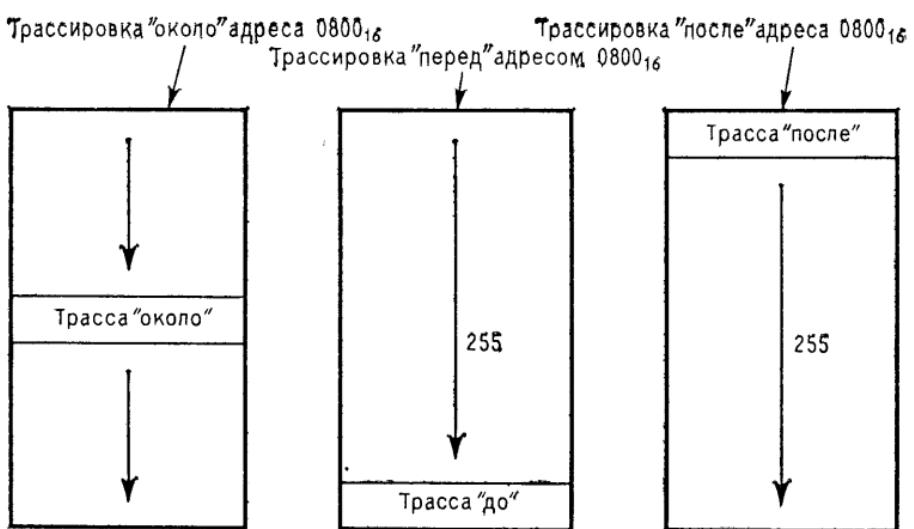


Рис. 8.11. Байт состояния эмулятора (состояние определенных управляемых линий процессора Z80 сообщается анализатором как байт состояния). Байт состояния может быть использован в качестве условия для спецификации трассы. Для отдельных линий состояния могут быть применены следующие мономонические обозначения: чтение, запись, код операции, чтение пам., запись пам., ввод, вывод.



Трасса "только": трассировка начинается, как и при использовании принципа "после". В буфер трассы помещается информация только о событиях, удовлетворяющих заданному условию

Рис. 8.12. Различные принципы запуска, используемые в анализаторе.

Трассировка после адреса = 8000₁₆

ТРАССА

АДРЕС, ДАННЫЕ, СОСТОЯНИЕ

ПОСЛЕ	8000H	E2H	3FH	
+001	0006H	LD	A, FFH	
+002	0008H	INC	HL	
+003	0009H	CP	L	
+004	000AH	JP	NZ, 0005H	
+005	0005H	LD	[HL], B	[HL] E2H
+006	0006H	LD	A, FFH	
+007	0008H	INC	HL	
+008	0009H	CP	L	
+009	000AH	JP	NZ, 0005H	
+010	0005H	LD	[HL], B	[HL] E2H
+011	0006H	LD	A, FFH	
+012	0008H	INC	HL	
+013	0009H	CP	L	
+014	000AH	JP	NZ, 0005H	
+015	0005H	LD	[HL], B	[HL] E2H

Рис. 8.13. Пример трассировки.

Трассировка только адресов 8000₁₆ и 8100₁₆

ТРАССА
АДРЕС, ДАННЫЕ, СОСТОЯНИЕ

ПОСЛЕ	0007H	FFH	3EH	
+001	8000H	42H	3FH	
+002	8100H	42H	3FH	
+003	8000H	43H	3FH	
+004	8100H	43H	3FH	
+005	8000H	44H	3FH	
+006	8100H	44H	3FH	
+007	8000H	45H	3FH	
+008	8100H	45H	3FH	
+009	8000H	46H	3FH	
+010	8100H	46H	3FH	
+011	8000H	47H	3FH	
+012	8100H	47H	3FH	
+013	8000H	48H	3FH	
+014	8100H	48H	3FH	
+015	8000H	49H	3FH	

Рис. 8.14. Пример трассировки.

Трассировка последовательности данных ОАА₁₆
с запуском после адреса 8000₁₆

АДРЕС, ДАННЫЕ, СОСТОЯНИЕ

ПОСЛЕ	8000H	AAH	3FH	
+001	8000H	ABH	3FH	
+002	0006H	LD	A, FFH	
+003	0008H	INC	HL	
+004	0009H	CP	L	
+005	000AH	JP	NZ, 0005H	
+006	0005H	LD	[HL], B	[HL] ABH
+007	0006H	LD	A, FFH	
+008	0008H	INC	HL	
+009	0009H	CP	L	
+010	000AH	JP	NZ, 0005H	
+011	0005H	LD	[HL], B	[HL] ABH
+012	0006H	LD	A, FFH	
+013	0008H	INC	HL	
+014	0009H	CP	L	
+015	000AH	JP	NZ, 0005H	

Рис. 8.15. Пример трассировки.

Четвертый вариант заключается в «трассировке только» конкретных событий, когда в буфере трассы регистрируется лишь информация, соответствующая условию запуска. В этом случае буфер по существу начинает заполняться так же, как и при использовании метода «трассировка после».

Задание запускающего события, или, более точно, спецификация трассы, позволяет оператору регистрировать определенную последовательность событий, среди которых предположительно имеются причины некорректного функционирования системы. Рисунки 8.13—8.15 дают представление о разнообразии спецификаций трассы в диапазоне от простой «трассировки после адреса 8000_{16} » до относительно сложной «трассировки последовательности данных = $0AA_{16}$ с запуском после адреса = = 800_{16} ».

Внутрисхемная эмуляция как средство поиска неисправностей

Системы проектирования микропроцессорных систем представляют собой весьма мощные средства отыскания неисправностей, однако круг лиц из числа обслуживающего персонала, имеющих доступ к такому оборудованию, ограничен. Существуют и другие факторы, препятствующие превращению СПМС в средство обслуживания, которое можно было бы применять без ограничений: физические размеры СПМС, необходимость наличия квалифицированных операторов для работы с ними, значительная стоимость. В то же время некоторые фирмы-изготовители, такие, например, как Applied Microsystems, Fluke, Hewlett Packard, Millenium и Solartron, продают отдельно эмуляторы, многие из которых в большей степени подходят для работы рядового специалиста по обслуживанию микропроцессорных систем.

С точки зрения инженера, занимающегося обслуживанием, внутрисхемная эмуляция представляет собой метод принятия управления тестируемым устройством и подачи тестовых, или стимулирующих, запрограммированных воздействий на его шины. Это имеет важное значение в ситуациях, когда тестируемое устройство не реагирует на обычные воздействия с клавиатуры. Взяв на себя управление, эмулятор может осуществлять прогон программ, предназначенных для проверки функционирования компонентов целевой системы и аналогичных тем программам, которые используются для самоконтроля. При обнаружении фактов некорректной работы специалист по обслуживанию может получить с помощью средств индикации эмулятора информацию о возможных источниках неисправностей. Примером эмулятора такого типа является устройство поиска неисправностей фирмы Fluke, описанное в гл. 10. Не все, однако, эмуляторы снабжены

клавиатурой и индикаторами. Так, эмулятор Solartron Micropad предназначен для использования совместно с тестовыми программами, разработанными на другой машине и затем загруженными в СППЗУ эмулятора. Эти тестовые программы управляют работой целевой системы, генерируя потоки данных, которые отслеживаются с помощью сигнатурного анализатора.

В последнее время на рынке появились конструктивно автономно оформленные эмуляторы, располагающие многими функциональными возможностями, характерными ранее лишь для СПМС, при значительно меньшей стоимости. Примером такого рода устройств является микровнутрисхемный эмулятор (МВСЭ) фирмы Micotek. Прилагаемый набор специализированных плат позволяет применять его для эмуляции большинства промышленно выпускаемых микропроцессоров, как 8-, так и 16-разрядных. Отдельные модели эмулятора различаются емкостью эмулирующих ЗУПВ и реализуемыми видами трассировки, среди которых наиболее информативной является трассировка в реальном времени. Управление эмуляторами осуществляется через интерфейс RS232 с использованием либо дисплея, либо вычислительной системы, снабженной соответствующим портом.

Управление изолированным эмулятором от дисплея дает удовлетворительные результаты лишь при использовании коротких тестовых программ (предназначенных, например, для тестирования ЗУПВ или формирования сигнатур ПЗУ с использованием заложенных в них программ). В число функциональных средств эмулятора входят строковый ассемблер, который удобно применять для небольших программ и для осуществления вставок в более крупные программные единицы, и двухпроходный обратный ассемблер, который генерирует метки для всех подпрограмм и команд перехода (неплохой штрих, характерный далеко не для каждой из дорогих систем).

В более серьезных ситуациях, связанных с обслуживанием микропроцессорных систем, применяется главная вычислительная система, что позволяет загружать в эмулирующую память тестовые программы, генерируемые с помощью ассемблеров или компиляторов в форматах Intel или Textronix. При использовании главной системы для обмена информацией с эмулятором необходима программа-драйвер. Ряд таких программ поставляется разработчиками для некоторых наиболее популярных систем, в том числе для микроЭВМ типов CP/M и Apple, а также для мини-ЭВМ класса DEC.

Возможности трассировки при использовании главной системы достаточно широки и позволяют оператору регистрировать информацию в реальном масштабе времени о данных, адресах и состоянии процессора в объемном буфере на 2048 слов. Однако возможности формулирования условий запуска относи-

тельно просты и ограничиваются главным образом заданием запускающих событий в виде адресов и сигналов состояния.

Эти примеры иллюстрируют два различных подхода к эмуляции: на основе систем проектирования, располагающих внутрисхемной эмуляцией как одной из своих функциональных возможностей, и с использованием автономных эмуляторов, управляемых от микроЭВМ, позволяющих достичь экономии средств. Для некоторых случаев применение автономных устройств может оказаться более целесообразным с практической точки зрения, так как тестовые программы в них зафиксированы, т. е. занесены в ПЗУ.

Заключение

Микропроцессорную систему можно заставить выполнять программу под управлением внешней, главной системы, которая берет на себя функции ЦПУ целевой системы. Поведение шин, памяти и схем ввода-вывода целевой системы может отслеживаться главной системой, даже если целевая система не имеет устройств ввода-вывода или имеет, но они неисправны. Внутрисхемная эмуляция может быть реализована с помощью СПМС или автономных эмуляторов, причем большинство систем располагает возможностью трассировки в реальном масштабе времени.

Глава 9

Сигнатурный анализ

Традиционный подход к поиску неисправностей в аналоговых системах заключается в разработке аннотированной схемы, на которой явным образом указываются уровни напряжения и форма сигналов, которые должны иметь место в различных точках устройства (рис. 9.1). Сравнивая показания вольтметра и изображения на экране осциллографа с приведенными на схеме,

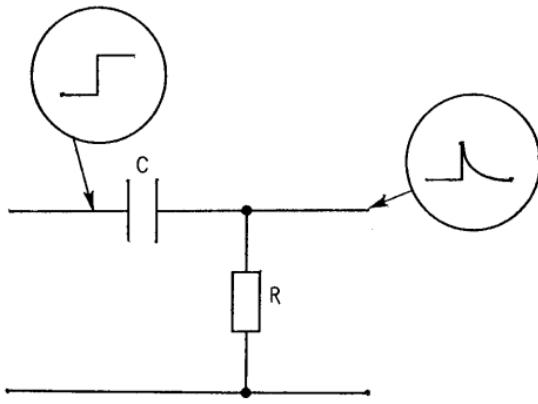


Рис. 9.1. Традиционная аннотированная схема, в точках которой показаны ожидаемые формы сигналов.

специалист по обслуживанию может искать неисправности и восстанавливать работоспособность системы, не будучи знаком детально с ее работой. Использование этого подхода для цифровых устройств осложняется рядом обстоятельств. Ни вольтметры, ни осциллографы не обеспечивают получения достаточно полезной информации: уровни напряжения лежат около 0 или 5 В, сигналы имеют большую длительность, сложную форму и появляются в произвольные моменты времени, что делает все потоки данных весьма схожими по внешнему виду.

Очевидно, необходим какой-то прибор, который может регистрировать эти сложные потоки данных и в то же время представлять информацию о них в распоряжение оператора в компактной форме. Этую функцию идеально выполняет сигнатурный

Сигнатурный анализатор

Назначение сигнатурного анализатора состоит в восприятии логических уровней в какой-то точке схемы и выработке кода (сигнатуры), характеризующего процесс, происходящий в этой точке. На рис. 9.3 показан типичный вид лицевой панели прибора. Анализатор подключается к исследуемой схеме в пяти

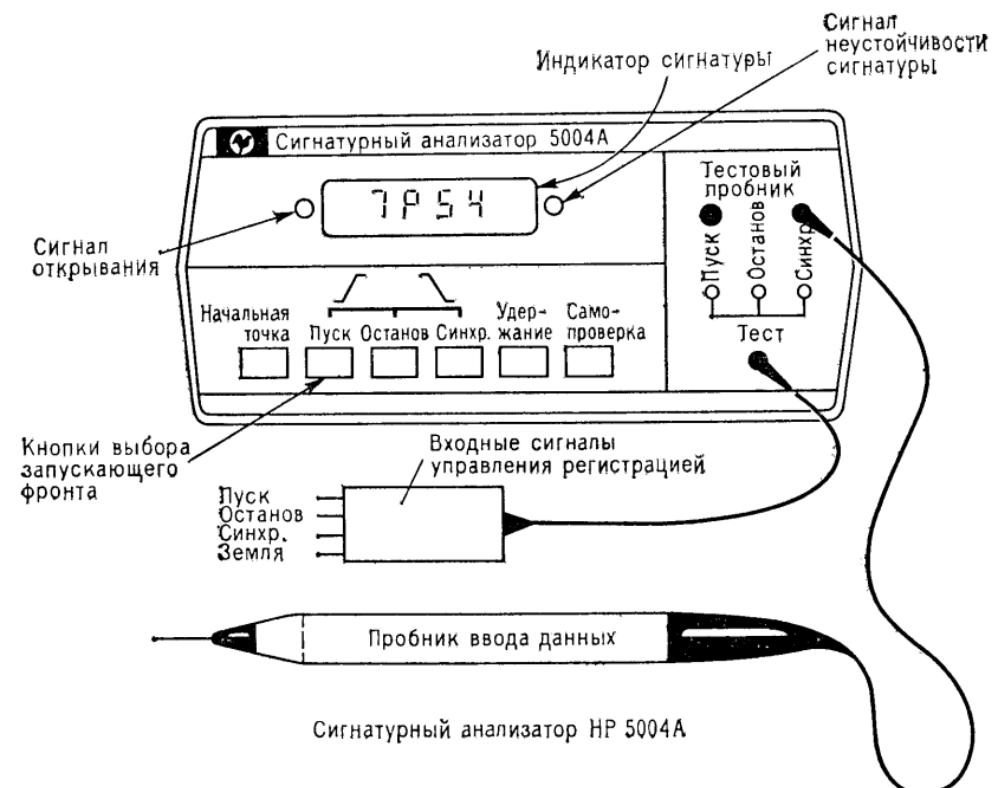


Рис. 9.3. Сигнатурный анализатор и его соединения с тестируемым устройством.

точках. Непосредственно через пробник поступают в анализатор данные от выбранной для тестирования точки. Остальные четыре соединения: пуск, останов, синхр. и «земля» — осуществляются через внешнюю переходную приставку. Как и при работе с логическим анализатором, тактовые импульсы используются для стробирования данных. Как правило, допустимыми являются частоты от 0 Гц до 10 Мгц, а с помощью переключателей на лицевой панели оператор может задать положительный или отрицательный фронт в качестве активного перепада сигнала. Сигналы пуска и останова определяют длину регистрируемого потока данных, или «сигнатурного окна».

анализатор. Как и осциллограф, он воспринимает поведение логического сигнала в некоторой точке схемы, но вместо воспроизведения формы изменений сигнала он вырабатывает четырехразрядный код (или сигнатуру), несущий информацию о потоке данных. Сама сигнатура физического смысла не имеет и служит лишь символом, представляющим набор логических 1 и 0, из которых состоит поток данных. Если изменяется хотя бы один бит набора, то меняется и сигнатура.

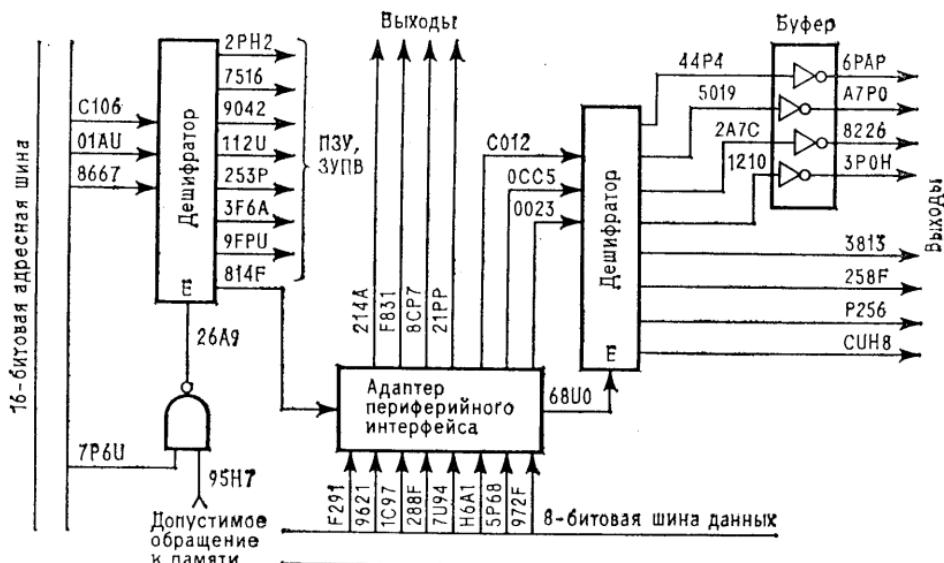


Рис. 9.2. Указание сигнатур на аннотированной цифровой схеме.

При использовании сигнатурного анализа аннотированная схема цифрового устройства может выглядеть как показано на рис. 9.2. Вместо значений напряжения и осциллограмм узлы схемы снабжаются указаниями сигнатур, описывающими потоки данных в этих узлах.

В микропроцессорных системах последовательность изменений сигнала в некоторой точке схемы, а значит, и сигнатура зависят от выполняемой в системе программы. Неотъемлемой принадлежностью сигнатурного анализа является стимулирующая, или тестовая, программа, под воздействием которой приводятся в действие компоненты и узлы системы и тем самым генерируются потоки данных. От узла схемы, который пребывает в неизменном состоянии, можно получить лишь ограниченную информацию. Стимулирующая программа должна обеспечивать переключение значений сигналов в узлах схемы между логическим 0 и логической 1 с таким расчетом, чтобы в них имели место потоки данных, отображающие функционирование системы.

Основным блоком прибора является 16-разрядный сдвиговый регистр с обратными связями (рис. 9.4). При получении достоверного сигнала запуска регистр устанавливается в исходное нулевое состояние, после чего данные вводятся в него под управлением заданного фронта синхросигналов. Благодаря цепям обратной связи данные, вводимые в регистр, подвергаются кодированию и обеспечивается учет всех битов потока данных, а не только шестнадцати последних в формировании окончательной

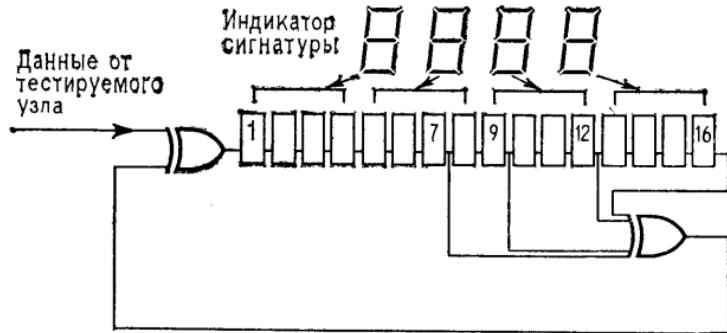


Рис. 9.4. 16-разрядный регистр с обратными связями, входящий в состав сингнатурного анализатора.

сингнатурой. При поступлении сигнала останова 16 бит, остающиеся в регистре, индицируются в шестнадцатеричном формате, образуя сингнтуру наблюдаемого потока данных.

Для индикации сингнтур используется нестандартный набор шестнадцатеричных символов: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, C, F, H, P, U. По сравнению со стандартным кодом этот набор имеет два важных преимущества:

1. Каждый символ может быть легко воспроизведен на семи-сегментном светодиодном индикаторе, причем не возникает никакой путаницы между цифрой 6 и строчной буквой b, а также между цифрой 8 и прописной буквой B, как это бывает при использовании стандартного набора.

2. У оператора не возникает соблазна спутать четырехразрядные шестнадцатеричные сингнтуры с системными адресами или проследить предысторию появления той или иной сингнтуры в попытке получить дополнительную информацию.

Имеющиеся на лицевой панели индикаторы «Открыто» и «Неустойчивая сингнтура» несут информацию о сингнтуах, используемых анализатором. Индикатор, показывающий, когда открывается и закрывается сингнтурное окно, мерцаает при повторяющихся открываниях и закрываниях. Индикатор неустойчивости сингнтуры светится, когда различаются сингнтуры, формируемые в следующих друг за другом окнах.

Сигнатурное окно

Сигнатурным окном управляют три входных сигнала: «Пуск», «Останов» и «Синхр.». В реальных условиях эти сигналы могут вырабатываться исследуемой схемой. Для повышения гибкости

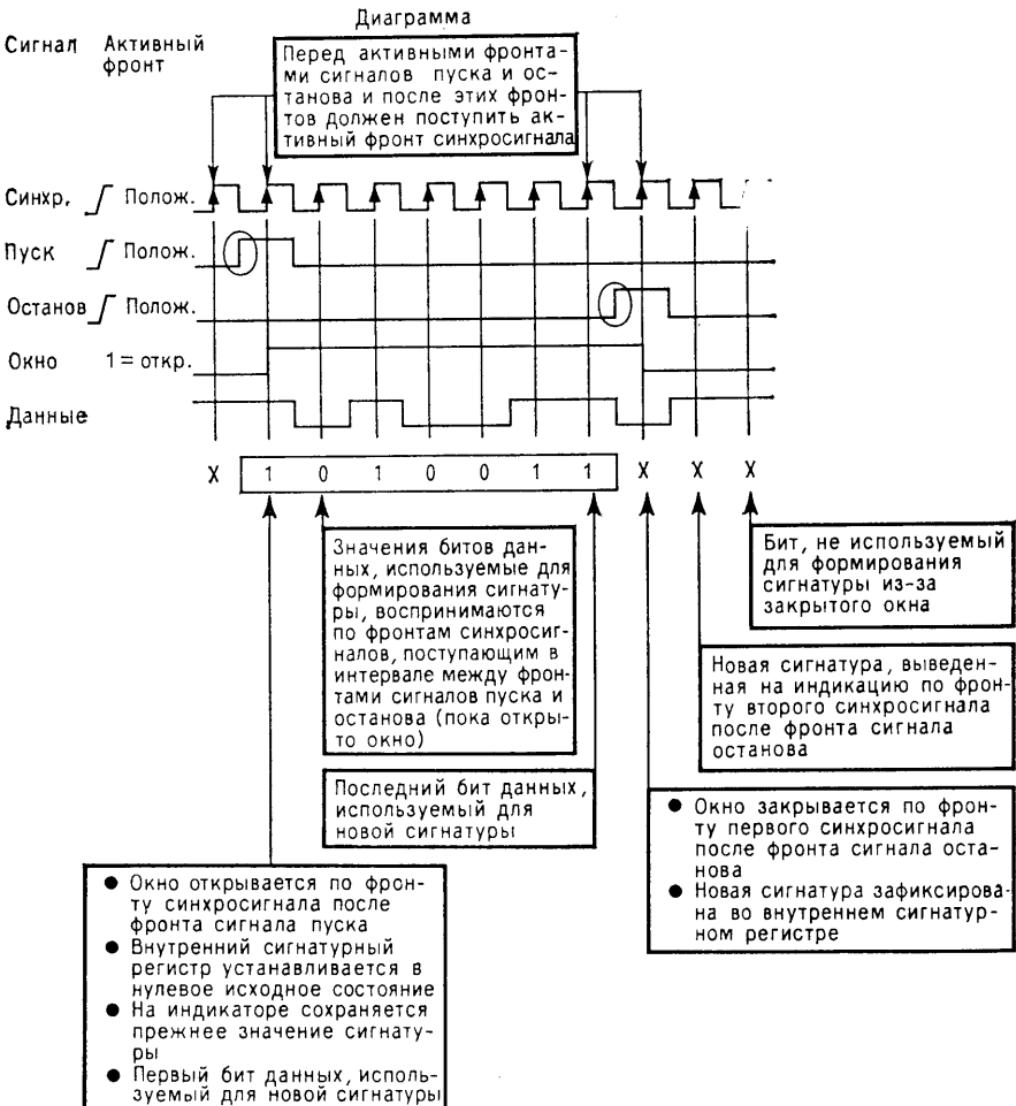


Рис. 9.5. Формирование сигнатурного окна.

управления окном предусмотрена возможность выбора активных, или запускающих, фронтов каждого из этих сигналов с помощью органов управления, имеющихся на лицевой панели анализатора. Обычно тот или иной фронт выбирается из соображе-

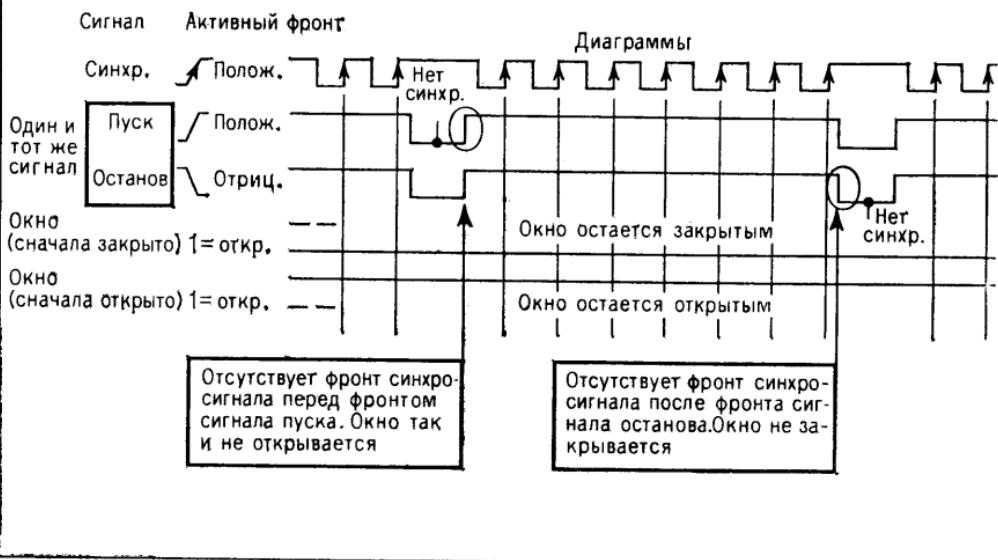
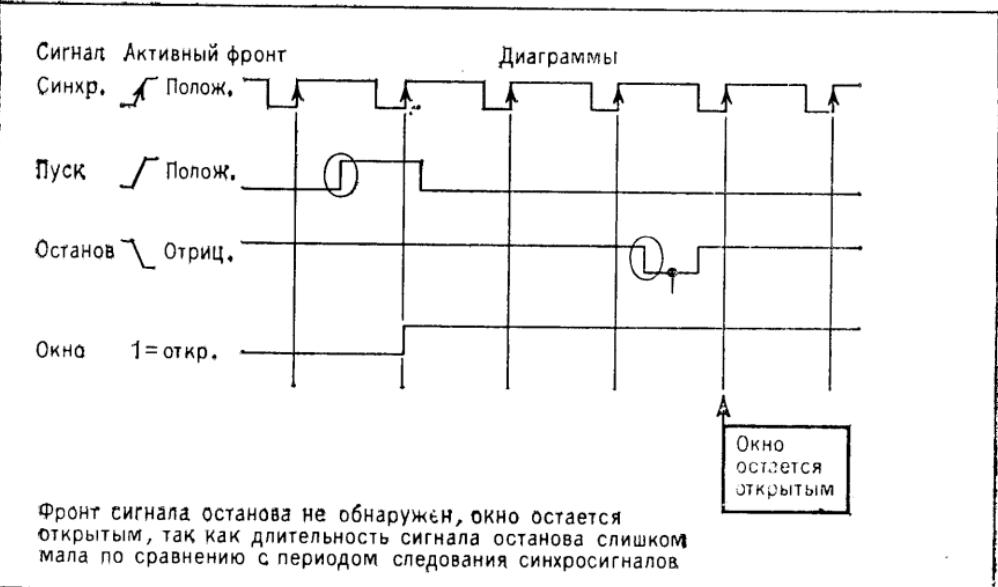
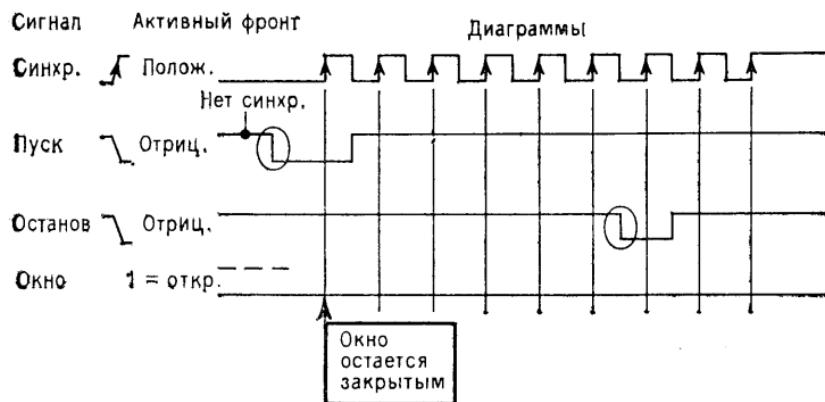
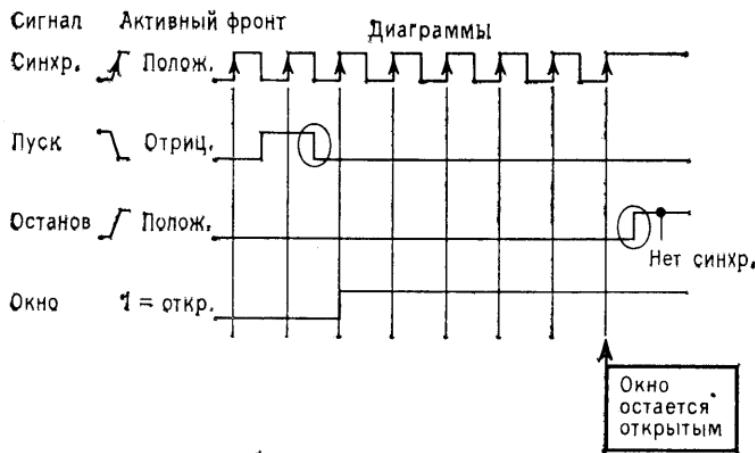
**α****б**

Рис. 9.6. Проблемы, которые могут возникнуть при неправильном формировании сигнатурных окон: *а* — отсутствие синхросигналов до или после фронтов сигналов пуска и останова; *б* — сигналы пуска или останова слишком коротки по сравнению с периодом следования синхросигналов.



Фронт сигнала пуска не обнаружен, окно не открывается, так как синхросигналы появляются после активного фронта сигнала пуска

a



Фронт сигнала останова не обнаружен, окно остается открытым, так как синхросигналы прекращаются до появления активного фронта сигнала останова

б

Рис. 9.7. Проблемы формирования сигнатурного окна: а — синхросигналы появляются лишь после фронта сигнала пуска; б — синхросигналы прекращаются до поступления фронта сигнала останова.

ний задания таких моментов времени, когда данные в исследуемой точке схемы имеют установленные значения. Значения логического сигнала в точке воспринимаются анализатором только при поступлении определенного фронта синхросигнала, а в другие моменты времени во внимание не принимаются. Для задания сигнатурного окна синхросигналы используются совместно с сигналами пуска и останова. Как показано на рис. 9.5, для того чтобы анализатор реагировал на поступление сигналов пуска и останова, необходимо, чтобы перед активным фронтом каждого из этих сигналов и после этого фронта поступал активный фронт синхросигнала.

Обратите внимание, что окно открывается не по фронту сигнала «Пуск», а по фронту синхросигнала, поступающего после фронта пускового сигнала. Аналогичным образом закрывание окна происходит по фронту синхросигнала, поступающего после сигнала «Останов». Первый элемент данных, которые должны быть занесены в сдвиговый регистр, поступает в него по фронту первого синхросигнала, следующего за фронтом сигнала «Пуск». Последний бит данных фиксируется в анализаторе фронтом синхросигнала, предшествующего фронту сигнала «Останов».

Для обеспечения должного открывания и закрывания сигнатурного окна необходимо внимательно выбирать сигналы управления окном. Временные диаграммы, приведенные на рис. 9.6 и 9.7, иллюстрируют проблемы, которые могут возникнуть при неправильном выборе. Одно из совершенствований системы запуска состоит в предоставлении возможности реализации двух режимов запуска с целью облегчения решения проблемы формирования сигнатурного окна. При обычном режиме необходимо поступление фронтов синхросигнала и до, и после изменения значений сигналов пуска и останова. Если, однако, сигналов пуска и останова достаточной длительности в распоряжении нет, возможен другой режим, допускающий запуск сигналами, длительность которых меньше половины периода следования синхросигналов. Хотя этот вариант менее предпочтителен (случайные импульсные сигналы могут вызвать ложное открывание и закрывание окна, что в свою очередь приводит к появлению неустойчивых сигнатур), он может оказаться единственным методом, посредством которого могут быть получены сигнатуры тестируемой системы.

Сигнатуры «земли» и напряжения питания

Несколько более внимательное рассмотрение того, каким образом обрабатываются два простых потока данных, должно помочь проиллюстрировать функционирование сдвигового ре-

стра с обратными связями. Сначала рассмотрим поток данных, имеющий постоянный потенциал земли. Активным фронтом каждого синхросигнала в анализатор вводится логический 0. Так как по цепям обратной связи на входы вентиля ИСКЛЮЧАЮЩЕЕ ИЛИ также подаются логические 0, регистр продолжает оставаться заполненным логическими нулями. Следовательно, сигнатура узла схемы, имеющего потенциал земли, всегда представляет собой 0000. Если же с помощью пробника опрашивается узел, сигнал в котором имеет постоянное значение логической 1, то содержимое сдвигового регистра будет меняться по мере того, как алгоритм функционирования обратных связей изменяет поток битов, поступающих в регистр. Окончательная сигнатура, очевидно, зависит только от количества синхросигналов, умещающегося в пределах сигнатурного окна. Поэтому сигнатура напряжения питания часто используется для предварительной оценки правильности выбора окна; она позволяет перед тем, как начать измерения более сложных потоков данных, убедиться, что между сигналами пуска и останова поступает нужное число синхросигналов.

Свободный прогон как средство задания стимулирующих воздействий

Свободный прогон микропроцессора представляет собой простой способ проверки работоспособности микропроцессорной системы (см. гл. 4). В рассматриваемом ниже примере микропроцессор Z80 принудительно подвергается свободному прогону за счет «монтажной» подачи кода НЕТ ОПЕРАЦИИ (NOP) на контакты его шины данных. Каким образом это делается, показано на рис. 4.1. Микропроцессор удаляется из установочного гнезда, и между ним и гнездом устанавливается колодка свободного прогона, внутренним монтажом которой обеспечивается подача на Z80 команды NOP. Переходная колодка выполняет следующие функции:

- отсоединяет микропроцессор от системной шины данных, предотвращая тем самым выполнение микропроцессором программы системы;
- подает на шину данных микропроцессора при считывании команды по любому адресу команду NOP, вынуждая процессор выполнить цикл перебора всего диапазона адресов;
- оставляет подключенными контакты адреса и управляющих сигналов, чтобы проверялось функционирование адресной шины и схем дешифрирования адреса, имеющихся на плате.

После начальной установки процессор считывает код NOP, после чего выполняет инкремент счетчика команд (т. е. инфор-

мации, подаваемой на адресную шину), чтобы попытаться осуществить чтение содержимого очередной области памяти. Так как в результате чтения по каждому адресу в микропроцессор поступает команда NOP, он последовательно перебирает весь диапазон адресов памяти; при этом проверяется функционирование схем дешифраторов адресов и поступление данных от кристаллов ПЗУ и ЗУПВ на шину данных. Формируемые при этом потоки данных можно проконтролировать путем сигнатурного анализа.

Пример: плата контроллера на базе микропроцессора Z80

Данный пример иллюстрирует применение модуля свободного прогона для тестирования платы контроллера, построенного на микропроцессоре Z80 (см. приложение 2), методами сигнатурного анализа; описываются сигналы, используемые для управления сигнатурным анализатором. В примере рассматривается подготовка документации, впоследствии применяемой при тестировании.

Сигнатуры адресной шины и дешифратора адресов

С линии A15 адресной шины подаются сигналы пуска и останова для сигнатурного окна. На рис. 9.8 показано изменение значения этого сигнала по мере перебора процессором своего

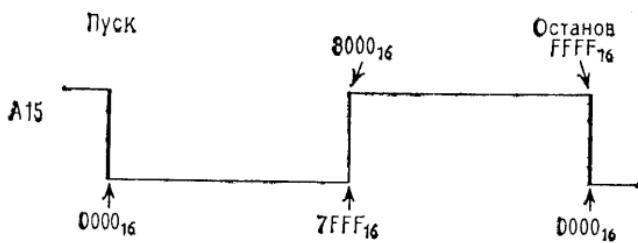


Рис. 9.8. Сигнал адресной линии A15 обеспечивает формирование сигнатурного окна для получения сигнатур свободного прогона адресов.

диапазона адресов (действия, связанные с регенерацией, в данном случае не учитываются). При считывании данных по адресу 8000_{16} сигнал A15 приобретает высокий уровень и сохраняет его, пока адрес не достигнет величины $FFFF_{16}$, после чего уровень A15 опять становится низким и начинается новый перебор адресов. Если выбрать отрицательные фронты сигнала A15 в качестве сигналов пуска и останова, тем самым будет задано

сигнатурное окно, в течение которого процессор последовательно выводит на шину все возможные адреса.

В микропроцессоре Z80 отрицательный фронт сигнала \overline{RD} указывает момент, когда выведенный на шину адрес приобретает установленное значение. Если использовать его в качестве синхросигнала анализатора, в анализатор будут поступать только достоверные значения адреса. (Следует отметить, что

Сигнатурная таблица MEDC

Заголовок: свободный прогон платы Z80

Функция	Пуск	Останов	Синхр.	Режим	Сигнатура питания	Сигнал пуска A15	Сигнал останова A15	Сигнал: RD	Синхро-
:	:	:	:	:	0001	процессора	процессора	процессора	процессора
:	:	:	:	:	Z 80	Z 80	Z 80	Z 80	Z 80
Квалифицированный синхросигнал: отсутствует									
Точка схемы	Контакт ИС	Контакт ИС	Контакт ИС	Контакт ИС	Сигнатура				
•	•	•	•	•	•	•	•	•	2H70
A12	Z1-2	Z2-18							HPP0
A13	Z1-3								1293
A14	Z1-4								HAP7
A15	Z1-5	Z9-13							3C96
									3827
									755U

Рис. 9.9. Сигнатуры свободного прогона платы контроллера на базе микропроцессора Z80.

в ходе операций регенерации линии \overline{RD} не активизируется и сигнатуры, вырабатываемые с использованием такого синхросигнала, абсолютно не учитывают работу адресной шины в процессе регенерации.)

Получаемые при этом сигнатуры адресной шины и дешифратора приведены на рис. 9.9. Наряду с аннотированными схемами для документирования сигнатур находит применение табличный способ их представления, иллюстрируемый этим рисунком. В таблице описываются стимулирующие воздействия, используемые для получения сигнатур, сигналы, применяемые для синхронизации, пуска и останова, а также конфигурация анализатора. В начале таблицы приведена сигнтура напряжения питания (сигнтура сигнала «земли» всегда имеет вид 0000). Для каждой точки схемы указаны номера контактов ИС, соединенные с этой точкой, и характеризующая их сигнтура.

Сигнатуры ПЗУ

При свободном прогоне микропроцессор пытается осуществить чтение по каждому адресу, выводя информацию из системной памяти на шину данных и генерируя потоки данных, характеризующих содержимое памяти. Разрыв шины данных, образующийся благодаря колодке свободного прогона, предотвращает поступление этой информации в процессор; вместо информации с системной шины данных в процессор подается команда NOP. Для того чтобы формировать сигнатуры информации, снимаемой с шины данных, необходимо выполнить два изменения. Во-первых, для того чтобы перейти от приема адресов к приему данных, необходимо организовать подачу синхронного сигнала в несколько более поздний момент временного цикла

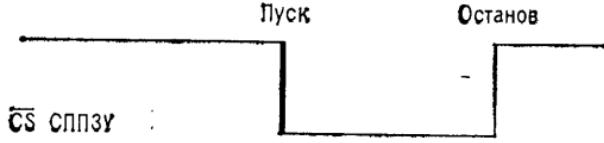


Рис. 9.10. Сигнал выбора кристалла СППЗУ формирует сигнатурное окно для получения сигнатур свободного прогона ПЗУ.

Z80, а именно в момент прохождения положительного фронта сигнала \overline{RD} . Во-вторых, если допустить, что сигнатурное окно охватывает данные, считываемые из всей памяти, имеющей объем 64К, то сигнатуры будут формироваться из содержимого как ПЗУ, так и ЗУПВ. Так как не существует простых приемов, обеспечивающих постоянство содержимого ЗУПВ при каждом включении питания системы, такие сигнатуры окажутся неповторяющимися и, следовательно, лишенными всякого смысла. Осмысленные сигнатуры можно получать, если заново определять длину сигнатурного окна с таким расчетом, чтобы оно охватывало только потоки данных, получаемые от ПЗУ системы. На рис. 9.10 показано, как это сделать, соединив входы пуска и останова с выходным сигналом выбора кристалла, определяющим обращение к ПЗУ (IC2). Необходимо учесть при этом, что изменение сигнатурного окна по сравнению с рассмотренным выше тестом адресной шины приведет к изменению значения сигнатуры, формируемой сигналом напряжения питания.

Программы стимулирующих воздействий .

Эффективное тестирование таких компонентов системы, как ЗУПВ и устройства ввода-вывода, может быть осуществлено только при управлении ими с помощью программ формирования стимулирующих воздействий. При анализе с программным

заданием стимулирующих воздействий используются программы, хранящиеся в ПЗУ, посредством которых на шину данных подаются повторяющиеся потоки данных и генерируются сигналы пуска/останова, определяющие формат сигнатурного окна.

Тестеры и эмуляторы

Наиболее рациональным периодом времени для формирования средств сигнатурного анализа является этап проектирования микропроцессорной системы. На этой стадии инженер-разработчик имеет возможность уделить внимание созданию программ генерации стимулирующих воздействий и ввести в состав аппаратуры дополнительное оборудование, которое может понадобиться для сигнатурного анализа, например дополнительную панель для установки ПЗУ, фиксаторы для выработки сигналов пуска и останова, переключатели и перемычки для введения схем ПЗУ в число функциональных узлов системы и для разрыва цепей обратной связи. В некоторых системах (в особенности это характерно для систем, выпускаемых производителями сигнатурных анализаторов) предусматриваются встроенные средства, облегчающие реализацию сигнатурного анализа. К сожалению, в большинстве систем подобные средства отсутствуют. Нежелание закладывать в систему предпосылки для упрощения последующего применения сигнатурного анализа имеет, по-видимому, множество причин, таких, например, как

- увеличение продолжительности времени проектирования системы;
- необходимость дополнительных затрат на аппаратные средства;
- незнание преимуществ сигнатурного анализа.

Каковы бы ни были эти причины, непредусмотрительность разработчиков, не заложивших в систему на этапе проектирования необходимые аппаратные и программные средства, не исключает возможности применения сигнатурного анализа. Существующие изделия могут быть доработаны с целью его реализации. Для облегчения этой задачи некоторые фирмы разработали тестеры и эмуляторы, с помощью которых программы генерации тестовых воздействий вводятся в проверяемую систему. Внешний вид одного из таких приборов, носящего название Solartron Micropolis, показан на рис. 9.11.

При использовании тестера или эмулятора микропроцессор изымается из тестируемого устройства и прибор подключается к этому устройству через процессорное гнездо. Сам процессор устанавливается в гнездо с нулевым фиксирующим усилием,

имеющееся на эмуляторе, где он может быть протестирован, а затем использован для прогона программы генерации тестовых воздействий для проверяемого устройства.

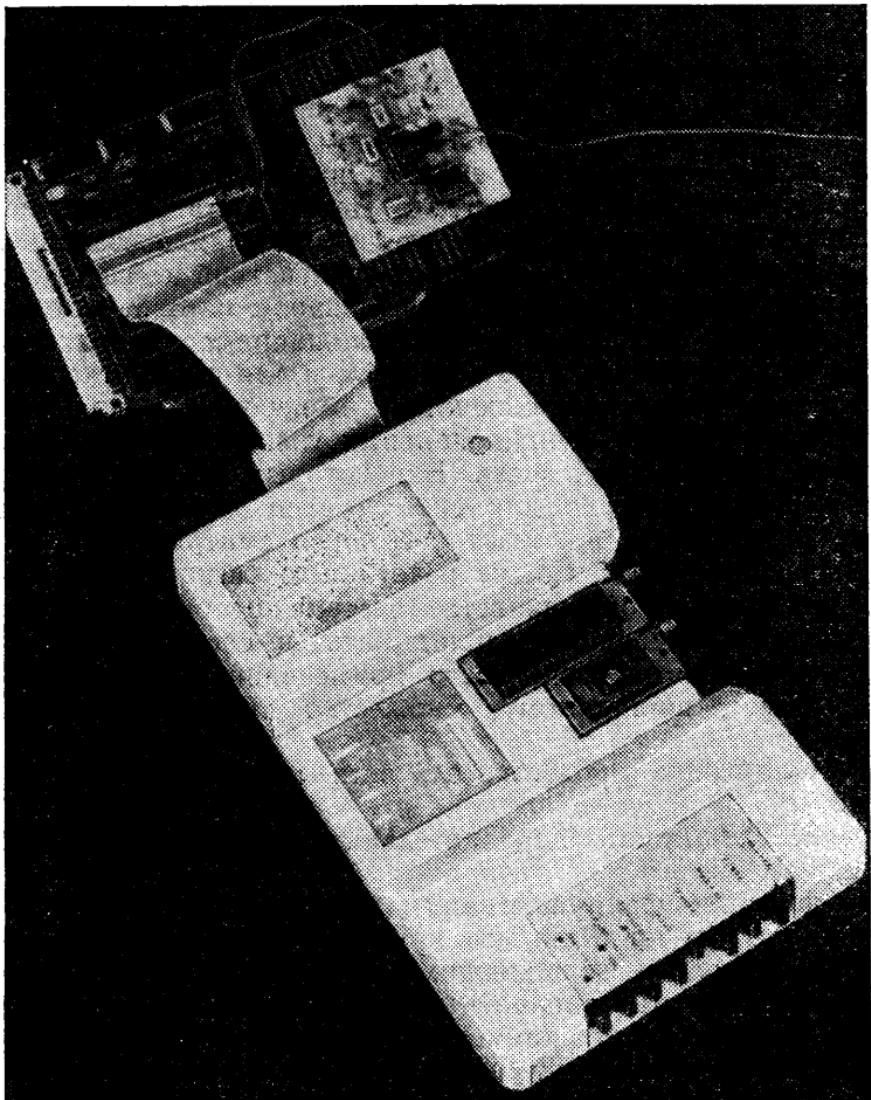


Рис. 9.11. Типичный внешний вид эмуляционной переходной приставки.

Наряду с эмуляцией как тестер, так и эмулятор Micropod позволяют осуществлять такие полезные операции, как свободный прогон, тестирование микропроцессора и т. д. В следующем разделе кратко описаны принципы организации и применения эмуляционной приставки Micropod.

Мікропод

Для работы с различными микропроцессорами используются специализированные модели приставки. На одной из сторон приставки имеется ряд кнопочных переключателей, с помощью которых оператор может задать один из следующих видов работы.

(A) *Свободный прогон* (Free Run). Приставка содержит необходимую аппаратуру для осуществления свободного прогона микропроцессора, вырабатывающую стимулирующие воздействия для контроля шины данных, адресной шины и дешифрователей тестируемого устройства.

(Б) *Тестирование микропроцессора* (Microprocessor Test). Имеющаяся в ПЗУ программа генерации тестовых воздействий осуществляет проверку самого микропроцессора. В программе используется большинство команд, входящих в набор команд микропроцессора; она вырабатывает характеристические сигнатуры для каждого внешнего вывода микропроцессора.

(В) *Нормальная работа* (Normal). При задании этого режима тестируемое устройство после начальной установки выполняет свою собственную мониторную программу.

(Г) *Тестовое воздействие* (Stimulus). Нажатием этой кнопки задается выполнение эмуляции. Имеется второе гнездо с нулевым фиксирующим усилием для установки СППЗУ, содержащего тестовые программы.

Для облегчения решения проблемы подбора подходящих сигналов, управляющих формированием сигнатурного окна, предусмотрено программное управление выходными сигналами пуска/останова. Эти сигналы вместе с сигналом синхронизации и общим проводом выведены на край приставки. Для того чтобы придать высокий или низкий уровень выходному сигналу пуска/останова, программа тестовых воздействий должна лишь осуществить запись в память приставки по адресу управления (F000 в процессоре 6502, 0000 в процессоре Z80). Младший бит информации, записанной по управляющему адресу, поступает на выходной контакт пуска/останова. Таким образом, под действием программы фрагмента

```
LDA #XXXX XXX0
STA F00016
```

выходной сигнал пуска/останова принимает низкий уровень, а фрагмент

```
LDA #XXXX XXX1
STA F00016
```

придает ему высокий уровень.

Глава 10

Контроль с применением ЭВМ

Эволюция методов и средств контроля на каждом этапе развития технологии заключается в переходе от индивидуального тестирования к массовому, от ручного к автоматическому. На ранних стадиях этого перехода автоматические системы контроля создаются отдельными пользователями для своих организаций, при этом стоимость аппаратуры может быть невысокой, а стоимость разработки вообще не принимается во внимание. По мере становления автоматизированных средств контроля и роста интереса к ним в дело включаются специализированные фирмы-изготовители, которые в свою очередь разрабатывают новые автоматические методы тестирования. Рынок производимых ими систем по-прежнему ограничен, а в разработках наблюдается генденция к проектированию больших, дорогих систем для крупносерийного производства. Впоследствии накапливаемый опыт делает возможным создание упрощенных вариантов тех же систем, доступных широкому кругу потребителей. Со временем появляются идеи по созданию тестового оборудования, ориентированного на применение на рабочем месте инженера или техника. Проблемы здесь несколько другие: необходимо осуществлять тестирование или обслуживание небольшого числа изделий, возможно, совершенно различных типов, а не контроль качества изготовления большого числа одинаковых изделий. Тестовая аппаратура для этих двух областей выглядит по-разному, и методы, используемые в одной из них, могут представлять лишь незначительный интерес для другой.

Можно наметить еще один этап развития методов и средств контроля. Тестовое оборудование может по-прежнему казаться дорогим — ведь его стоимость, как правило, составляет значительную часть заработной платы пользователя! На это можно не обращать внимания, если аппаратура постоянно находится в применении, так как ее стоимость окупится экономией времени. Но по-прежнему нерешенной остается проблема небольших подразделений, научно-исследовательских лабораторий, учебных заведений и центров подготовки специалистов. Во всех

```

{0A00} 29 RAMSTA EQU 00A0H
{0B00} 30 RAMSTB EQU 00B0H
{0C00} 31 RAMSTC EQU 00C0H
{0D00} 32 RAMSTD EQU 00D0H
{0E00} 33 RAMSTE EQU 00E0H
{0F00} 34 RAMSTF EQU 00F0H
          35 ORG 0F800H
          36 EXTEND
          37 ; ЗАГРУЗКА В АККУМУЛЯТОР ТЕСТОВЫХ ДАННЫХ 55
          38 LDA #055H
          39 ; ОЧИСТКА РЕГИСТРА X
          40 LDX #000H
          41 ; СТАРТОВЫЙ СИГНАЛ ДЛЯ ЗУПВ, СТОПОВЫЙ ДЛЯ ИНТЕРФЕЙСНОГО
          ; АДАПТЕРА
          42 START STA 0F000H
          43
          44 ; ПРОВЕРКА 256 БАЙТОВ ЗУПВ ЗА РАЗ,
          45 ; НАЧИНАЯ СО СТРАНИЦЫ 0
          46
          47
          48 LOOP 0 STA >RAMST0,X ; ЗАПИСЬ 01010101
          49 CMP >RAMST0,X ; ЧТЕНИЕ ЯЧЕЙКИ
          50 ASL A ; СДВИГ ВСЕХ БИТОВ ВЛЕВО
          51 STA >RAMST0,X ; ЗАПИСЬ 10101010
          52 CMP >RAMST0,X ; ЧТЕНИЕ ЯЧЕЙКИ
          53 LSR A ; СДВИГ ВСЕХ БИТОВ ВПРАВО
          54 INX
          55 BNE LOOP0 ; СЛЕДУЮЩАЯ ЯЧЕЙКА
          56
          57 ; ВЫПОЛНЕНИЕ ДЛЯ СЛЕДУЮЩИХ 256 БАЙТОВ, Т.Е. СТРАНИЦЫ 1
          58

```


STX 0F000H

7*

F917	8EF000	230	
231		232	; ЗДЕСЬ НАЧИНАЕТСЯ ПРОВЕРКА ПОРТОВ
233			
F91A	A9FF	234	LDA # 0FFH
F91C	8DA003	235	STA 0A003H ; ПОРТ А КАК ВЫХОДНОЙ
F91F	8DA002	236	STA 0A002H ; ПОРТ В КАК ВЫХОДНОЙ
F922	A9AA	237	LDA # 0AAH ; ЗАГРУЗКА "ШАХМАТНОЙ ДОСКИ"
F924	8DA000	238	PORTL1 STA 0A000H ; ЗАПИСЬ 10101010 В В
F927	8DA001	239	STA 0A001H ; ЗАПИСЬ 10101010 В А
F92A	CDA000	240	CMP 0A000H ; ЧТЕНИЕ ПОРТА В
F92D	CDA001	241	CMP 0A001H ; ЧТЕНИЕ ПОРТА А
F930	E8	242	INX ; ПОВТОРЕНИЕ
F931	D0F1	243	BNE PORTL1 ; ВЫПОЛНЕНО 256 РАЗ
F933	4A	244	LSR A ; СДВИГ ВПРАВО 8 БИТ
F934	8DA001	245	PORTL2 STA 0A001A ; ЗАПИСЬ 01010101 В А
F937	8DA000	246	STA 0A000H ; ЗАПИСЬ 01010101 В В
F93A	CDA001	247	CMP 0A001H ; ЧТЕНИЕ ПОРТА А
F93D	CDA000	248	CMP 0A000H ; ЧТЕНИЕ ПОРТА В
F940	E8	249	INX ; ПОВТОРЕНИЕ
F941	D0F1	250	BNE PORTL2 ; ВЫПОЛНЕНО 256 РАЗ
F943	4CF804	251	JMP START ; НАЧАТЬ ЦИКЛ СНАЧАЛА

Программа 9.1. Программа тестовых воздействий для контроля ЗУПВ и средств ввода-вывода.

При задании режима тестового воздействия эмуляционная приставка выполняет запись в 4К-байтовый блок памяти тестируемого устройства. Если речь идет о модели приставки, пред-

назначенной для работы с Z80, то информация, направляемая процессором по любому адресу в диапазоне от 0000_{16} до $0FFF_{16}$, не попадает в целевую систему, а направляется вместо этого в СППЗУ программы тестовых воздействий. В модели, ориентированной на процессор 6502, СППЗУ занимает старшие 4 К байт памяти с адресами в диапазоне $F000_{16}$ — $FFFF_{16}$.

Программа генерации тестовых воздействий для одноплатной ЭВМ

На рис. 9.12 приведена схема алгоритма стимулирующей программы (программа 9.1), предназначеннной для тестирования ЗУПВ и пользовательских портов одноплатной микроЭВМ (см. рис. П2.1 в приложении 2). Была выбрана именно эта микроЭВМ, так как ее простая архитектура и набор ОЗУ, ЗУПВ и средств ввода-вывода делают ее удобной для демонстрации описываемого метода тестирования. Программа размещается в СППЗУ и ориентирована на использование приставки 6502.

После того как векторами начальной установки $FFFC_{16}$ и $FFFD_{16}$ обеспечено указание начального адреса программы тестовых воздействий, т. е. адреса $F800_{16}$, начинается выполнение программы путем придания низкого уровня сигналу пуска/останова записью холостого слова по управляющему адресу $F000_{16}$. Затем с помощью теста «шахматная доска», заключаю-

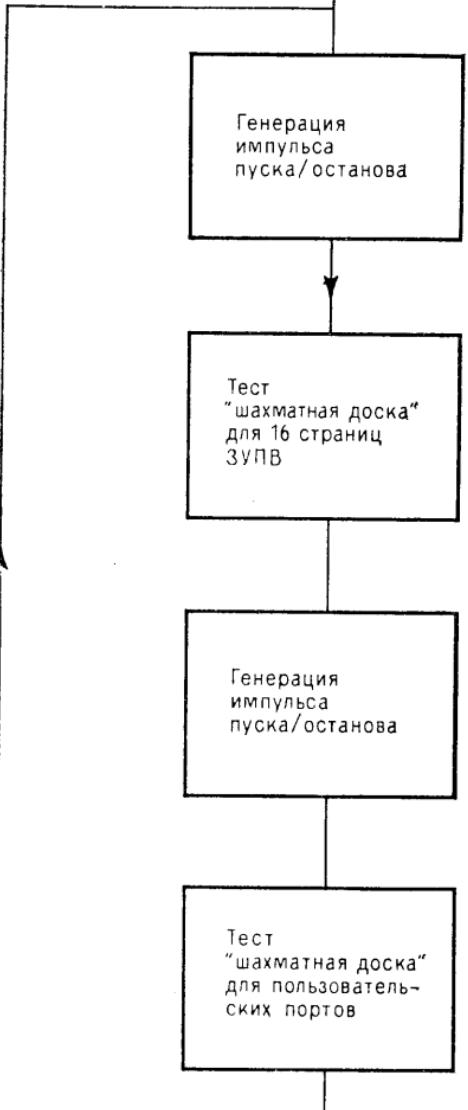


Рис. 9.12. Схема алгоритма программы тестовых воздействий для контроля памяти и портов.

тестовых воздействий, т. е. адреса $F800_{16}$, начинается выполнение программы путем придания низкого уровня сигнала пуска/останова записью холостого слова по управляющему адресу $F000_{16}$. Затем с помощью теста «шахматная доска», заключаю-

щегося в записи и чтении слов 55_{16} и AA_{16} , проверяются все ячейки ЗУПВ на плате микроЭВМ. Служащая для этого программа может показаться довольно примитивной по форме, так как в ней не используются ни обращения к подпрограммам, ни косвенная адресация. Эти действия не применены сознательно; дело в том, что для их реализации необходимо корректное функционирование как страницы 0, так и страницы 1, т. е. в работу должны включаться области памяти, работоспособность которых и должна быть выяснена с помощью теста. По оконча-

Сигнатурная таблица MEDC

Заголовок : AIM SIG9 : ЗУПВ Тестирование ЗУПВ

Функция	Пуск	Останов	Синхр.	Режим	Сигнатура	Сигнал	Сигнал	Синхросигнал
	↓	↓	↓	↓	2	питания	пуска	останова
					P254	памяти	памяти	памяти
Квалифициатор								
синхросигнала:	↓	По CS	По ЗУПВ	Задано	Отсутствует	Отсутствует	Отсутствует	Отсутствует
Точка								
схемы								
D0	Контакт ИС	Сигнатурा						
	Z2-14	Z6-14	Z11-14	Z17-14		958F		
D1	Z2-13	Z6-13	Z11-13	Z17-13			9CU7	
D2	Z2-12	Z6-12	Z11-12	Z17-12				958F
D3	Z2-11	Z6-11	Z11-11	Z17-11				
D4	Z3-14							
:	:	:	:	:	:	:	:	:

Рис. 9.13. Сигнатуры ЗУПВ.

нии тестирования ЗУПВ программа придает сигналу пуска/останова высокий уровень, а затем осуществляет тем же методом «шахматной доски» проверку двух пользовательских портов, после чего происходит возврат к началу программы и повторение всей процедуры контроля.

Сигнал пуска/останова определяет по существу два сигнатурных окна, одно из которых охватывает последовательность тестирования ЗУПВ, а другое — тестовую информацию портов. На рис. 9.13 приведены сигнатуры 8-битовой шины данных, полученные при тестировании ЗУПВ. Для ввода данных на вход синхронизации анализатора подается синхросигнал процессора ϕ_2 . Благодаря подаче на вход квалифициатора синхросигнала процессорного сигнала CS (Выбор кристалла) ЗУПВ в формировании окончательных сигнатур принимают участие только данные, передаваемые в ЗУПВ или из него, т. е. данные, появляющиеся на шине данных в процессе выполнения самой программы, в анализатор не поступают. Поэтому, как видно на рис. 9.13, сигнатуры линий данных через одну повторяются.

Наконец, на рис. 9.14 приведены сигнатуры, полученные при тестировании порта методом, аналогичным тестированию ЗУПВ. В качестве квалификатора синхросигнала используется сигнал \overline{CS} обращения к пользовательскому порту; при этом

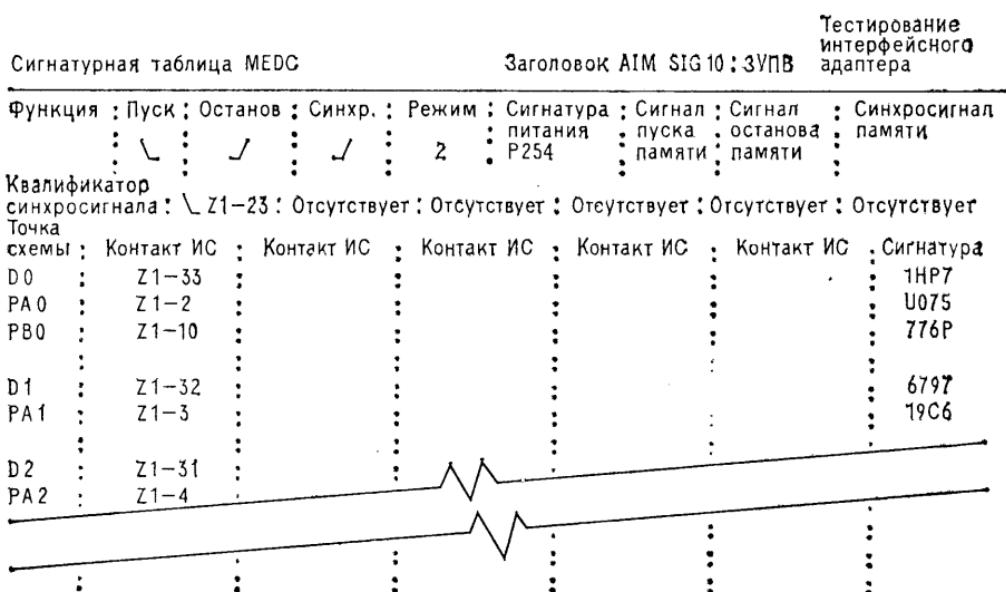


Рис. 9.14. Сигнатуры порта ввода-вывода.

в анализатор поступает только та информация с шины данных, которая передается по шине при операциях чтения и записи с участием пользовательских портов.

Заключение

Свободный прогон микропроцессора представляет собой относительно простой метод подачи тестовых воздействий на значительную часть микропроцессорной системы, который создает предпосылки для контроля устройств системы посредством сигнатурного анализа. Сигнатуры, снимаемые с адресной шины и дешифраторов, являются средством проверки работоспособности этих устройств. Сигнатуры шины данных позволяют не только проконтролировать работоспособность и содержимое ПЗУ, но и выявить дефекты шины, например константные сигналы, короткие замыкания и перемыкание линий, наличие устройств с постоянным разрешением выходных сигналов и т. д.

Для эффективного тестирования методами сигнатурного анализа ЗУПВ и устройств ввода-вывода необходимо применение

программ генерации тестовых воздействий. Такие программы должны обеспечивать повторяемость вырабатываемых потоков данных; корректность их выполнения зависит от правильности функционирования тестируемых компонентов.

Использование эмуляторов или тестеров позволяет применить методы сигнатурного анализа к существующим изделиям путем предоставления специалисту по контролю возможности подавать на тестируемые устройства необходимые тестовые воздействия.

ФАЙЛ: AIMSIG:AIM

HEWLETT-PACKARD: АССЕМБЛЕР 6500

ОБЛАСТЬ ОБЪЕКТ-
ПАМЯТИ НЫНЬ КОД

1

“650X”,

2 * ****
3 * ПРОГРАММА СТИМУЛИРУЮЩИХ ВОЗДЕЙСТВИЙ AIM65 ДЛЯ
4 * СИГНАТУРНОГО АНАЛИЗА.ПРЕДНАЗНАЧЕНА ДЛЯ ПРОВЕРКИ 4К ЗУПВ
5 * ПОВТОРЯЮЩИМИСЯ 255-БАЙТОВЫМИ БЛОКАМИ,ПРОВЕРЯЮТСЯ
6 * ТАКЖЕ ПОРТЫ ВВОДА-ВЫВОДА УНИВЕРСАЛЬНОГО ИНТЕРФЕЙСНОГО
АДАПТЕРА

7 * ****
8 * ЧТОБЫ НАЧАЛОСЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ,ВЕКТОР НАЧАЛЬНОЙ
9 * УСТАНОВКИ ДОЛЖЕН СОДЕРЖАТЬ НАЧАЛЬНЫЙ АДРЕС
10 *
11

FFFF 00F8

12 ORG OFFFCN
13 HEX 00,F8 ; ЗАГРУЖЕН НАЧАЛЬНЫЙ АДРЕС
14 ; В ОЗУ ЗАПИСЫВАЮТСЯ,А ЗАТЕМ СЧИТЫВАЮТСЯ ТЕСТОВЫЕ
15 ; НАБОРЫ 55 И AA,ЭТОТ ТЕСТ НАЗЫВАЕТСЯ ШАХМАТНАЯ ДОСКА.
16
17
18 ; ВЫРАВНИВАНИЕ ВСЕХ БЛОКОВ ОЗУ

19 RAMST0 EQU 00000H
20 RAMST1 EQU 00100H
21 RAMST2 EQU 00200H
22 RAMST3 EQU 00300H
23 RAMST4 EQU 00400H
24 RAMST5 EQU 00500H
25 RAMST6 EQU 00600H
26 RAMST7 EQU 00700H
27 RAMST8 EQU 00800H
28 RAMST9 EQU 00900H

в этих случаях контрольное оборудование может или использоваться от случая к случаю, или, как это имеет место при обучении, требоваться в больших количествах. Если удастся в большей степени реализовать возможности микроЭВМ, уже имеющихся во многих организациях, стоимость тестового оборудования может быть снижена. Пусть микроЭВМ возьмет на себя управление, вычисления и индикацию. В этой главе будут рассмотрены все три вида автоматического контрольного оборудования:

- автоматическая контрольная аппаратура (АКА);
- полуавтоматическое контрольное оборудование для производственных условий;
- контрольное оборудование с управлением от микроЭВМ.

Автоматическая контрольная аппаратура

Мы можем сказать об этой аппаратуре несколько слов, не углубляясь в подробности. Предмет изучения здесь обширен, изделия дороги, а круг лиц, имеющих к ним отношение, относительно узок. Поэтому автоматическое контрольное оборудование применяется в основном в процессе производства на крупных фирмах-изготовителях. Оно должно быстро обрабатывать множество изделий. Например, при изготовлении полупроводниковых компонентов необходим контроль ИС после сборки, но нужно и проверять кристаллы на многих предшествующих этапах производства. Бессмысленно тратить деньги на присоединение проводников к кристаллу, который потерял работоспособность уже на технологическом этапе диффузии; не менее бессмысленно тратить деньги на продолжение диффузии, если ранее имели место нарушения в проведении процесса очистки и т. д.

Некоторые из тестов могут быть одинаковыми для многих разнообразных полупроводниковых приборов, например проверка качества металлизации и глубины диффузии. Другие специфичны для конкретных приборов; так, проверка напряжения смешения в операционном усилителе не имеет ничего общего с измерением времени задержки логического вентиля. Все виды тестирования объединяет возможность подключения к нескольким, хотя бы наиболее характерным точкам схемы и подачи в некоторые из этих точек определенных последовательностей напряжений или токов с одновременным наблюдением результирующих сигналов в других точках. Тесты часто являются статическими, представляя собой проверку соответствия быстро сменяющихся выходных напряжений последовательности входных, причем каждый набор входных напряжений удерживается достаточно длительное время для того, чтобы установились стабильные значения выходных. Такие тесты могут

быть с успехом применены для проверки законченных устройств, конструктивно оформленных на печатных платах. При этом на заданные точки платы, обычно выведенные на торцевой разъем, подаются входные сигналы, и реакция платы на их подачу сравнивается с ожидаемой. Количество подаваемых сигналов может быть ограничено соображениями повышения скорости контроля в предположении, что сокращенный контроль все-таки выявит подавляющее большинство неисправных ИС при минимальных затратах. Это так называемый функциональный контроль устройства, т. е. контроль правильности логики его функционирования.

Если логическая схема конструктивно представляет собой часть печатной платы, то контроль логической схемы можно было бы более точно называть контролем устройства, хотя обычно его называют внутрисхемным тестированием. Итак, мы уже имеем иерархию тестов:

- тесты устройств;
- внутрисхемные тесты;
- функциональные тесты.

Эту иерархию можно продолжить, рассматривая плату как составную часть некоторого прибора или аппаратного блока, который в свою очередь является компонентом системы. Для контроля составных систем можно либо попробовать применить функциональное тестирование системы в целом, либо разработать средства контроля отдельных блоков аппаратуры. Имея дело с неисправным прибором, следует искать отказы или с помощью функционального тестирования прибора в целом, или на уровне отдельных плат.

Подобный подход, в соответствии с которым «то, что один человек считает устройством, для другого является системой», составляет основу совместно развивающихся в последние годы внутрисхемного и функционального тестирования. С помощью простых логических тестов, описанных в гл. I, выполняется внутрисхемное тестирование, при котором определенные тестовые наборы подаются на цифровые устройства без снятия этих устройств с платы. Функционирование системы в целом при этом не проверяется. Неисправности локализуются с точностью до одной ИС, но причины искажения временных соотношений не выявляются. Логический анализ представляет собой средство функционального тестирования, потому что используемые в нем сигналы являются «реальными», т. е. действия выполняются с максимальной скоростью, что позволяет выявлять временные ошибки.

Общие принципы построения автоматического тестового оборудования иллюстрируются рис. 10.1. Рассмотрим содержание этой схемы.

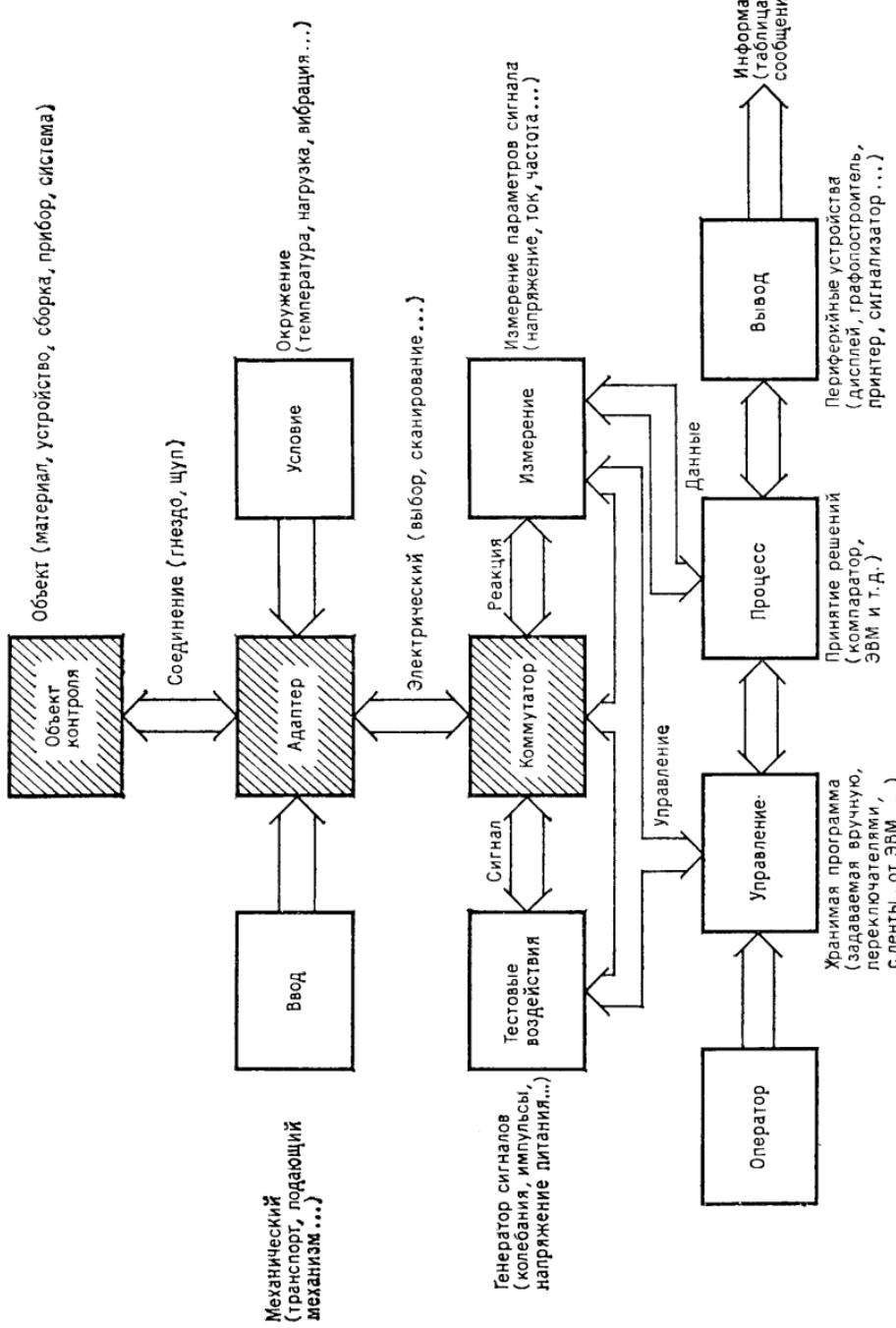


Рис. 10.1. Функциональное представление измерительной системы.

Объект контроля. Объектом контроля может быть материал, из которого сделан компонент, устройство, в состав которого входит компонент, или сборка из нескольких устройств, прибор, в составе которого используется эта сборка, или система, в состав которой входит этот прибор.

Адаптер. Служит для подсоединения объекта контроля к системе. Адаптер может представлять собой испытательный стенд, гнездо, кабель, любое средство двустороннего обмена информацией между объектом и системой.

Ввод. Может понадобиться по очереди подключать к одной и той же системе контроля множество устройств. Оборудование ввода должно включать механизмы подачи компонентов, руки роботов, транспортеры.

Условие. Для того чтобы тестирование имело смысл, объект контроля должен работать в определенных условиях, к числу которых могут относиться параметры источника питания, окружающая температура, механическая ориентация прибора.

Коммутатор. Тестируемое устройство или прибор могут иметь множество контактов. Коммутирующая часть системы может осуществлять выбор контактов, которые должны быть подключены к системе в любой момент времени, с помощью мультиплексоров или релейных схем.

Тестовые воздействия. В качестве их могут потребоваться самые разнообразные сигналы от синусоидальных колебаний до импульсов высокой частоты и сигналов управляемой формы. В автоматических системах контроля может оказаться необходимой возможность управления сигналами в широком диапазоне параметров.

Измерение. В системе должен быть предусмотрен широкий набор измерительных средств, включающий цифровые вольтметры, частотомеры, фазометры и т. д. Они также должны быть управляемыми, чтобы обеспечивалось соответствие диапазонов их измерений параметрам используемых тестовых воздействий.

Управление. Оно может быть достаточно простым, как, например, ручное операторное управление. В автоматических системах оператор только запускает и останавливает процесс, но в системе должны быть также устройства управления с предварительным программированием, которые могут быть простыми, как плата с перемычками, или сложными, как мини-ЭВМ.

Данные. Информация, получаемая от измерительной системы, может иметь вид, не удобный ни для оператора, ни для

управляющей программы. Весьма типична ситуация, когда в зависимости от результатов предыдущего теста должна быть выполнена та или иная последовательность других тестов. В этом отношении характерной ситуацией является выход значений измеряемых величин за максимальные и минимальные пределы, установленные для данного прибора. Блок ПРОЦЕСС включает помимо всего прочего выполнение преобразований данных, таких, например, как перевод из последовательной формы в параллельную.

Вывод. В конце концов информация передается на индикаторы или регистрирующие устройства, чтобы ею мог воспользоваться человек — наблюдатель. Такая информация может представлять данные, необходимые для полной кольцевой проверки измерительной системы.

Никакая подобная схема не может учесть все возможные нюансы построения систем контроля. Некоторые из названных функций могут быть объединены в одном блоке, другие, здесь не упомянутые, могут вызвать затруднения с точки зрения отнесения их к той или иной части схемы. Следует сделать два заключительных замечания. Для того чтобы системы кратко охарактеризованного выше класса можно было называть полностью автоматическими, они должны быть, во-первых, оптимальным образом ориентированы на решение задач тестирования и отыскания неисправностей, а во-вторых, способны принимать решения по ходу выполнения теста.

Хотя полностью автоматические системы, предназначенные для тестирования полупроводниковых приборов в процессе их массового изготовления, а также для внутрисхемного и функционального контроля плат, изготавливаемых крупными сериями, весьма дорогие, некоторые из заложенных в них принципов находят применение в относительно недорогих установках, первоначально предназначаемых для решения задач обслуживания. Их производительность достаточно высока, чтобы их можно было использовать на линиях поточного производства и в наладочных подразделениях, и они лишь начинают находить применение для обслуживания изделий в условиях эксплуатации, которое было целью разработки подобных установок.

В следующем разделе описаны некоторые другие варианты построения автоматического контрольного оборудования и средств контроля, реализуемых на основе применения ЭВМ.

Автоматизированный контроль

Логический анализ, внутрисхемная эмуляция и сигнатурный анализ представляют собой эффективные средства поиска неисправностей в микропроцессорных системах. Однако у них есть

недостатки, заключающиеся в трудности их применения без детального знания системы и в трудоемкости их практического воплощения, которое требует разработки и реализации программ генерации тестовых воздействий, а также большого объема документации.

Некоторые фирмы пришли к мысли о необходимости изготавления контрольной аппаратуры, которая содержала бы называемые средства, но в форме, делающей возможным использование этой аппаратуры не очень квалифицированным персоналом. Например, фирма IBM разработала для собственного применения систему, которая получила название «установка для обслуживания». Это компактный прибор с внутренней микроЭВМ, который применяют эксплуатационники фирмы. В его составе имеется портативная клавиатура с индикатором; загрузка программ, необходимых для выполнения обслуживания, производится из встроенной памяти на гибких дисках. Интерфейс с обслуживаемыми устройствами осуществляется с помощью набора специальных адаптеров.

В то же время известны такие средства обслуживания, как прибор для поиска неисправностей Fluke 9010A, ориентированный на более широкий рынок сбыта и представляющий собой простую в обращении установку, которая может быть применена для отладки самого разнообразного микропроцессорного оборудования. Ниже приводятся некоторые характеристики этого прибора и описывается его применение для обслуживания типичной небольшой микропроцессорной системы, содержащей различные запоминающие устройства и устройства ввода-вывода.

Многие процедуры обслуживания, такие, например, как тестирование ввода-вывода, являются в значительной степени одинаковыми для всех микроЭВМ, различающихся между собой в непринципиальных деталях.

Прибор для поиска неисправностей Fluke 9010A

Прибор 9010А построен на базе микропроцессорной ЭВМ, носящей название центральной. Соединения с тестируемым устройством осуществляются через интерфейсную переходную приставку, которая включается в процессорное гнездо целевой системы. С помощью универсального логического пробника можно ввести в прибор логические сигналы из тестируемого устройства и подать в любую его точку тестовые воздействия (рис. 10.2). Каждая интерфейсная приставка предназначена для работы с определенным микропроцессорным комплектом. Она содержит микропроцессор того же типа, что входит в комплект, и является основой реализации эмуляционных возможностей при-

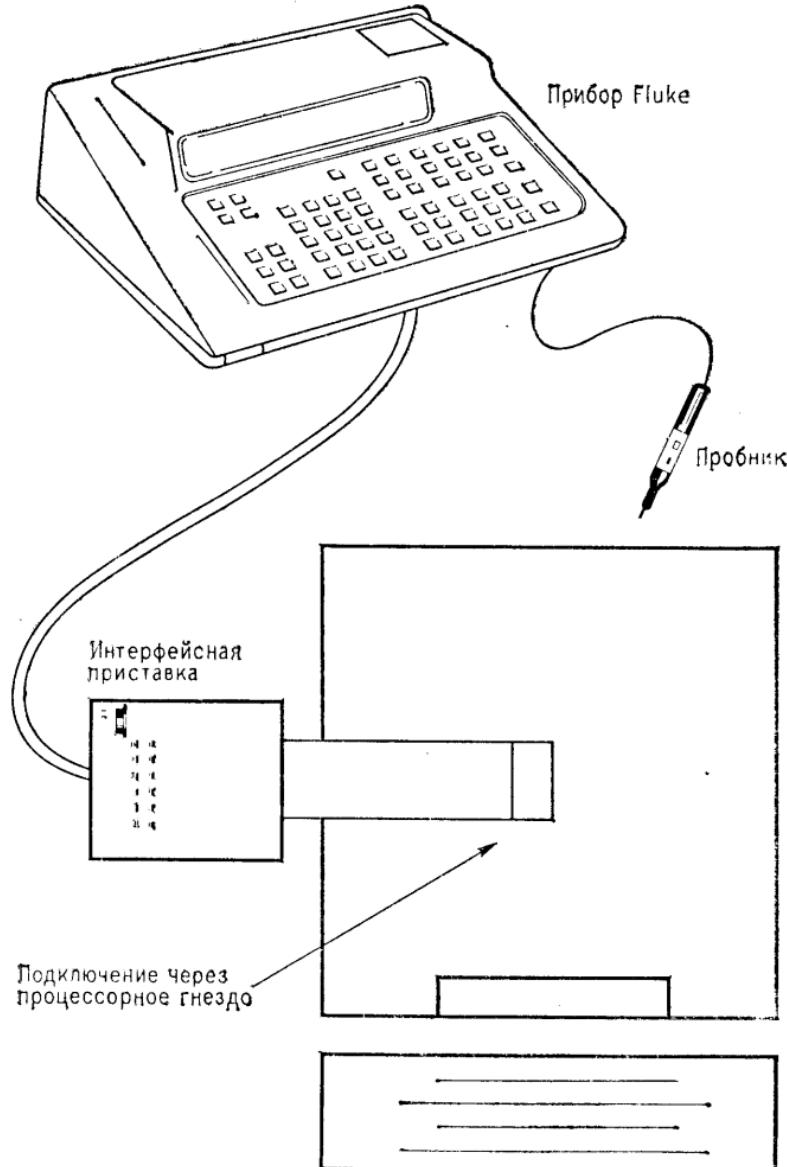


Рис. 10.2. Подключение прибора для поиска неисправностей Fluke 9010А к целевой системе.

бора 9010А. В составе центральной ЭВМ имеется кассетный магнитофон, служащий для хранения и по мере необходимости выборки тестовых программ и описаний тестируемых устройств. Оператор взаимодействует с прибором посредством клавиатуры и 32 светодиодных символьных индикаторов. Прибор снабжен, кроме того, последовательным интерфейсом ввода-вывода, соответствующим стандарту RS232.

Операционные возможности

Прибор 9010А выполняет функции внутрисхемного эмулятора, принимая на себя управление шинами адресов, данных и управления тестируемого устройства через его процессорное гнездо. Он располагает набором специальных функциональных возможностей, которые упрощают и ускоряют процедуры тестирования. Диапазон этих возможностей простирается от режима обучения, при котором воспроизводится карта памяти тестируемого устройства (во всем диапазоне адресов ЗУПВ, ПЗУ и ввода-вывода), до режима тестирования, при реализации которого используются либо аппаратно зафиксированные тестовые программы, либо программы, сделанные на заказ и загруженные из кассетного магнитофона.

Пример: реализация программы тестирования одноплатной микроЭВМ

Набор устройств проверяемой одноплатной ЭВМ AIM 65 фирмы Rockwell International типичен для многих известных микроЭВМ. В ней использован процессор 6502, статическое ЗУПВ емкостью 4К байт, а также резидентное ПЗУ, объем которого может достигать 20К байт и определяется применяемыми программными средствами. Имеется широкий набор интерфейсов, с помощью которых могут быть подключены следующие устройства:

- матричная клавиатура (интерфейс чтения с терминала ввода-вывода PIOT 6532);
- принтер (часть универсального интерфейсного адаптера VIA 6522);
- 20-разрядный светодиодный индикатор (адаптер параллельного интерфейса PIA 6520);
- устройства хранения информации на ленте и устройства с токовой петлей (последовательный интерфейс — часть универсального интерфейсного адаптера VIA 6522);
- пользовательский параллельный порт (VIA 6522).

Перед генерацией тестовой программы необходимо предварительно составить представление о работе системы, являющейся объектом контроля. Для этого с платы снимается микропроцессор и вместо него подключается интерфейсная приставка прибора 9010А. После этого на плату подается питание и оператор начинает исследование карты памяти системы.

Составление карты памяти

Специфической особенностью прибора 9010А является наличие среди его функциональных возможностей режима обучения. Алгоритм обучения состоит в исследовании адресного простран-

ства заведомо исправной системы путем записи данных в каждую ячейку и последующего чтения содержимого этой ячейки. В зависимости от значения считываемых данных алгоритм осуществляет измерение областей памяти и определяет их принадлежность к ПЗУ, ЗУПВ и системе ввода-вывода, составляя тем самым карту распределения памяти. Для каждого блока ПЗУ

ADDRESS SPACE INFORMATION					
RAM	@	0000-0FFF			
RAM	@	A400-A47F			
ROM	@	B000-BFFF	STG	EA7C	
ROM	@	C000-CFFF	STG	E7E4	
ROM	@	D000-DFFF	STG	7915	
ROM	@	E000-EFFF	STG	6E7E	
ROM	@	F000-FFFF	STG	BD87	
I/O	@	A002-A003	BITS	FF	
I/O	@	A00A-A00C	BITS	FF	
I/O	@	A481	BITS	FF	
I/O	@	A483	BITS	FF	
I/O	@	A802-A805	BITS	86	
I/O	@	A80A	BITS	FF	
I/O	@	AC00-AC03	BITS	3F	

Рис. 10.3. Информация о распределении адресов AIM 65, полученная в режиме обучения.

генерируется сигнатура путем выработки циклического избыточного кода его содержимого.

По желанию этап обучения может быть исключен путем редактирования или заменен путем ввода адресной информации непосредственно с клавиатуры. Один из вариантов результирующей карты памяти представлен на рис. 10.3. Адресная информация, полученная в режиме обучения, в данном случае была отредактирована, чтобы получить отдельные сигнатуры для каждой 4К-байтовой ИС ПЗУ, а не общую сигнатуру для всего 20К-байтового блока ПЗУ.

Тестовые возможности 9010А

После формирования карты памяти работающей системы можно с помощью прибора 9010А тестировать аналогичные платы, пользуясь встроенным тестовым программами. В центральной ЭВМ прибора предусмотрены микропрограммы контроля работы системных шин, ПЗУ, ЗУПВ и частичного контроля устройств ввода-вывода.

— При контроле шин проверяется электрическая целостность

шин адресов, данных и управления, для чего на шины подаются тестовые наборы и проверяется успешность их прохождения.

— Контроль ПЗУ осуществляется с использованием тех же циклических избыточных кодов, которые фигурируют в режиме обучения. При любом отклонении от исходных сигнатур выводится соответствующее сообщение на индикацию.

— Для контроля ЗУПВ имеются два теста, короткий и длинный. С помощью короткого теста проверяются возможность чтения/записи информации в каждую битовую ячейку памяти и правильность функционирования дешифратора адреса ЗУПВ. Длинный тест ЗУПВ обеспечивает более полное тестирование, включающее проверку помехоустойчивости памяти при работе с наборами данных, представляющими собой «худшие случаи».

— Контроль схем ввода-вывода выполняется аналогично контролю ЗУПВ, т. е. проверяется работоспособность битовых ячеек, про которые в режиме обучения выяснено, что они допускают и чтение, и запись информации.

Программное тестирование

Любой из перечисленных выше встроенных тестов может быть инициирован непосредственно с клавиатуры или вызван из выполняющейся программы. В приборе 9010А используется язык программирования, аналогичный Бейсику и позволяющий программисту объединять тесты для формирования полной тестовой последовательности, содержащей сообщения оператору о том, какие действия он должен выполнять в определенные моменты. Программные средства не ограничиваются возможностью формирования последовательностей встроенных тестов. Они включают также простой метод составления программ для более полного тестирования устройств ввода-вывода. Ниже рассматриваются примеры таких программ, предназначенных для контроля трех многофункциональных интерфейсных ИС, входящих в состав одноплатной ЭВМ.

Контроль универсального интерфейсного адаптера 6522

Адаптер 6522, предназначенный для прикладных применений, содержит два порта ввода-вывода, которые выведены на разъем, находящийся на задней стенке прибора. Для проведения тестирования к разъему подключается ответная часть, в которой проводным монтажом соединены соответствующие контакты портов А и В (A0 с B0, A1 с B1 и т. д.). Тестовый набор записывается в порт В, а затем считывается в порт А и проверяется. Процедура выполняется с «нарастанием» тестовых наборов,

PROGRAM 2 190 BYTES

```

DPY-USER VIA TEST
REG1 = 00
WRITE @ A003 = 00
WRITE @ A002 = 00
WRITE @ A002 = FF
1: LABEL 1
WRITE @ A000 = REG1
READ @ A001
IF REG1 = REGE GOTO 3
2: LABEL 2
DPY-VIA ERROR*
STOP
3: LABEL 3
INC REG1
IF 100 > REG1 GOTO 1
REG1 = 00
WRITE @ A002 = 00
WRITE @ A003 = FF
4: LABEL 4
WRITE @ A001 = REG1
READ @ A000
IF REG1 = REGE GOTO 5
GOTO 2
GOTO 5
5: LABEL 5
INC REG1
IF 100 > REG1 GOTO 4
DPY-VIA OK

```



Программа 10.1. Тестовая программа для пользовательского универсального интерфейсного адаптера.

т. е. с изменением их от 00 до FF. По ее завершении порты меняются ролями и вся она повторяется. При обнаружении любой ошибки оператор получает сообщение VIA ERROR, сопровождаемое звуковым сигналом и остановкой теста (программа 10.1).

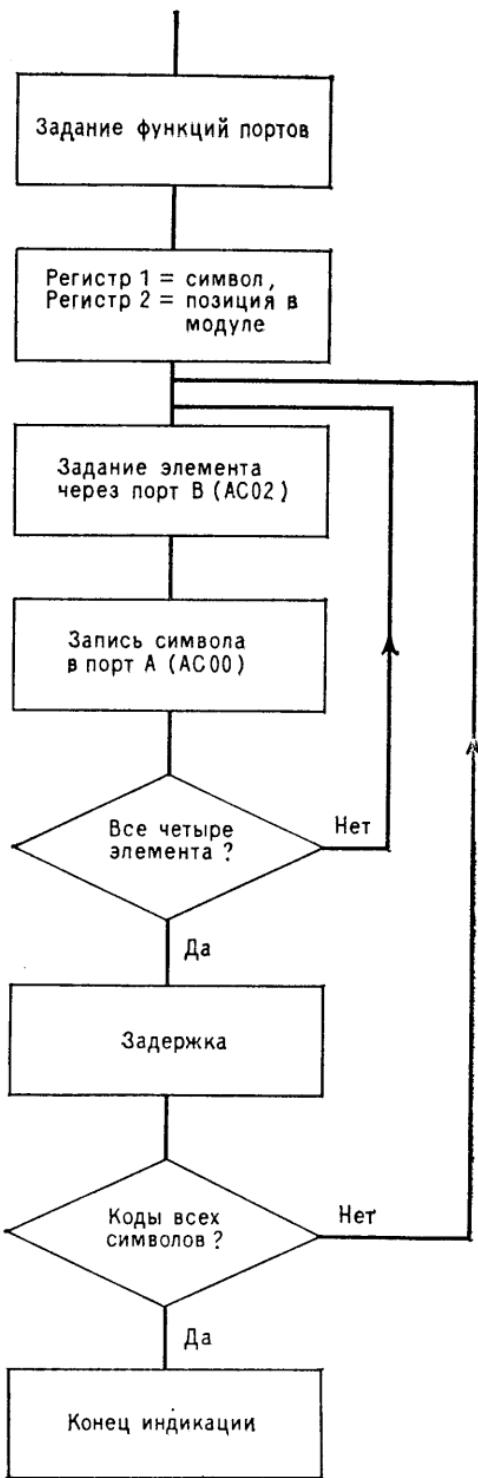
В составе прибора 9010А имеется шестнадцать 32-разрядных регистров, пронумерованных от 0 до F, содержимое которых может быть использовано в качестве переменных при арифметических операциях. Функции некоторых регистров специализированы. Так, в регистре E находится последнее слово данных, использованное оператором или выработанное в приборе 9010А. В приведенном примере регистр 1 используется для хранения текущего значения тестового набора, записанного в порт вывода, а в регистре E размещаются данные, полученные из порта ввода при операции чтения.

PROGRAM O 194 BYTES

```

DPY-DISPLAY TEST****
WRITE @ AC01 = 00
WRITE @ AC03 = 00
WRITE @ AC00 = FF
WRITE @ AC02 = FF
WRITE @ AC01 = 04
WRITE @ AC03 = 04
REG1 = A0
REG2 = 00
1: LABEL 1
DPY- AIM CHARACTER CODE
DPY-$1
2: LABEL 2
WRITE @ AC00 = REG2
WRITE @ AC02 = REG1
INC REG2
IF 4 > REG2 GOTO 2
REG3 = 0
3: LABEL 3
INC REG3
IF 20 > REG3 GOTO 3
REG2 = 0
INC REG1
IF EO > REG1 GOTO 1
DPY-FINISHED*

```



Контроль светодиодного индикатора и интерфейса 6520

Двадцатиразрядный светодиодный индикатор состоит из пяти 4-разрядных индикаторных модулей, каждый из которых содержит собственную внутреннюю память и схему управления светодиодными сегментами (см. приложение 2). Индикатор подключается к шинам адресов, данных и управления с помощью адаптера параллельного интерфейса 6520. Для вывода символа на индикатор с помощью порта А интерфейса 6520 выбираются индикаторный модуль и позиция отдельного символа в этом модуле, а через порт В в эту позицию передается ASCII-код нужного символа.

Тестовая программа начинается с инициализации обоих портов как портов вывода (программа 10.2). Затем через порт А задается позиция символа, а через порт В выводится сам символ. Программа подает разрешение сразу на все пять модулей и с помощью программного цикла выполняет запись выводимого символа во все позиции всех индикаторных модулей. Затем выбирается следующий символ, и процесс повторяется, пока через каждую позицию индикатора не пройдет весь набор символов. Тестовая процедура проходит достаточно медленно для того, чтобы оператор мог визуально проконтролировать появление каждого символа в каждой позиции индикатора.

Контроль клавиатуры и интерфейса 6532

Матричная клавиатура подключается к процессору через интерфейс чтения с терминала ввода-вывода 6532. Порт А используется в качестве выходного и служит для стробирования столбцов матрицы. Порт В настраивается как порт ввода и обеспечивает просмотр строк матрицы в поисках нажатой клавиши. Программа тестирования клавиатуры (программа 10.3) начинается с инициализации портов интерфейса. Затем через порт А выводятся приведенные ниже двоичные наборы, содержащие указание оператору, какую клавишу он должен нажать.

Набор в порте А	Запрашиваемая клавиша
01111111	Пробел
10111111	Перевод строки
11011111	P
11101111	0
11110111	7
11111011	4
11111101	5
11111110	F1

С помощью этих действий программа выполняет быструю проверку всех строк и столбцов клавиатуры (очевидно, для

PROGRAM 3 319 BYTES

```

WRITE @ A481 = FF
WRITE @ A480 = FF
WRITE @ A483 = 00
WRITE @ A482 = 00
REG8 = FE
DPY-PRESS SPACE BAR ON AIM#
EXECUTE PROGRAM 4
REG8 = FD
DPY-SPACE BAR OK - TRY LF KEY
EXECUTE PROGRAM 4
REG8 = FB
DPY-LF KEY OK - TRY P#
EXECUTE PROGRAM 4
REG8 = F7
DPY-P OK TRY O-AS IN OMNIPOTENT
DPY-+#
EXECUTE PROGRAM 4
REG8 = EF
DPY-O OK -TRY 7#
EXECUTE PROGRAM 4
REG8 = DF
DPY-7 OK - TRY 4#
EXECUTE PROGRAM 4
REG8 = BF
DPY-4 OK - TRY S-AS IN SENSUOUS#
EXECUTE PROGRAM 4
REG3 = 7F
DPY-S OK - TRY F1 FUNCTION KEY#
EXECUTE PROGRAM 4
DPY-KEYBOARD TEST COMPLETE#

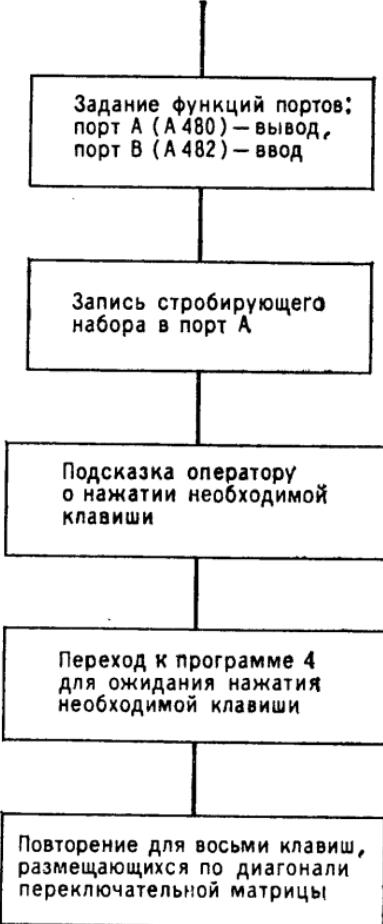
```

PROGRAM 4 · 37 BYTES

```

WRITE @ A480 = REG8
1: LABEL 1
READ @ A482
IF REG8 = REG8 GOTO 2
GOTO 1
2: LABEL 2

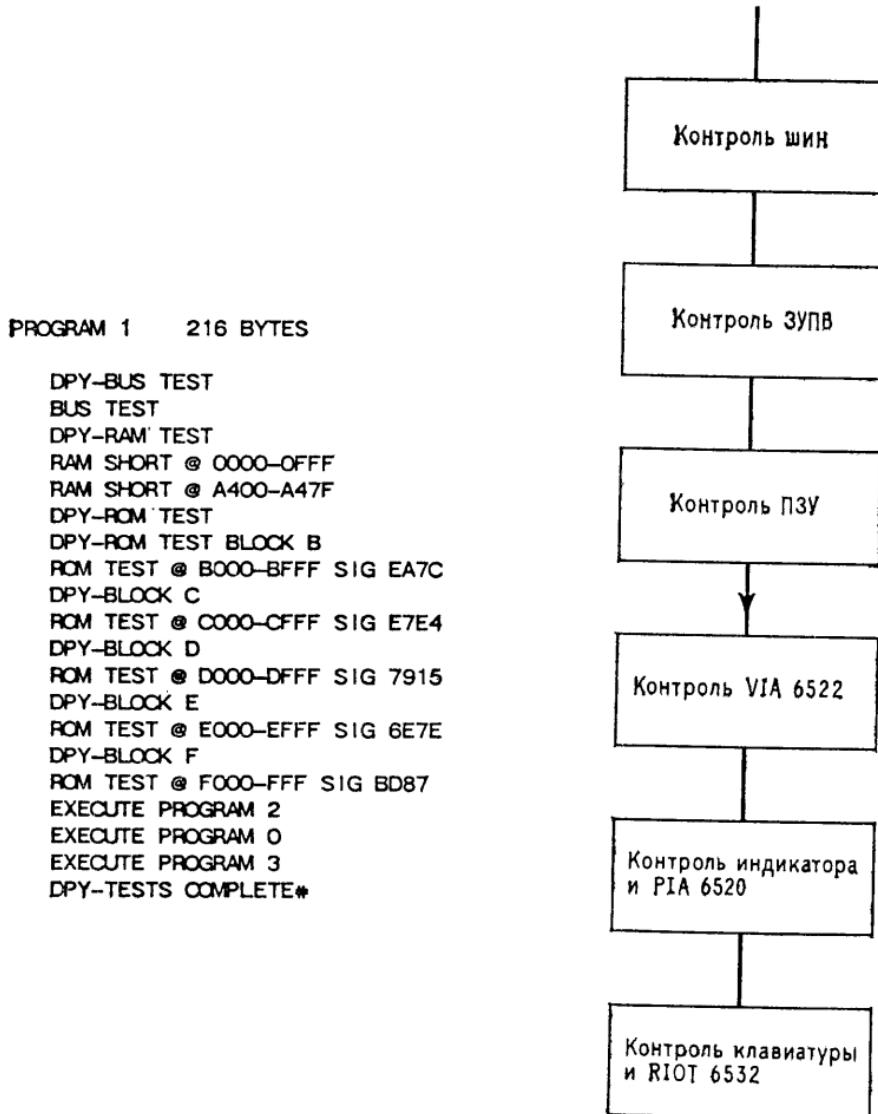
```



Программа 10.3. Короткий тест для матричной клавиатуры.

более тщательной проверки каждой клавиши и для выявления возможных замыканий между строками и столбцами можно было бы разработать более всесторонний тест). После нажатия правильной клавиши программа выдает оператору указание о нажатии очередной клавиши.

Этот пример иллюстрирует помимо всего прочего использование подпрограмм. Так, оператором «EXECUTE PROGRAM 4» вызывается подпрограмма, которая считывает информацию из порта А и обеспечивает ожидание нажатия нужной клавиши, после чего происходит возврат в основную программу 10.3.



Программа 10.4. Основная программа, управляющая тестовой последовательностью.

Формирование последовательности тестов

Программа 10.4 представляет собой итоговую контрольную последовательность, которая объединяет встроенные тесты системных шин, ЗУПВ и ПЗУ с описанными выше специальными программами проверки средств ввода-вывода. В данном случае результирующая программа имеет ярко выраженный линейный характер. Однако достаточно просто создать тестовую

программу, которая при обнаружении некорректного функционирования могла бы выдавать программы более детального контроля. При этом оператор получал бы с помощью сообщений, выводимых на индикатор, информацию об источнике неприятностей и указание о действиях, которые он должен выполнить, например, заменить определенную ИС.

Выходы

Рассмотренный конкретный пример иллюстрирует некоторые возможности прибора для поиска неисправностей 9010А. Разработка тестов для специальных устройств ввода-вывода требует хорошего знания системы, в составе которой они работают. Эта разработка упрощается за счет наличия простого языка программирования и возможности чтения и записи содержимого отдельных ячеек памяти. Истинная ценность подобных приборов определяется предоставляемым ими набором средств, необходимых при реализации процедур контроля, начиная с основной тестовой программы, позволяющей оператору следить за выполнением последовательности тестов, и кончая программами тестовых воздействий, составляющими каждый отдельный тест.

Сигнатурный анализ под управлением ЭВМ

Большой объем документации и необходимость руководства действиями оператора в ходе реализации тестовых последовательностей делают сигнатурный анализ идеальной сферой применения управления с помощью ЭВМ. Чтение сигнатур и сравнение их с образцовыми, нанесенными на аннотированной схеме или определенными в виде таблиц, — занятие утомительное и предрасполагающее к ошибкам. Обнаружив некорректную сигнатуру, оператор должен обратиться к дереву поиска неисправностей, с помощью которого можно найти причину отказа. Из рассматриваемого ниже примера видно, каким образом применение ЭВМ может облегчить решение обеих этих задач.

Необходимый комплекс технических средств показан на рис. 10.4. Программы генерации тестовых воздействий вводятся в тестируемое устройство с помощью эмуляционной приставки, описанной выше. Оператор реализует последовательность контроля, руководствуясь сообщениями, поступающими от системного контроллера НР85 (рис. 10.5). Все эти сообщения вместе с сигнатурами, сформированными в заранее работоспособной системе, и кодами конфигурации для прибора Locator хранятся на ленте в файле RECORDS (рис. 10.6).

Выполнение программы контроля начинается с перевода анализатора в режим дистанционного управления (рис. 10.7 и про-

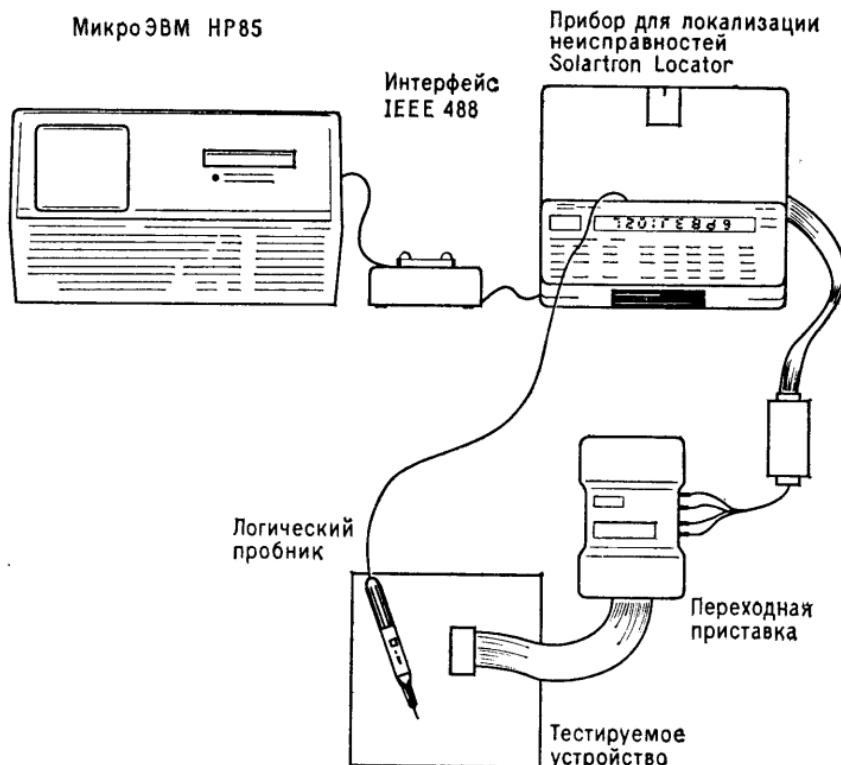


Рис. 10.4. Типовой комплекс технических средств контроля. Тестовое воздействие вводится от эмуляционной приставки. МикроЭВМ HP85 управляет прибором Locator через интерфейс IEEE 488.

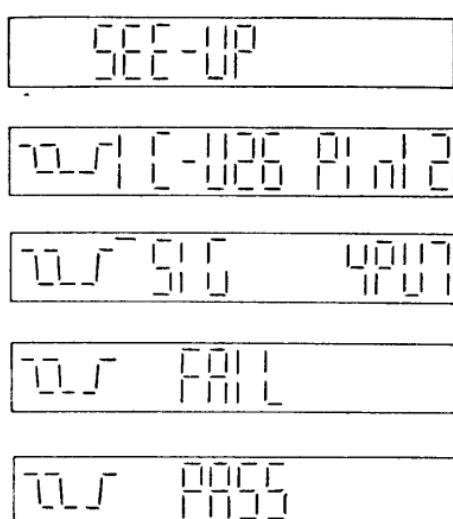


Рис. 10.5. Примеры сообщений, выводимых на индикацию.

грамма 10.5). Затем передаются настроечные коды, входящие в первую запись файла RECORDS и определяющие режим сигнатурного анализа и конфигурацию сигнатурного окна, т. е. полярность фронтов пуска/останова и синхронизации. По оператору 90 снова производится обращение к файлу RECORDS.

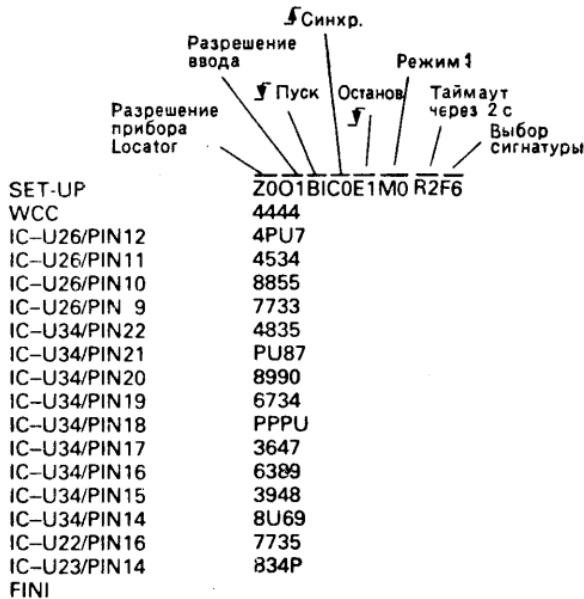


Рис. 10.6. Файл образцовых записей содержит конфигурационные коды. Следующие друг за другом записи состоят из операционных команд и сигнатур, полученных на заведомо работоспособной системе.

с целью получения сообщения для оператора и эталонной сигнатуры исследуемой точки.

В программе используется, как и в предыдущем примере, метод регистрации сигнатур нажатием кнопки «Фиксация»

```

1 REM *****
2 REM * ПРОГРАММА ДИСТАНЦИОННОЙ ПРОВЕРКИ СИГНАТУРЫ *
3 REM *****
5 CLEAR
10 REMOTE 716; REM ЛОКАТОР ПЕРЕВОДИТСЯ НА
    ДИСТАНЦИОННОЕ УПРАВЛЕНИЕ ПО АДРЕСУ 16
20 DIM A$(25),B$(25)
30 ASSIGN #1 TO "RECORDS"
40 ON KEV #1,"ABORT" GOTO 300
45 KEY LABEL
50 READ #1; C$,D$
60 OUTPUT 716; "TOP/"; C$; "/"; REM ДЛЯ НАЧАЛЬНОЙ УСТАНОВКИ
    ЛОКАТОР СЛУЖИТ ПЕРВАЯ
    ЗАПИСЬ ФАЙЛА "RECORDS"

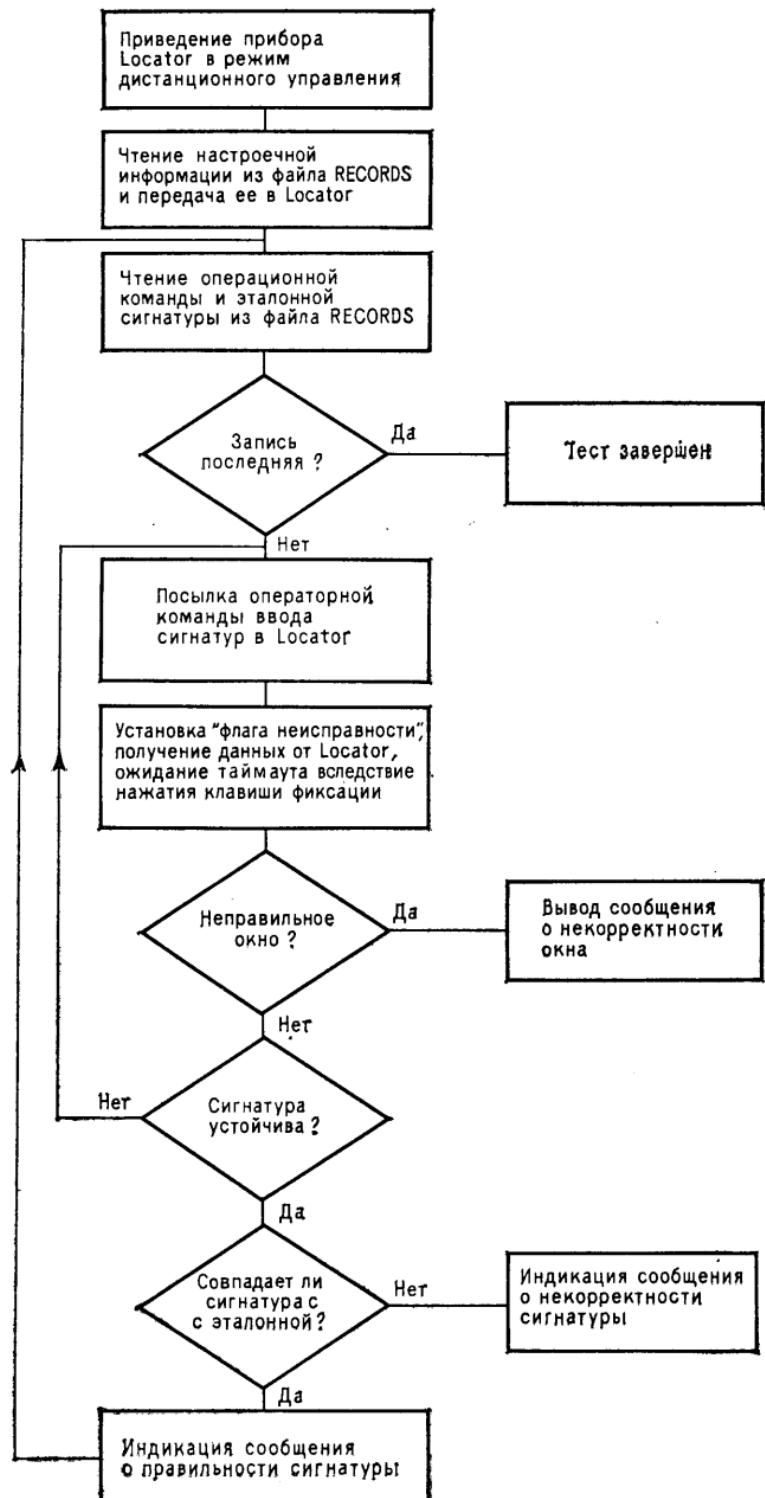
```

```

70 WAIT 3000
80 OUTRUT 716; D$
90 READ # 1; C$,D$: REM ПОСЛЕДУЮЩИЕ ЗАПИСИ СОДЕРЖАТ
    ИНСТРУКЦИИ ОПЕРАТОРА И
    КОРРЕКТНЫЕ СИГНАТУРЫ
100 IF POS(C$,"FINI") THEN 300
110 OUTRUT 716; "TOP/"; C$;"/"
120 WAIT 2000
130 OUTPUT 716 ;"T1": REM ВОЗОБНОВЛЕНИЕ НЕПРЕРЫВНЫХ
    ЗАМЕРОВ
140 GOSUB 800
145 IF POS(B$,"PROBLEM")=0 THEN 150: REM ПРОСТОЙ БЕЗ
    ЧТЕНИЯ СИГНАТУР
    ПРЕДПОЛАГАЕТ
    ПРОБЛЕМУ ВВОДА
147 PRINT B$@ BEEP
148 GOTO 300
150 IF POS(B$(17),"#") THEN 110: REM СИГНАТУРА СТАБИЛЬНА?
160 IF POS(B$,D$)=0 THEN 210: REM СОВПАДАЕТ ЛИ
    С ЭТАЛОННОЙ?
170 OUTPUT 716 ;"TOP/PASS/"
180 PRINT C$;"SIGNATURE OK": REM ЕСЛИ ДА ТО ПЕРЕЙТИ
    К СЛЕДУЮЩЕМУ ЗАМЕРУ
    ПОСЛЕ СООБЩЕНИЯ 'PASS'
190 WAIT 1000
200 GOTO 90
210 OUTPUT 716 ;"TOS2P/FAIL/": REM ЕСЛИ СИГНАТУРА НЕ
    СОВПАДАЕТ С ЭТАЛОННОМ,
    ВЫВЕСТИ СООБЩЕНИЕ
    ОБ ОШИБКЕ
220 PRINT "SIGNATURE FAIL ON "; C
300 END
310 REM ****
320 REM *
800 REM *          ОЖИДАНИЕ НАЖАТИЯ КНОПКИ
805 REM *
807 B$ = "START/STOP/CLOCK PROBLEM": REM УСТАНОВКА ФЛАГА
    "ПРОБЛЕМА ВВОДА"
810 ENTER 716 ;A$: REM ЧТЕНИЕ ЛОКАТОРА
820 IF POS(A$,"TIMEOUT") THEN 850: REM ОЖИДАНИЕ ТАЙМАУТА
    ВСЛЕДСТВИЕ
    НАЖАТИЯ КНОПКИ
    HOLD (ИЛИ ПРОБЛЕМА
    ВВОДА)
830 B$ = A$
840 GOTO 810: REM ЗАПИСЬ ПОСЛЕДНИХ ВХОДНЫХ ДАННЫХ в B$
850 RETURN
860 REM ****

```

Программа 10.5. Пример программы, служащей для контроля сигнатур. Программа управляет работой сигнатурного анализатора, выдает указание оператору через индикатор прибора Locator, воспринимает сигнатурь и сравнивает их с хранящимися в файле RECORDS.



Программа 10.5. Пример программы, служащей для контроля сигнатур. Программа управляет работой сигнатурного анализатора, выдает указание оператору через индикатор прибора Locator, воспринимает сигнтуры и сравнивает их с хранящимися в файле RECORD.

(HOLD); для этого служит подпрограмма, указанная в строке 800. Еще одно дополнительное преимущество программы заключается в способности реагировать на ситуацию «тайм-аута», возникающую при неправильной организации сигнатурного окна, т. е. при отсутствии сигналов пуска, останова или синхронизации. При возникновении этой ситуации выполнение теста прекращается и на индикатор ЭВМ HP85 выводится сообщение «Неправильный пуск/останов/синхронизация».

Vcc	SIGNATURE OK
IC-U26/PIN12	SIGNATURE OK
IC-U26/PIN11	SIGNATURE OK
SIGNATURE FAIL ON	IC-U26/PIN10

Рис. 10.8. Пример распечатки информации шинного контроллера (HP85); в данном случае четвертая зарегистрированная сигнатура не совпала с эталонной.

Проверка устойчивости сигнатур осуществляется оператором 190; нестабильные сигнатуры сопровождаются символом #. Наконец, принятая сигнатура сравнивается с соответствующей записью файла RECORDS. При отсутствии совпадения выполнение теста прерывается с выводом сообщения FAIL и описания точки, в которой обнаружена некорректность (рис. 10.8). В случае совпадения выполнение теста продолжается применительно к следующей точке и очередной записи файла RECORDS и т. д., пока не будет обнаружена последняя запись FINI.

Заключение

Управление тестированием с применением сигнатурного анализа и документирование этого тестирования упрощаются и ускоряются при управлении от ЭВМ. В рассмотренном примере структура программы упрощена, так как основная цель заключалась в пояснении принципов. Практические программы контроля могут быть сделаны более эффективными путем придания им способности принятия логических решений в соответствии с положением точки, в которой имеет место неисправность, что упростит работу оператора.

Часть III

Обслуживание с небольшими затратами

Глава 11

Тестеры на базе микроЭВМ

Не вызывает сомнений, что самый легкий путь внутрь микропроцессорной системы с целью ее отладки проходит через установочное гнездо процессора. Эмуляция, при которой микропроцессор удаляется из своего гнезда и к объекту тестирования

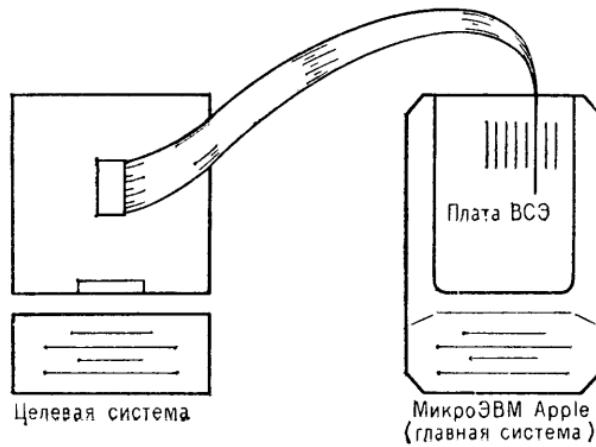


Рис. 11.1. Подключение целевой системы к основной через гнездо процессора и 40-проводный кабель.

(целевой системе) подключается через это гнездо другая (главная) микропроцессорная система, является на сегодня наиболее эффективным из известных средств диагностики.

Представление об эмуляторах обычно ассоциируется с дорогими системами проектирования или не менее дорогостоящими специально заложенными в изделие при разработке средствами поиска неисправностей. Существует, однако, возможность преобразования микроЭВМ в эмуляционную главную ЭВМ, позволяющую контролировать аппаратные средства и оценивать работоспособность программных средств целевой системы. В данной главе описывается простое устройство, которое, будучи применено совместно с микроЭВМ Apple II, превращает ее в при-

бор для поиска неисправностей, способный тестировать другие микроЭВМ, пользуясь своим собственным процессором.

Внутрисхемный эмулятор (ВСЭ) Apple II конструктивно оформлен в виде одной стандартной платы фирмы Apple. К целевой микроЭВМ эмулятор подключается посредством 40-проводного кабеля с резиновой изоляцией, на конце которого смонтирован разъем с двухрядным расположением штырей (рис. 11.1).

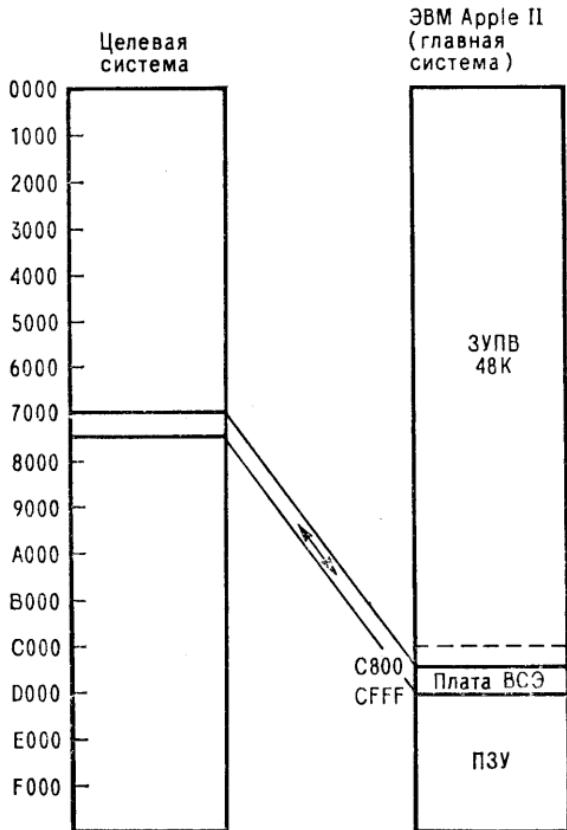


Рис. 11.2. Отображение любого 2К-байтового блока адресного пространства целевой системы в свободную область памяти C800 — CFFF машины Apple.

Плата ВСЭ предоставляет главной микроЭВМ возможность ограниченной эмуляции, позволяя ей перемещать содержимое любого 2К-байтного блока адресного пространства целевой системы в обычно свободную область памяти C800₁₆ — CFFF₁₆ микроЭВМ Apple II (рис. 11.2). Область памяти целевой системы, задаваемая для исследования, указывается программой, которая заносит соответствующую информацию в «регистр выбора». Так как это указание осуществляется под управлением программы, можно составить для главной системы такие тестовые программы, с помощью которых будет обеспечен контроль всей памяти целевой системы.

Тестовые программы могут быть написаны либо на языке высокого уровня, таком, как Бейсик, либо на машинном языке и предназначены для контроля основных функциональных блоков целевой системы, к числу которых относятся системные шины; ЗУПВ; ПЗУ; устройства ввода-вывода.

Программы на Бейсике, реализующие даже простые тесты, имеют относительно невысокую производительность. Разумный подход состоит в написании стандартных тестовых модулей на машинном языке и последующем использовании программы на Бейсике для формирования полной тестовой последовательности и указаний оператору при обнаружении некорректного функционирования.

В этой главе описываются тестовые модули для проверки системных шин, контроля ЗУПВ и ПЗУ, а в завершающей части главы рассматривается конкретный пример, иллюстрирующий применение ВСЭ Apple для тестирования типичной одноплатной микроЭВМ.

Аппаратные средства внутрисхемного эмулятора

На рис. 11.3 представлена принципиальная схема платы ВСЭ. Сигналы с адресных линий A0—A10, а также управляющие сигналы R/W, 00, \overline{RES} и SYNC поступают прямо из главной системы в целевую через ИС U4 и U5, содержащие по восемь усилителей формирователей сигналов. Адресные сигналы A11—A15 для целевой системы не формируются из своих аналогов, имеющих место в микроЭВМ Apple II, а вырабатываются регистром выбора блока U3. Для получения пяти старших разрядов адреса для целевой системы в этот регистр заносится управляющее слово, причем «стробирование» осуществляется сигналом «Задание ввода-вывода» ЭВМ Apple II. Таким образом, если ВСЭ размещен в области памяти 5, сигналам A11—A15 присваиваются нулевые значения с помощью следующей короткой программы:

```
LDA = 00 установка A11—A15 в состояние 0
STA C500 активизация линии ввода-вывода области 5
```

После занесения такой информации в регистр выбора блока при любой операции чтения или записи по адресам памяти Apple II, лежащим в диапазоне $C800_{16}$ — $CFFF_{16}$, приводится в действие дешифратор адреса U1 и подается разрешение на вывод информации из регистра U3, в результате чего формируется адрес памяти целевой системы, принадлежащий области 0000_{16} — $07FF_{16}$.

Дешифратор адреса U1 подает, кроме того, разрешение на восьмиразрядный приемопередатчик U2. Тому, кто знаком

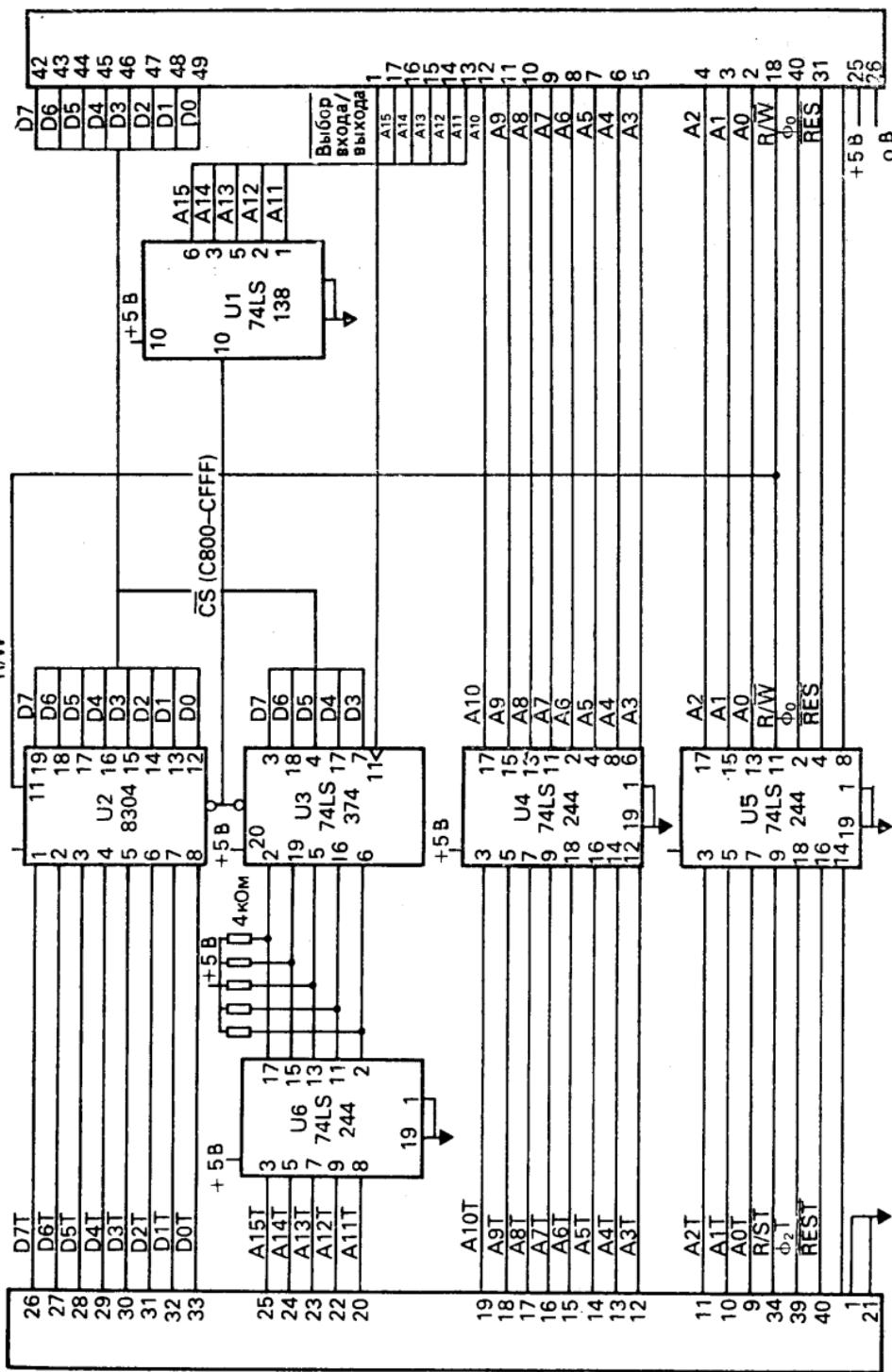


Рис. 11.3. Принципиальная электрическая схема внутрисхемного эмулятора Apple.

с разъемом ввода-вывода машины Apple, может показаться, что дешифратор U1 не нужен, потому что «строб ввода-вывода» имеет для адресов $C800_{16}$ — $CFFF_{16}$ низкий уровень. Однако более тщательное рассмотрение временных соотношений позволяет

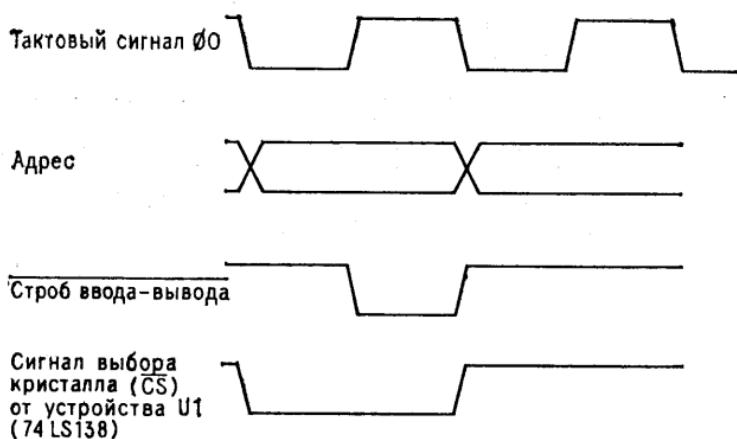


Рис. 11.4. Временные соотношения сигналов стробирования области ввода-вывода ЭВМ Apple и сигнала выбора кристалла от U1.

заметить, что он поступает слишком поздно, чтобы его можно было использовать для подачи разрешения на медленную память или устройства ввода-вывода целевой системы (рис. 11.4).

Тестовые программные средства

Тестовые программы подразделяются на две категории:

— программы, служащие для испытаний и контроля различных функциональных блоков и узлов целевой микроЭВМ, т. е. ее системных шин, ЗУПВ, ПЗУ и устройств ввода-вывода;

— общие программы контроля, служащие для направления действий оператора в процессе прохождения тестовых последовательностей, вызова тестов функциональных блоков и узлов и выполнения действий, обычно реализуемых с помощью «дерева поиска неисправностей», т. е. выяснения причины неправильного функционирования и выдачи рекомендации по устранению этой причины (например, «заменить ИС 28») или запуска нового теста для получения более полной информации.

Ниже описываются три программы функционального тестирования, написанные на языке ассемблера микропроцессора 6502. Каждая из этих программ размещается в «окне отображения памяти» ($C800_{16}$ — $CFFF_{16}$) между ЭВМ Apple II и целевой системой.

ИСХОДНЫЙ ФАЙЛ: APPTOG

0000 : 1 ; ****
0000 : 2 ; ТЕСТ ШИН АДРЕСА И ДАННЫХ
0000 : 3 ; ПЕРЕКЛЮЧАЕТ ИНФОРМАЦИЮ НА АДРЕСНОЙ
ШИНЕ AAAA-5555
0000 : 4 ; 256 РАЗ,
0000 : 5 ; ПЕРЕКЛЮЧАЕТ ИНФОРМАЦИЮ НА ШИНЕ
ДАННЫХ AA-55
0000 : 6 ; 256 РАЗ
0000 : 7 ; ****
0000 : 8
C500 : 9 SELECT EQU \$C500 ; ВЫБОР БЛОКА 2К
----NEXT OBJECT FILE NAME IS APPTOG OBJ0
2100 : 10 ORG \$2100
2100:A2 00 11 LDX #00 ; ОБНУЛЕНИЕ
; СЧЕТЧИКОВ
2102 : 12 ; ПРОВЕРКА АДРЕСНОЙ ШИНЫ
2102:A9 AA 13 ; ABUS LDA #\$AA ; ВЫБОР АДРЕСОВ
; 10101XXX
2104:8D 00 C5 14 ; STA SELECT
2107:AD AA CA 15 LDA \$CAAA ; РАЗР.ПРЕР.AAAA
; НА ЦЕЛЕВОЙ ШИНЕ
210A:A9 55 16 LDA #\$55 ; ВЫБОР АДРЕСОВ
; 01010XXX
210C:8D 00 C5 17 STA SELECT
210F:AD 55 CD 18 LDA \$CD55 ; РАЗР.ПРЕР.5555 НА
; ЦЕЛЕВОЙ ШИНЕ
2112:CA 19 DEX
2113:D0 ED 20 BNE ABUS ; ПОВТОРИТЬ 256 РАЗ
2115 : 21 ; ПРОВЕРКА ШИНЫ ДАННЫХ
2115:A9 00 22 DBUS LDA #\$00 ; ВЫБОР АДРЕСОВ
; 00000XXX
2117:8D 00 C5 23 STA SELECT
211A:A9 55 24 LDA #\$55 ; 01010101 НА ШИНЕ
; ДАННЫХ
211C:8D 00 C9 25 STA \$C900 ; 0100 В ЦЕЛЕВОЙ
; СИСТЕМЕ
211F:A9 AA 26 LDA #\$AA ; 10101010 НА ШИНЕ
; ДАННЫХ
2121:8D 00 C9 27 STA \$C900 ; 0100 В ЦЕЛЕВОЙ
; СИСТЕМЕ
2124:CA 28 DEX
2125:D0 EE 29 BNE DBUS ; ПОВТОРИТЬ 256 РАЗ
2127:60 30 RTX ; ПЕРЕКЛЮЧЕНИЕ
; ЗАКОНЧЕНО

*** SUCCESSFUL ASSEMBLY: NO ERRORS

2102 ABUS 2155 DBUS C500 SELECT
2115 DBUS C500 SELECT

2102 ABUS

Программа 11.1. Программа «переключения» всех линий шин адресов и
данных.

(а) Переключательный тест шин адресов и данных

Перед тем как приступить к выполнению сложных тестов по проверке интегральных узлов, целесообразно проверить работоспособность системных шин. «Переключательный тест» служит для испытания каждой линии шин адресов и данных путем попаременной подачи на линию сигналов высокого и низкого уровня. Этот тест представлен в виде программы 11.1. Он начинается с задания адресов памяти целевой системы (первым из которых является $10101xxxxxxxxxxxx_2$), после чего выполняется холостое чтение по адресу $AAAA_{16}$, при котором на адресную шину целевой системы подается набор битов с чередующимися единичными и нулевыми значениями. Затем снова производится установка содержимого регистра выбора, в результате которой оказываются выбранными адреса, начинающиеся с $01010xxxxxxxxxxxx_2$, после чего выполняется холостое чтение по адресу 555_{16} , т. е. по адресу, представляющему собой инверсию предыдущего. Описанная процедура повторяется 256 раз, а затем проводится аналогичный тест шины данных целевой системы.

Описанный тест позволяет оператору проверить с помощью осциллографа или логического пробника, что

- все линии целевой системы являются управляемыми, т. е. ни одна из них не сохраняет постоянного высокого или низкого уровня;

- каждая линия полностью исправна, т. е. соединяет контакт процессорного гнезда с соответствующими выводами всех нужных ИС.

Отсутствие коротких замыканий между линиями также можно проверить с помощью более сложной тестовой программы, подающей на каждую линию сигнал определенной частоты, т. е. специальную последовательность битов. Для выяснения, нет ли замыканий между линиями, в этом случае применяется частотомер или осциллограф.

Очевидно, при 256-кратном переключении системных шин оператору не хватает времени на визуальную проверку хотя бы одной точки тестируемой схемы. Приведенная ниже короткая программа иллюстрирует применение программы тестирования шин (BTEST) в составе контрольной программной последовательности.

```

180 PRINT "BUS TESTING — PROVE TARGET SYSTEM
        BUSES"
190 print "(PRESS SPACE FOR NEXT TEST)"
200 CALL BTEST
210 IF PEEK(-16384) <= 127 THEN 200

```

Если включить этот тест в цикл, в котором осуществляется также проверка клавиатуры, тест будет повторяться до тех пор,

пока оператор не подаст нажатием клавиши пробела сообщение, что он удовлетворен результатами просмотра.

(б) Тест ЗУПВ “шахматная доска”

Основной принцип контроля ЗУПВ заключается в записи в память некоторого тестового набора данных, считывании записанной информации и проверке правильности выполнения записи и чтения. Применяемые тестовые наборы могут быть самыми разнообразными, причем различные наборы могут быть чувствительны к разным видам неисправностей. Одним из наиболее универсальных способов контроля, позволяющим проверить чтение и запись в каждой битовой ячейке ЗУПВ, является тест «шахматная доска». Реализация некоторых более сложных тестов занимает несколько часов.

Программа 11.2 представляет собой программу тестирования ЗУПВ. Выбирается некоторая область ЗУПВ, в нее заносится число 55_{16} (01010101_2), затем содержимое ячейки считывается и сравнивается с исходным. Если сравнение дает отрицательный результат, выполнение теста прекращается и ЭВМ Apple II выдает сообщение о неисправности ЗУПВ. При успешном завершении сравнения те же операции повторяются с инверсным словом AA_{16} (10101010_2). Затем тест повторяется для следующей области памяти и т. д., пока не будут испытаны и проконтролированы все области с адресами, входящими в окно $C800_{16} — CFFF_{16}$.

Перед тем как обратиться из основной тестовой программы к программе контроля ЗУПВ, необходимо осуществить запись информации в регистр выбора с таким расчетом, чтобы разместить проверяемое ЗУПВ в тестовом окне внутрисхемного эмулятора. Например, следующий фрагмент программы определит задание для проверки областей ЗУПВ целевой системы с адресами в диапазоне $0800_{16} — 0FFF_{16}$.

```
250. PRINT "RAM TESTING 0800 — 0FFF"
260 POKE SELECT,08 : CALL RAMTEST
```

Сигнатуры ПЗУ

Широко распространенным методом контроля ПЗУ является формирование байтовой контрольной суммы, которая образуется путем сложения всех данных, хранящихся в ПЗУ. Однако при этом всегда существует вероятность того, что неисправности останутся невыявленными вследствие взаимной компенсации нескольких ошибок. Более точный метод, при использовании которого взаимная компенсация ошибок относительно маловероятна,

ИСХОДНЫЙ ФАЙЛ: APPRAM

```

0000:      1 ; *****
0000:      2 ; ПРОГРАММА КОНТРОЛЯ ЗУПВ "ШАХМАТНАЯ ДОСКА"
0000:      3 ; (C800-CFFF)
0000:      4 ;
0000:      5 ;
FDED:      6 COUT    EQU    $FDED   ; СИМВОЛ ДЛЯ ЭКРАНА
FD8E:      7 CROUT   EQU    $FD8E   ; С-ВОЗВРАТ К ЭКРАНУ
FDE3:      8 PRHEX   EQU    $FDE3   ; ВЫВОД ШЕСТНАДЦАТИЧНОЙ ЦИФРЫ
0008:      9 POINT   EQU    08     ; УКАЗАТЕЛЬ
0000:     10 ;
-----NEXT OBJECT FILE NAME IS APPRAM.OBJ0
2090:      11 ORG    $2090
2090:      12 ;
2090 : A9 00
2092 : 85 08
2094 : AB
2095 : A9 CB
2097 : 85 09
2099 : A9 55
209B : 91 08
209D : D1 08
209F : F0 03
20A1 : 4C BC 20
20A4 : A9 AA
20A6 : 91 08
20AB : D1 08
20AA : F0 03
20AC : 4C BC 20
20AF : E6 08
-----APPRAZM.OBJ0
0000:      1 ; *****
0000:      2 ; УКАЗАТЕЛЬ НА С800
0000:      3 ;
0000:      4 ;
0000:      5 ;
0000:      6 LDA    # 00    ; УКАЗАТЕЛЬ НА С800
0000:      7 STA    POINT
0000:      8 TAY
0000:      9 LDA    # $CB
0000:     10 STA    POINT + 1
0000:     11 LDA    # $55   ; НАЧАЛО ТЕСТА С 55
0000:     12 STA    (POINT),Y ; ЗАПОМИНАНИЕ
0000:     13 CMP    (POINT),Y ; ОБРАТНОЕ ЧТЕНИЕ И ИСПРАВЛЕНИЕ
0000:     14 BEQ    OK
0000:     15 JMP    ERROR ; СООБЩЕНИЕ ОБ ОШИБКЕ И КОНЕЦ
0000:     16 STA    # $AA ; ТЕПЕРЬ ПРОВЕРКА АА
0000:     17 CMP    (POINT),Y ; ЗАПОМИНАНИЕ
0000:     18 BEQ    OK1
0000:     19 STA    (POINT),Y ; ОБРАТНОЕ ЧТЕНИЕ И ИСПРАВЛЕНИЕ
0000:     20 CMP    (POINT),Y ; СООБЩЕНИЕ ОБ ОШИБКЕ И КОНЕЦ
0000:     21 BEQ    INC POINT ; СЛЕДУЮЩАЯ ЯЧЕЙКА
0000:     22 JMP    ERROR ; СООБЩЕНИЕ ОБ ОШИБКЕ И КОНЕЦ
0000:     23 STA    # $AA ; ТЕПЕРЬ ПРОВЕРКА АА
0000:     24 CMP    (POINT),Y ; ЗАПОМИНАНИЕ
0000:     25 BEQ    OK1
0000:     26 STA    (POINT),Y ; ОБРАТНОЕ ЧТЕНИЕ И ИСПРАВЛЕНИЕ
0000:     27 CMP    (POINT),Y ; СООБЩЕНИЕ ОБ ОШИБКЕ И КОНЕЦ
0000:     28 INC POINT

```

```

29      BNE      START
20B1 : D0 E6
30      INC      POINT + 1
20B3 : E6 09
31      LDA      POINT + 1
20B5 : A5 09
32      CMP      #$D0
20B7 : C9 D0      ; КОНЕЦ БЛОКА (CFFF)?
33      BNE      START      ; ТЕСТ ОКОНЧЕН
20B9 : D0 DE
34      RTS
20BB : 60
35      ;
36      ; ПРОГРАММА ИНДИКАЦИИ ОШИБОК
20BC :      ;
37      ; 
38      ERROR      LDX      #00      ; УКАЗАТЕЛЬ НА СООБЩЕНИЕ
20BC : A2 00
39      NEXT1      LDA      MESS,X      ; СООБЩЕНИЕ НА ЭКРАН
20BE : BD D5 20
40      JSR      COUT      ; ОЧЕРЕДНОЙ СИМВОЛ
20C1 : 20 ED FD
41      INX
42      CPX      #$0F      ; СООБЩЕНИЕ ОКОНЧЕНО?
43      BNE      NEXT1      ; АДРЕС ОШИБКИ НА ЭКРАН
20C4 : E8
44      LDA      09
45      SEC
46      SBC      #$C8      ; СООБЩЕНИЕ ОБ ОШИБКЕ
20CC : E9 C8
47      JSR      PRHEX      ; С-ВОЗВРАТ К ЭКРАНУ
20CE : 20 E3 ED
48      JSR      CROUT      ; И КОНЕЦ
20D1 : 20 BE FD
49      RTS
20D4 : 60
50      ; 
51      MESS      ASC      "      ; СООБЩЕНИЕ ОБ ОШИБКЕ
20D5 :      ;
20D5 : A0 C5 D2      ; СООБЩЕНИЕ ОБ ОШИБКЕ
20D8 : D2 CF D2
20DB : A0 CF CE
20DE : A0 D0 G1
20E1 : C7 C5 A0
*** SUCCESSFUL ASSEMBLE; NO ERRORS

```

Программа 11.2. Программа тестирования ЗУ ВП методом "шахматной доски".

```
1 ; ****
2 ; ПРОГРАММА ОЦЕНКИ СИГНАТУРЫ БЛОКА
3 ; 2К БАЙТ (C800-CFFF); КАЖДЫЙ БАЙТ ПРЕОБРАЗУЕТСЯ В
4 ; ПОСЛЕДОВАТЕЛЬНУЮ ФОРМУ (БИТО-БИТ);
5 ;
6 ;
7 ;
8 ;-----NEXT OBJECT FILE NAME IS APPSIG.OBJ0
2000: 9 ORG $2000 ; ЗАПОМИНАНИЕ ДЛЯ СУММИРОВАНИЯ
1900: 10 COUNT EQU $1900 ; МЛ.БАЙТ ТЕКУЩЕЙ СИГНАТУРЫ
1901: 11 SIGL EQU $1901 ; СТ.БАЙТ ТЕКУЩЕЙ СИГНАТУРЫ
1902: 12 SIGH EQU $1902 ; СЧЕТЧИК БАЙТОВ
0008: 13 POINT EQU $000B ; ВРЕМЕННОЕ ЗАПОМИНАНИЕ
1903: 14 TEMP EQU SIGH + 1 ; ПЕЧАТЬ БАЙТА
FDDA: 15 PRBYTE EQU $FDDA ; ШЕСТНАДЦАТИБИТНАЯ
;-----NEXT OBJECT FILE NAME IS APPSIG.OBJ0
FD8E: 16 CROUT EQU $FD8E ; ГЕНЕРАЦИЯ С -ВОЗВРАТА
2000: 17 ; ; НУЛЕВОЙ СДВИГ РЕГИСТРА
2000:A9 00 18 START LDA $00
2002:8D 01 19 STA SIGL
2005:8D 02 19 20 STA SIGH
2008:A9 00 21 WSTART LDA # 00 ; ТЕПЛИЙ ЗАПУСК
200A:85 08 22 STA POINT
200C:A8 23 TAY
200D:A9 C8 24 LDA # $C8 ; ЗАПУСК БЛОКА С 800
```

200F : 85 09 POINT +1
2011 : B1 08 (POINT), Y ; ПОЛУЧЕНИЕ БЛОКА
2013 : 8D 03 19 TEMP
2016 : A2 08 LDX #08 ; ДЛЯ 8 БИТ
2018 : AD 03 19 STA TEMP
201B : 29 01 AND #01 ; БИТ 0 В COUNT
2010 : 8D 00 19 STA COUNT
2020 : 20 36 20 FEEDBACK ; ПРИЛОЖЕНИЕ ОБРАТНОЙ СВЯЗИ
2023 : 6E 03 19 ROR TEMP ; ГОТОВНОСТЬ К СЛЕДУЮЩЕМУ БИТУ
2026 : CA DEX
2027 : DO EF NBIT ; ВОЗВРАТ ЗА СЛЕДУЮЩИМ БИТОМ
2029 : E6 08 POINT ; ОЧЕРЕДНОЙ БАЙТ
202B : D0 E4 BNE NBYTE
2020 : E6 09 INC POINT +1
202F : A5 09 LDA POINT +1
2031 : C9 D0 CMP #\$D0
2033 : D0 DC BNE NBYTE ; КОНЕЦ БЛОКА CFFFF)?
2035 : 60 RTS
2036 : 43 ;
2036 : 44 ;
2036 : 45 ; АЛГОРИТМ РАБОТЫ ОБРАТНОЙ СВЯЗИ —
2036 : 46 ; СУММИРОВАНИЕ БИТОВ 15,11,8 И 4 С ПОСТУПАЮЩИМ
2036 : 47 ; БИТОМ.ПРИ ВВОДЕ В COUNT ЗАНОСИТСЯ ВХОДНОЙ БИТ
2036 : 48 ;
2036 : AD 02 19 FEEDBACK LDA SIGH ; ВЕРХНЯЯ ПОЛОВИНА СИГНАТУРЫ

2039 : 10 03	50	BPL	NEX1	TEST-BIT 15
203B : EE 00 19	51	INC	COUNT	
203E : 6A	52	NEX1	ROR	A
203F : 90 03	53	BCC	NEX2	; TEST-BIT 8
2041 : EE 00 19	54	INC	COUNT	
2044 : 6A	55	NEX2	ROR	A
2045 : 6A	56	ROR	A	
2046 : 6A	57	ROR	A	
2047 : 90 03	58	BCC	NEX3	; TEST-BIT 11
2049 : EE 00 19	59	INC	COUNT	
204C : AD 01 19	60	NEX3	LDA	SIGL
204F : 2A	61	ROL	A	; НИЖНЯЯ ПОЛОВИНА СИГНАТУРЫ
2050 : 2A	62	ROL	A	
2051 : 90 03	63	BCC	NEX4	; TEST-BIT 6
2053 : EE 00 19	64	INC	COUNT	
2056 : 6E 00 19	65	NEX4	ROR	COUNT
2059 : 2E 01 19	66	ROL	SIGL	; СУММА В ТРИГТЕР ПЕРЕНОСА
205C : 2E 02 19	67	ROL	SIGH	; ПЕРЕНОС В БИТ 0 МЛ.БАЙТА
205F : 60	68	RTS		; ПЕРЕНОС В БИТ 0 СТ.БАЙТА
2060 : AD 02 19	69	DISPLAY	LDA	SIGH
2063 : 20 DA FD	70	JSR	PRBYTE	; СТ.БАЙТ НА ИНДИКАЦИЮ
2066 : AD 01 19	71	LDA	SIGL	; НА ДИСПЛЕЙ APPLE
2069 : 20 DA FD	72	JSR	PRBYTE	; МЛ.БАЙТ НА ИНДИКАЦИЮ
206C : 20 8E FD	73	JSR	CROUT	; НА ДИСПЛЕЙ APPLE
206F : 60	74	RTS		; С-ВОЗВРАТ

опирается на использование циклических избыточных кодов содержащего ПЗУ. Этот метод контроля был первоначально использован в системах передачи данных, но за последнее время нашел широкое применение как основа сигнатурного анализа. Контроль с использованием циклической избыточности может быть реализован и аппаратно, и программно, но объяснить его принцип, по-видимому, проще, пользуясь аппаратной моделью.

На рис. 9.4 представлена схема 16-разрядного сдвигового регистра с обратными связями. Все биты данных последовательно вводятся в регистр. Двоичное слово, формирующееся в регистре по окончании потока данных, представляет собой четырехразрядный шестнадцатеричный циклический избыточный код. Благодаря цепям обратной связи на вход регистра подается сумма по модулю 2 данных, поступающих по этим цепям, и новых данных, вводимых в схему; тем самым обеспечивается участие каждого бита содержимого проверяемого ПЗУ в формировании окончательного циклического избыточного кода, т. е. сигнатуры.

Программа 11.3 выполняет те же действия. Каждый байт тестируемого ПЗУ последовательно (начиная с бита 0 и кончая битом 7) воспринимается подпрограммой FEEDBACK, которая осуществляет суммирование по модулю 2 битов из разрядов 6, 8, 11 и 15 регистра с поступающим битом. После обработки подпрограммой FEEDBACK всех $2K \times 8$ бит данных в областях SIGH и SIGL формируется окончательная циклическая избыточная сумма, т. е. сигнатура.

Для контроля ПЗУ с объемом более $2K$ байт используются три стандартные программы.

Название программы	Выполняемые действия
NSIG (Новая сигнатура)	Устанавливает в нулевое состояние «сдвиговый регистр», состоящий из SIGH и SIGL, и формирует сигнатуру $2K$ -байтового ЗУПВ
CSIG (Продолжение сигнатурь)	Делает то же, что и предыдущая программа, но не выполняет начального сброса «сдвигового регистра», в результате чего продолжается формирование сигнатуры для ПЗУ, превышающего по объему $2K$ байт
DISPLAY	Выводит на индикацию содержимое «сдвигового регистра», т. е. областей SIGH и SIGL в шестнадцатеричном формате

Приводимая ниже программа иллюстрирует использование всех трех названных выше для формирования сигнатуры ПЗУ объемом 4К байт, размещенного в диапазоне адресов памяти целевой системы $B000_{16}$ — $BFFF_{16}$.

```

320 REM BO HEX IS 176 DECIMAL
330 POKE SELECT, 176
340 CALL NSIG : REM FIRST 2K BYTES
350 REM B8 HEX IS 184 DECIMAL
360 POKE SELECT, 184
370 CALL CSIG : REM CONTINUE WITH NEXT 2K BYTES
380 CALL DISPLAY:REM DISPLAY FINAL SIGNATURE

```

Пример реализации тестовой программы:

Внутрисхемный эмулятор Apple может быть применен для работы со многими микроЭВМ, построенными на основе микропроцессорного комплекта 6500 и имеющими рабочую частоту 1 МГц, при условии, что все схемы на плате эмулятора тактируются процессорным синхросигналом Ø2. Идеальным примером целевой системы может служить микроЭВМ AIM 65, в состав которой входят ЗУПВ емкостью 4К байт, 20К-байтовое ПЗУ и широкий набор устройств ввода-вывода: два универсальных интерфейсных адаптера 6522, адаптер параллельного интерфейса 6520 и адаптер чтения с терминала ввода-вывода 6532. Краткое описание микроЭВМ и карта распределения ее памяти приведены в приложении 2.

Тестовая программная последовательность представлена в виде программы 11.4. Она начинается с контроля системных шин, после которого выполняется тестирование 4 К-байтового

```

47 REM ****
48 REM * APPLESOFT PROGRAM SEQUENCING TESTS *
49 REM ****
50 REM AIM65 TEST ROUTINE
60 HOME
70 REM DEFINE SYSTEM ADDRESSES
80 SELECT == 15100
90 DISPLAY = 8288
100 NSIG = 8192
110 CSIG = 8200
120 BTTEST = 8448
130 RAMTEST = 8336
140 PRINT " "
150 PRINT "LOADING MACHINE CODE TESTS"
160 PRINT "BLOAD APPTESTS"
170 PRINT " "

```

```

180 PRINT "BUS TESTING-PROBE TARGET SYSTEM BUSES"
190 PRINT "(PRESS SPACE FOR NEXT TEST)"
200 CALL BTTEST
210 IF PEEK (-16384) < = 127 THEN 200
220 PRINT ""
230 PRINT "RAM TESTING 0000-07FF"
240 POKE SELECT,0:CALL RAMTEST
250 PRINT "RAM TESTING 0800-0FFF"
260 POKE SELECT,08:CALL RAMTEST
270 PRINT ' "; CHR$(13); "RAM TESTS COMPLETE"
280 PRINT ""
290 PRINT "ROM SIGNATURES BLOCKS B,C,D,E,F"
300 PRINT ""
310 FOR N=176 TO 240 STEP 16
320 POKE SELECT,N:CALL NSIG
330 POKE SELECT,(N+8):CALL CSIG
340 CALL DISPLAY
350 NEXT N
360 PRINT " "; CHR$(13); " ROM SIGNATURES COMPLETE"
370 PRINT ""
380 PRINT " VIA TEST"
390 POKE SELECK,160:REM SELECT BLOCK AXXX
400 APRT=51201:BPRT=51200
405 ADIR=51203:BDIR=51202
410 POKE ADIR,0:POKE BDIR,255
415 REM A INPUT — B OUTRUT
420 FOR N=0 TO 225
430 POKE BPRT,N
440 IF PEEK(APRT)<>N THEN PRINT "VIA ERROR"
450 NEXT N
460 POKE BDIR,0:POKE ADIR,255
465 REM BINPUT — A OUTPUT
470 FOR N=0 TO 255
480 POKE APRT,N
490 IF PEEK(BPRT)<>N THEN PRINT " VIA ERROR"
500 NEXT N
510 PRINT " "; PRINT "TEST COMPLETE"
520 END

```

Программа 11.4. Программа организации последовательности тестирования в системе Applesoft.

ЗУПВ и тестирование ПЗУ с формированием сигнатур для каждого из пяти 4К-байтовых блоков. Завершает программу тест пользовательского универсального адаптера 6522. Для выполнения этого теста порты адаптера соединяются между собой

с помощью колодки, в которой проводным монтажом выполнены соединения A0—B0, A1—B1 и т. д. Подпрограмма начинается с назначения порта А портом ввода, а порта В — портом вывода. Затем производятся запись в порт В «нарастающих» тестовых наборов данных и проверка данных, считываемых из порта А. Затем порты меняются ролями и весь тест повторяется (рис. 11.5).

```
JRUN
LOADING MACHINE CODE TESTS
BUS TESTING-PROBE TARGET SYSTEM BUSES
(PRESS SPACE FOR NEXT TEST)
RAM TESTING 0000-07FF
RAM TESTING 0800-0FFF
RAM TESTS COMPLETE
ROM SIGNATURES BLOCKS B,C,D,E,F
B89C
A181
F727
B072
8ABE
ROM SIGNATURES COMPLETE
UIA TEST
TEST COMPLETE
```

Рис. 11.5. Типичный вид результатов тестирования исправной платы.

Этот пример иллюстрирует некоторые приемы использования внутрисхемного эмулятора Apple. В программе более глубокого контроля имело бы место тестирование интерфейсных устройств индикатора, принтера, клавиатуры и т. д., а также присутствовали бы более подробные указания для оператора. Однако основные идеи, заложенные в рассмотренной программе, помогают освоению принципов внутрисхемной эмуляции и дают представление о возможностях серийно выпускаемых приборов. Отмеченные обстоятельства позволяют считать ВСЭ Apple не только практическим инструментом поиска неисправностей, но и удобным недорогим средством обучения.

Заключение

Стоимость контрольного оборудования может в гораздо большей степени определяться стоимостью индикаторов, устройств ввода и управления, чем стоимостью тех электронных устройств,

которые собственно и обеспечивают поиск неисправностей. Возлагая функции индикации, ввода и управления на микроЭВМ, можно свести объем специализированных средств контроля к одной недорогой плате. В рассмотренном примере для управления платой ВСЭ при тестировании ПЗУ и ЗУПВ целевой системы применены программные средства. Описанная система эмуляции является не только практическим средством непосредственного контроля, но и удобным объектом изучения методов ВСЭ.

Глава 12

Одноплатные анализаторы

Сигнатурный анализатор

Серийно выпускаемые приборы, реализующие функцию сигнатурного анализа, могут иметь и другие функциональные возможности, как, например, измерение частоты, подсчет событий, логический анализ. Это делает их универсальными, но в то же время сложными и дорогими тестовыми установками. Однако дешевый прибор, предназначенный только для сигнатурного анализа, может быть конструктивно оформлен в виде небольшой платы. При наличии дешифраторов индикации, вырабатывающих изображения букв А, С, F, Н, Р и У вместо обычно применяемых шестнадцатеричных символов А, В, С, D, Е и F, можно сделать прибор автономным. Привлекательным является и вариант, когда в распоряжении имеется недорогая микроЭВМ, в которую можно вводить сигнатурные данные, программным путем преобразовывать их и представлять на встроенном индикаторе или дисплее; такой подход весьма целесообразен при использовании ЭВМ для автоматизации поиска неисправностей.

Рассматриваемый здесь одноплатный анализатор по своим характеристикам во многом аналогичен серийным приборам; его построение предусматривает возможность сопряжения с микроЭВМ. Однако для упрощения процесса тестирования исходная конфигурация включает четыре светодиодных цифровых индикатора и интегральный дешифратор TIL308; поэтому устройство функционирует как автономный сигнатурный анализатор. Дешифратор TIL308 преобразует четырехразрядные двоичные числа в изображения символов. При подаче соответствующего сигнала биты данных могут быть зафиксированы в индикаторе. К сожалению, индицируемый набор символов не совпадает ни с традиционным набором шестнадцатеричных цифр, ни с набором символов, применяемых в сигнатурном анализе. Поэтому при сравнении сигнатур, получаемых с помощью данного устройства и серийных приборов, необходимо пользоваться таблицей соответствия. При составлении описания новой системы с помощью указанного устройства это обстоятельство не является серьезным недостатком. Набор входных управляющих сигналов

устройства ограничен сигналами пуска, останова и синхронизации и еще тремя сигналами, которые могут быть заданы переключателями, определяющими, какие фронты сигналов пуска, останова и синхронизации являются активными. Специфической особенностью входных сигналов пуска и останова является то, что должна быть возможность подачи их в виде одного и того же сигнала тестируемого устройства и обеспечения корректной работы анализатора при любом заданном сочетании положительных и отрицательных фронтов.

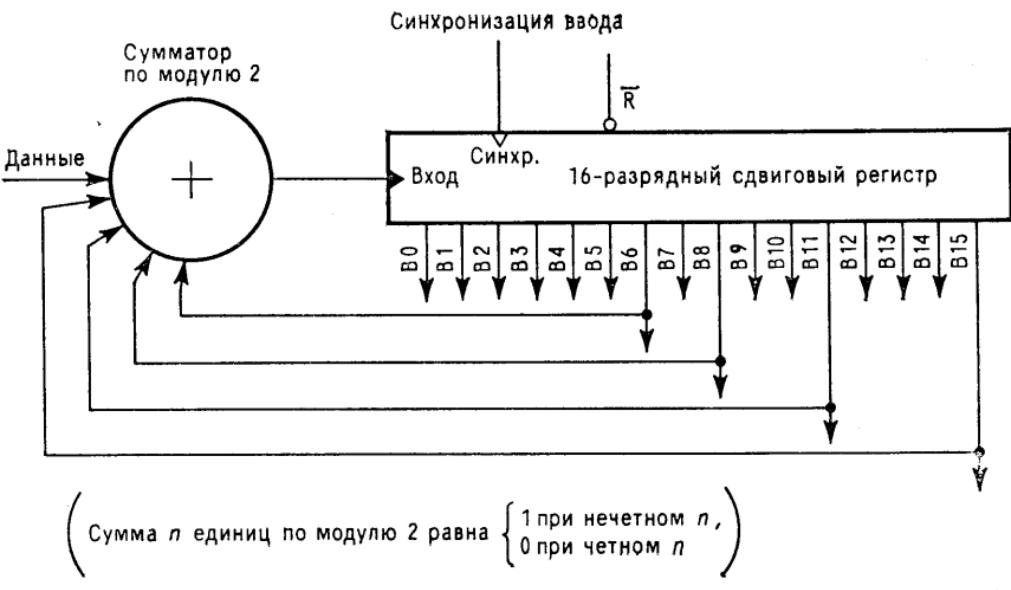


Рис. 12.1. Основной блок сигнатурного анализатора.

Схема сигнатурного анализа, имеющаяся в составе рассматриваемого контрольного устройства, состоит из 16-разрядного сдвигового регистра и сумматора по модулю 2, на входы которого поступают сигналы обратной связи и входные данные для сигнатурного анализа. Сигналы обратной связи берутся с разрядов 6, 8, 11 и 15 сдвигового регистра, причем последовательный ввод данных в регистр осуществляется со стороны младшего бита, имеющего номер 0. Данные, фиксируемые в регистре после останова ввода, представляют собой сигнатуру.

Для реализации этой схемы используются две ИС 74LSI64, каждая из которых имеет 8-разрядный сдвиговый регистр, и ИС, содержащая четыре вентиля ИСКЛЮЧАЮЩЕЕ ИЛИ, из которых организуется сумматор по модулю 2 (рис. 12.1).

В соответствии с порядком работы анализатора сдвиговый регистр предварительно устанавливается в нулевое состояние, и до поступления сигнала пуска синхросигналы на него подаваться не должны. После открытия сигнатурного окна синхросигналы

начинают поступать на регистр, и из входных данных формируется их сигнатура. По окончании окна находящиеся в регистре данные могут быть зафиксированы в памяти индикатора (или в 8-разрядных регистрах, если используется интерфейс ЭВМ). Схема устанавливается в исходное состояние и готова к поступлению очередного пускового сигнала.

Совершенно обязательно, чтобы данные, поступающие одновременно с фронтом останова, не вводились в сдвиговый регистр, так как этот бит данных, будучи лишним, исказит сигнатуру.

Удобно пояснить эту ситуацию на примере формирования сигнатур 16-разрядного счетчика, представленном на рис. 12.2. (Де-

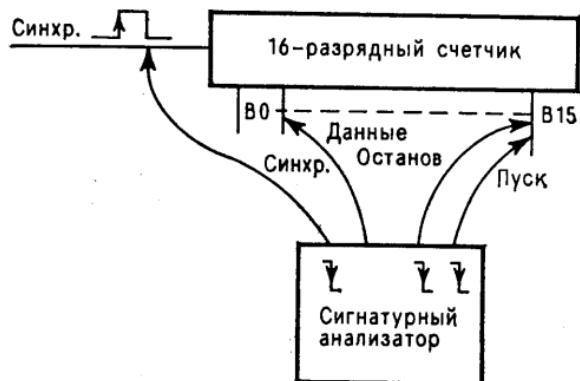


Рис. 12.2. Использование сигнатурного анализатора для исследования работы 16-разрядного счетчика.

ло в том, что сигнатуры адресных шин большинства 8-разрядных микропроцессоров при тестировании их в режиме свободного прогона совпадают с сигнатурами 16-разрядного счетчика.)

Если изменение содержимого счетчика производится по положительному фронту синхросигнала, то в момент поступления отрицательного фронта содержимое счетчика имеет установившееся значение. Полный цикл работы счетчика требует подачи 64 К синхросигналов, а отрицательный фронт 15-го разряда счетчика может быть использован в качестве сигналов и пуска, и останова в процессе регистрации сигнатур.

Таким образом, первым битом данных, записываемым в сдвиговый регистр, должен быть тот, который поступает при нахождении всех разрядов счетчика в нулевом состоянии, а последний бит должен вводиться, когда на выходах всех разрядов счетчика имеет место единичный сигнал. Это означает, что сдвиг по синхросигналу, поступающему после того, как 15-й разряд счетчика изменит свое состояние с единичного на нулевое впервые после поступления сигнала пуска, не должен иметь места. Выше-

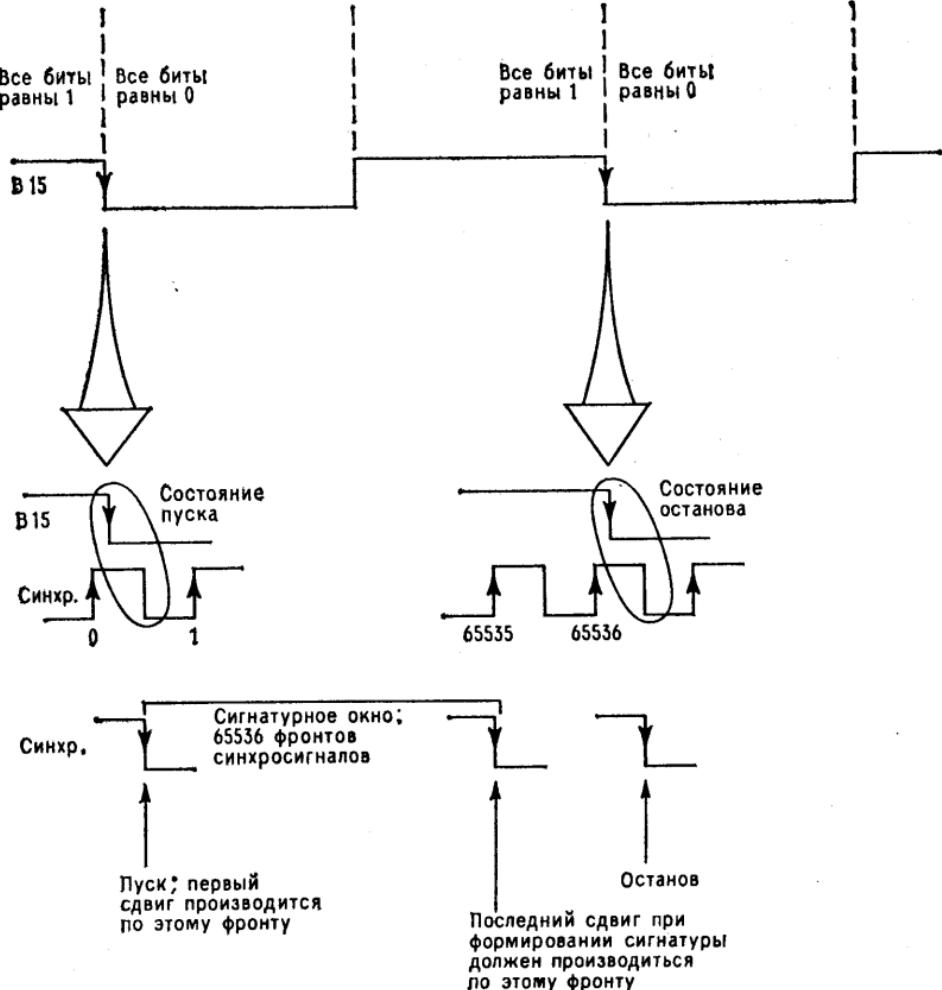


Рис. 12.3. Временные диаграммы к схеме на рис. 12.2.

изложенное иллюстрируется временными диаграммами, приведенными на рис. 12.3.

Рассматриваемая проблема решается в описываемом устройстве путем введения небольшой задержки между синхросигналом тестируемого устройства и сигналом, по которому в регистре осуществляется сдвиг. Задержка позволяет схеме пуска/останова изменять состояние и тем самым разрешать или запрещать поступление синхросигнала на управление сдвигом регистра перед поступлением активного фронта синхросигнала на вход приема данных.

Таким образом, чтобы был обеспечен ввод в анализатор корректной информации, вводимые данные должны фиксироваться незадержанными синхросигналами. С учетом этих соображений

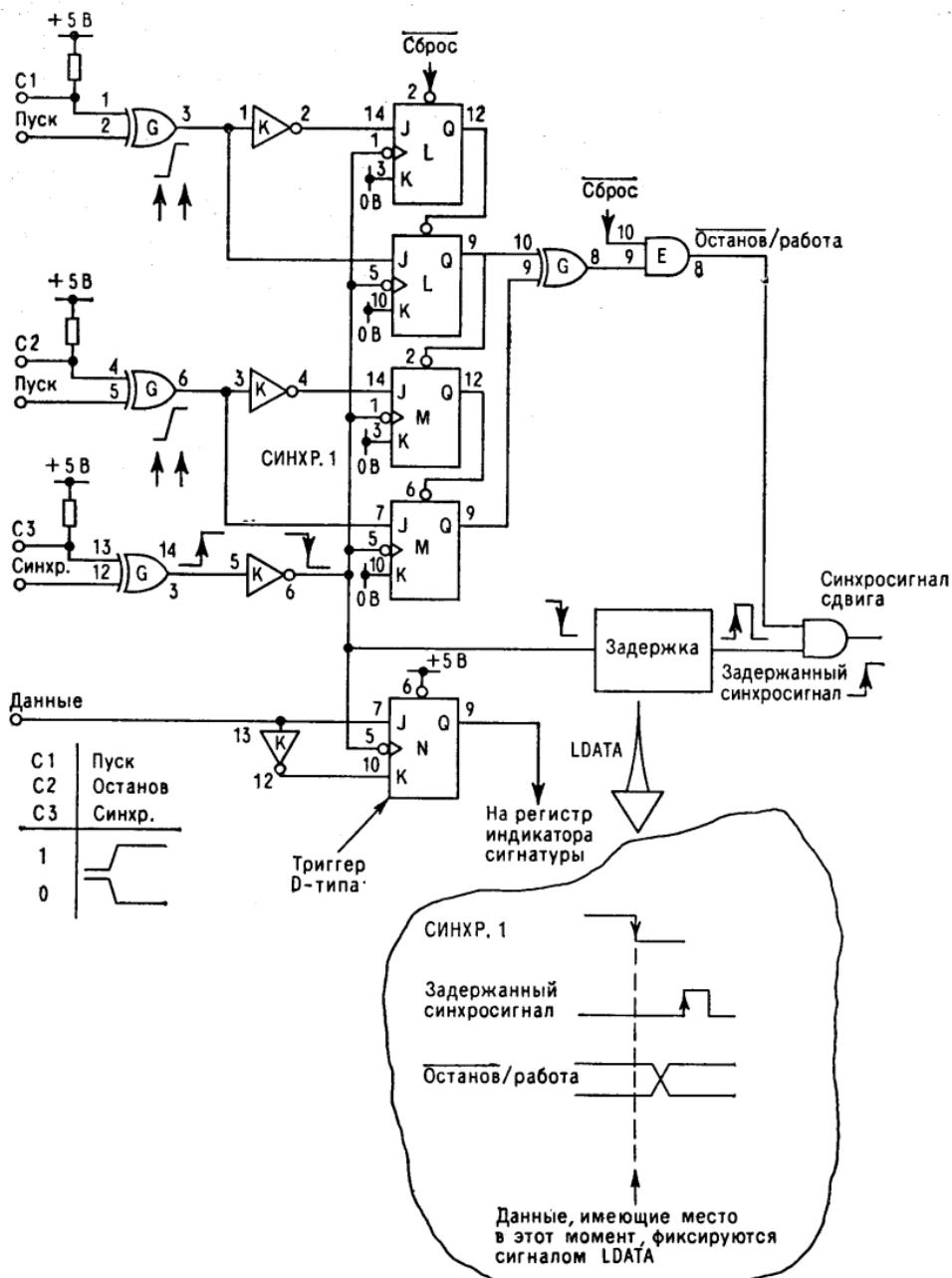


Рис. 12.4. Схема пуска/останова сигнатурного анализатора.

в схеме используются триггеры типа D, реализованные путем соответствующего включения триггеров JK, как показано на рис. 12.4. Триггеры JK применены, потому что требуемое для надежной записи время удержания данных на их входах меньше.

чем у триггеров типа D. Введение задержки ограничивает максимальную рабочую частоту величиной около 8 МГц. Этого вполне достаточно для тестирования 8-разрядных систем, а также для целей обучения и подготовки персонала.

На рис. 12.4 показана также схема пуска/останова, построенная на JK-триггерах, тактируемых незадержанными синхросигналами. Такая синхронизация необходима, так как у некоторых микропроцессоров значения сигналов на адресных линиях не остаются неизменными на протяжении всего процессорного цикла. Если при этом использовать сигнал A15 для создания сигналов пуска и останова сигнатурного анализатора, окажется неправильно сформированным сигнатурное окно. Применение схемы, синхронизирующей сигналы пуска и останова сигналами тактирования объекта контроля, обеспечивает восприятие схемой анализатора только тех данных, которые имеют место в момент изменения состояния синхросигналов. Рабочие фронты сигналов пуска, останова и синхронизации задаются с помощью вентиляй ИСКЛЮЧАЮЩЕЕ ИЛИ в качестве управляемых инверторов. Для управления этими вентилями могут быть использованы однополюсные переключатели (или логические сигналы, вырабатываемые интерфейсом микроЭВМ).

Сигнал останова не оказывает своего действия, пока не будут установлены в единичное состояние два триггера, связанные с сигналом пуска. Оба сигнала, для того чтобы они правильно выполняли свои функции, должны пребывать в состоянии, предшествующем их срабатыванию, не меньше, чем в течение прохождения активного фронта хотя бы одного синхросигнала. На протяжении всей длительности сигнатурного окна имеет высокий уровень сигнал RUN (Работа); правильная регистрация данных в сигнатурном анализаторе обеспечивается за счет использования задержанных синхросигналов. Отрицательным фронтом сигнала RUN запускаются две схемы формирования одиночных импульсов, одна из которых вырабатывает импульс начальной установки схемы пуска/останова, а вторая инициирует последовательность импульсов, которыми данные из сдвигового регистра фиксируются в памяти индикаторов и/или в 8-разрядных регистрах, используемых в качестве интерфейса ЭВМ. После соответствующей задержки подается также сигнал начальной установки сдвигового регистра. Этот сигнал может быть использован в качестве сигнала прерывания для микроЭВМ, если декодирование и индикация сигнатур осуществляются с ее применением.

Сигнал начальной установки, обозначенный как \bar{R} на рис. П4.1, полной электрической схемы анализатора имеет достаточную длительность для того, чтобы исключить мерцание индикаторов и избежать слишком частых прерываний в случае работы анализатора под управлением ЭВМ.

Логический анализатор

Использование микроЭВМ для обработки данных, получаемых с помощью недорогого логического анализатора, даже более привлекательно, чем применение ЭВМ для совместной работы с сигнатурным анализатором. Объем информации, которая должна быть получена при работе с логическим анализатором, существенно больше, и она наглядно изображается на экране монитора. Может понадобиться представить ее в шестнадцатеричном или двоичном формате, а иногда в формате ASCII; применение микроЭВМ для обработки и индикации информации позволяет уменьшить объем оборудования, которое должно быть предусмотрено для этих целей в составе одноплатового прибора.

Приводимое ниже описание схемы имеет общий характер, конкретные устройства отдельно не рассматриваются; дается лишь представление о принципах построения типичных анализаторов.

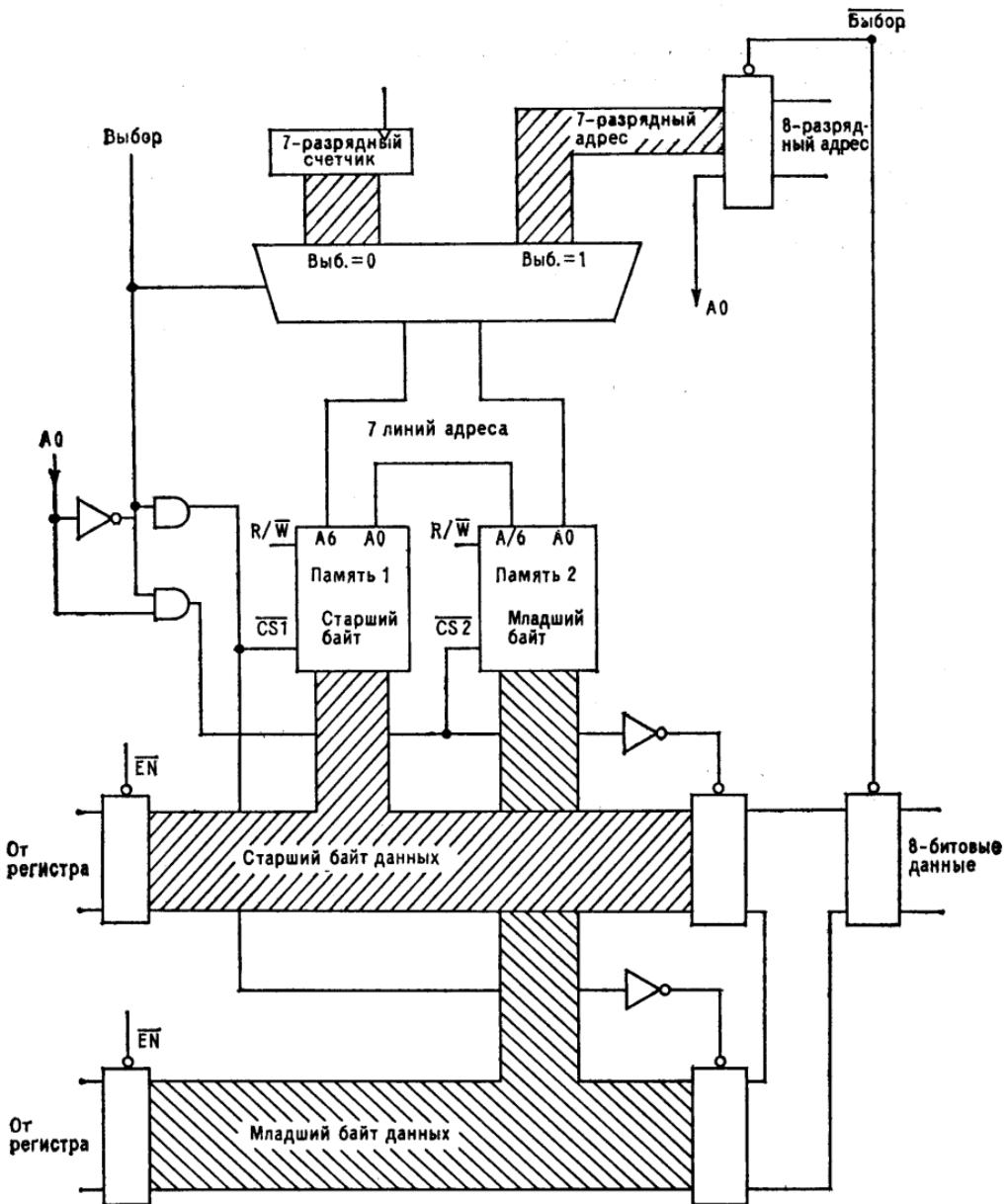
Центральным устройством анализатора является небольшое ЗУПВ, которое позволяет избежать необходимости занесения данных в память ЭВМ на стадии регистрации. То обстоятельство, что данные, поступающие до запускающего события, запоминанию не подвергаются, позволило упростить рассматриваемую схему.

Для адресации памяти в процессе записи регистрируемых данных служит счетчик. Когда его содержимое достигает максимального значения, запись останавливается и микроЭВМ получает возможность обращаться к памяти анализатора.

В схеме анализатора, представленной на рис. П4.1, применена 8-разрядная память. Шестнадцатибитовые данные записываются в два блока памяти. Когда ЭВМ читает информацию из памяти анализатора, схема выбора обеспечивает попеременное чтение байтов из этих блоков. Эта организация иллюстрируется рис. 12.5.

Хотя практически осуществляется регистрация 8-битовых данных, целесообразно иметь возможность работы с 16-битовыми запускающими словами. Для этого служат четыре 4-разрядные схемы сравнения, подключенные к линиям данных и к дополнительным линиям запуска. Выходные сигналы этих схем сравнения поступают на входы вентиля И, выходной сигнал которого является сигналом запуска. Группы по 4 бит могут быть выборочно заблокированы с помощью показанных на схеме переключателей, при этом на соответствующий вход вентиля И подается сигнал 1.

Для увеличения разрядности запускающего слова могут быть введены дополнительные схемы сравнения; при необходимости в схему может быть добавлен квалифиликатор запуска. Схема этого



Примечания: 1) сохраняются только 128 единиц 16-разрядных данных
2) данные читаются в последовательности $L_1 H_1 L_2 H_2 \dots$ (сначала младший, а затем старший байт)

Рис. 12.5. Вариант построения адресно-логической схемы 16-разрядного аналогоизмерителя.

квалификатора имела бы такой же вид, как схема квалификатора синхросигнала, а ее выходной сигнал Т подавался бы на вентиль И, выходной сигнал которого являлся бы сигналом запуска, обработанным квалификатором.

Если анализатор был подготовлен к работе, вводимые данные фиксируются во входных регистрах синхросигналом, обработанным квалификатором, и загружаются в первую область памяти. Положительный фронт сигнала R/ \bar{W} задержан относительно фронта входного синхросигнала в соответствии с временем прохождения информации из входных регистров в память.

Приращение счетчика адресов не производится, пока не возникает условие запуска. Когда это условие появляется, синхросигнал пропускается еще через один элемент задержки на вход этого счетчика и осуществляет инкремент счетчика. При заполнении счетчика сигнал его выходного переноса используется для установки триггера, который отключает выходы разрядов счетчика от линий адреса памяти, после чего к этим линиям получает доступ микроЭВМ.

Инверсный выход триггера, вырабатывающего сигнал «Данные готовы», служит для придания высокого уровня сигналу R/W, с тем чтобы на то время, пока микроЭВМ читает данные из памяти анализатора, запись данных в эту память производиться не могла.

Заключение

Выше были рассмотрены одноплатные анализаторы: сигнальный анализатор, в котором применен недорогой индикатор и используется упрощенный принцип запуска, и обобщенный вариант логического анализатора. Акцент сделан на основополагающие принципы организации подобных устройств; однако пользователю достаточно приведенной информации для того, чтобы разрабатывать и применять аналогичные устройства.

Глава 13

Средства обмена информацией

Интерфейсный преобразователь

Разнообразие стандартов на интерфейсы, используемые для последовательной передачи данных, может быть причиной трудностей поиска неисправностей. Может оказаться, что контрольное оборудование построено таким образом, что допускает использование только сигналов, соответствующих стандарту RS232, и делает невозможным работу с интерфейсом «токовая петля».

В тех случаях, когда плата интерфейса конструктивно отделена от тестируемой платы микропроцессорной системы, может оказаться удобно вырабатывать интерфейсные сигналы для управления дисплеем или принтером непосредственно из выходных логических сигналов и обойтись при этом без платы, в которой могут быть неисправности. Описываемое ниже недорогое устройство предназначено для использования во всех ситуациях, когда необходимо преобразование уровней сигналов, причем для его функционирования не нужны никакие другие питающие напряжения, кроме 5 В. Устройство может быть применено для стыковки интерфейсных схем токовой петли и схем, соответствующих стандарту RS232 (или RS423), для выработки сигналов, соответствующих обоим этим стандартам из логических сигналов, а также для обратного преобразования. В устройстве предусмотрены некоторые дополнительные возможности, причем из соображений снижения стоимости и сложности их реализация осуществляется с помощью перемычек, а не переключателей. Схема устройства приведена на рис. 13.1. Некоторые способы его применения рассматриваются ниже.

Преобразование сигналов интерфейсов RS232 и RS423

Для выработки сигналов RS232 необходим источник питания с отрицательным напряжением. Хотя обычно для этих выходных сигналов нужны напряжения ± 12 В, можно при коротких линиях связи обойтись питанием ± 5 В. Для интерфейса RS423 питающие напряжения 5 В являются стандартными. Отрицательное

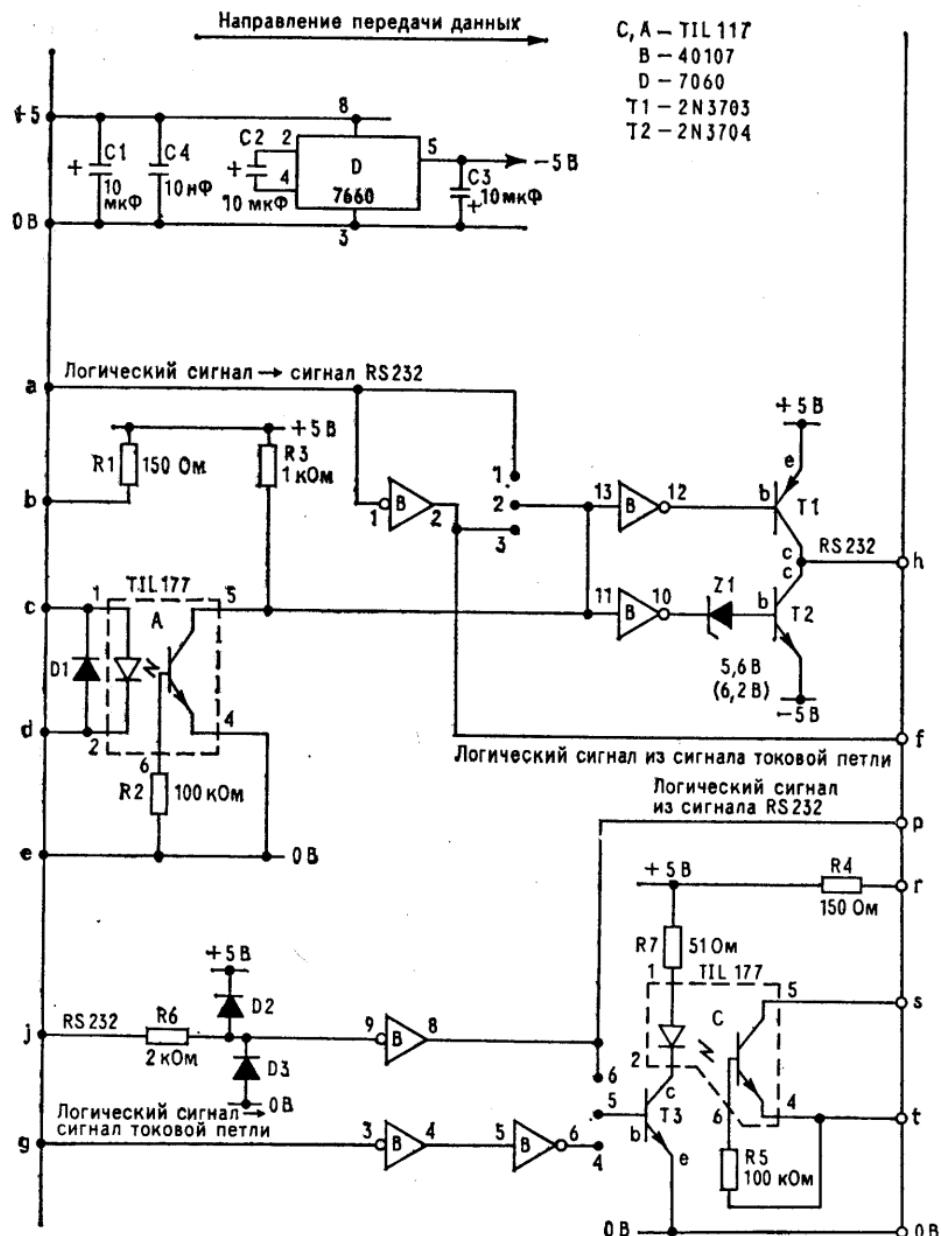


Рис. 13.1. Универсальное устройство преобразования уровней.

напряжение питания 5 В можно получить из напряжения +5 В, применяя диодную схему накачки или интегральный преобразователь, такой, например, как 7660. На рис. 13.2 показаны схемы преобразователей логических уровней в уровнях стандарта RS232; в качестве входных цепей использованы КМОП-вентили с диодными фиксаторами, предохраняющими схему от действия

уровней сигналов стандарта RS232. Существуют специализированные устройства сопряжения, такие, как 75152 и 75150, обычно применяемые для выполнения таких преобразований. Схема

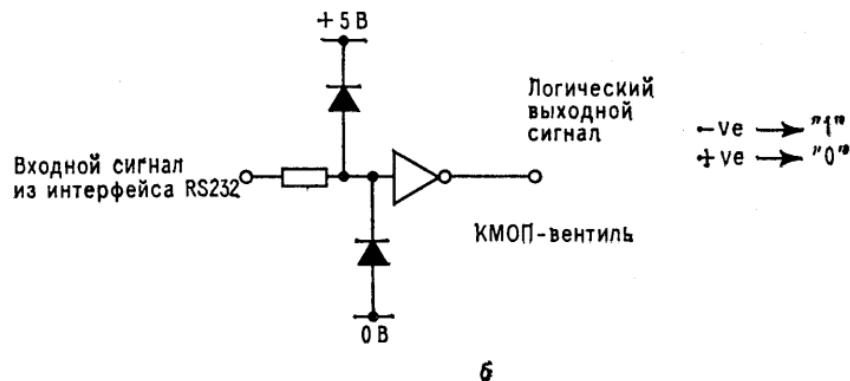
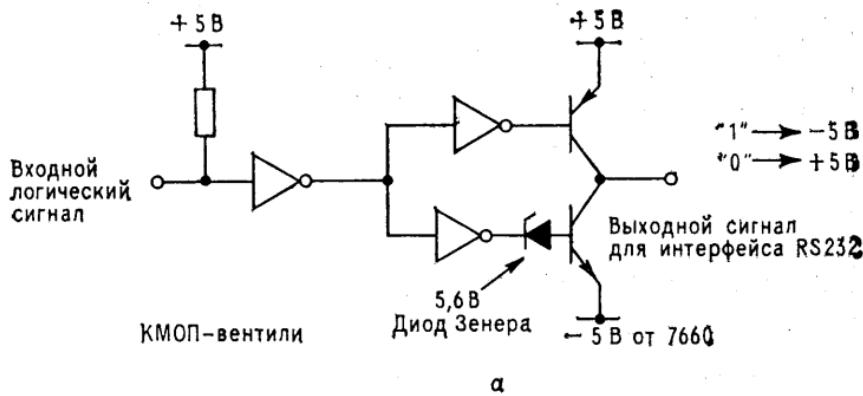


Рис. 13.2. Преобразование логических сигналов в сигналы интерфейса RS232:
а — логический сигнал → сигнал RS232; б — сигнал RS232 → логический сигнал.

же, представленная на рис. 13.2, иллюстрирует реализацию преобразования уровней с помощью обычных дискретных компонентов в ситуации, когда интегральных устройств под рукой не оказывается.

Преобразование сигналов токовой петли

В приемниках схем с токовой петлей обычно применяются оптические пары (оптраны), как показано на рис. 13.3. Необходимо грамотно выбрать коллекторное сопротивление R_c , так как при слишком большой его величине снижаются частотные качества схемы. Величина 1 кОм препятствует насыщению входного транзистора и в то же время обеспечивает достаточно низкое

значение коллекторного напряжения при прямых токах диода в диапазоне 15—22 мА. Входной диод оптрана служит для защиты от случайных напряжений обратной полярности. Передатчик токовой петли, показанный на рис. 13.1, также снабжен

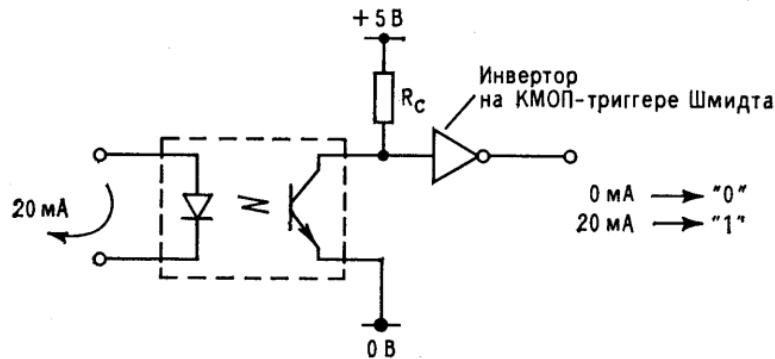


Рис. 13.3. Преобразование сигналов токовой петли в логические сигналы.

оптраном, который обеспечивает развязку в передающей части схемы. Источник тока в этом случае должен находиться в приемной части.

Интегральные конвертеры, такие, как HCPL—4100 и HCPL—4200, могут выполнять преобразование логических сиг-

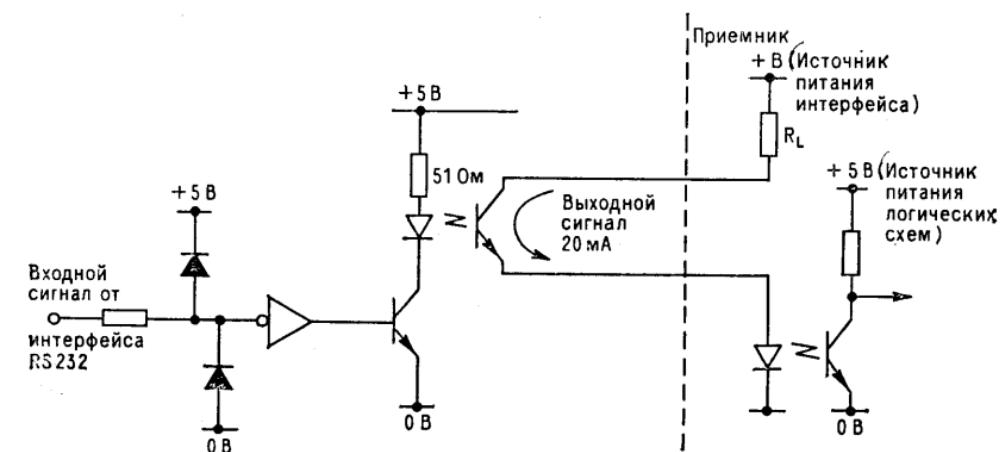


Рис. 13.4. Преобразование сигналов RS232 в сигналы токовой петли при размещении источника тока в приемнике.

налов в сигналы токовой петли и обратное преобразование с использованием минимального числа дополнительных компонентов. Однако падения напряжения в приемнике и передатчике составляют при этом около 2,4 В, в то время как в описанной выше схеме эти величины составляют примерно 1,6 В для приемника и почти 0 В для передатчика.

Таблица 13.1. Варианты конфигурации преобразователя сигналов

Вид преобразования	Компоненты	Перемычки	Вход	Выход	Примечания
Логика → RS232	B, D, T1, T2, Z1, R6	2—3, ab	a	h	R3 должно быть изъято
RS232 → Логика	D2, D3, C2, C3, C1 C4, R1 = 100 кОм	g — 0 В	j	R	
Токовая петля → Логика	R1, R2, R3, D1, R5, R7, A, B, C, R4, T3	1—2, de 1—2	bc	f	Активный режим
Логика → Токовая петля	C1, C4, D3 = 100 кОм RES	5—4, t-0 В 5—4	g	rs	Пассивный режим
Токовая петля → RS232	Все компоненты g соединены с 0 В	d — e, 1—2 1—2	bc	st	Активный режим
RS232 → Токовая петля		6—5, t-0 В 6—5	j	h	Пассивный режим
			j	rs	Активный режим
			j	st	Пассивный режим

На рис. 13.4 показана схема преобразования сигналов RS232 в сигналы токовой петли, а данные табл. 13.1 показывают, каким образом должны быть использованы компоненты и установлены перемычки при различных применениях описанного устройства.

Приставка к логическому анализатору для обработки последовательных данных

Известен ряд недорогих логических анализаторов, которые могут обрабатывать информацию, поступающую по нескольким каналам, число которых может достигать 16. С помощью входной платы, выполняющей преобразование последовательных данных в параллельные и подающей в анализатор по окончании каждого преобразования синхросигнал, можно расширить функциональные возможности анализатора поиском неисправностей в системах последовательной передачи информации.

Рассматриваемое здесь устройство такого типа может быть использовано совместно с анализаторами, имеющими не менее восьми каналов, которые могут запускаться словом, включающим информацию с не менее чем восемь младших входных каналов. Конкретный анализатор, фигурирующий в описываемом примере, снабжен установленным на задней стенке разъемом

для контрольного кабеля в резиновой изоляции, через который выводится наружу напряжение питания +5 В. Хотя этот отвод и не предназначен для питания внешних устройств, его мощности достаточно для питания последовательно-параллельного преобразователя, если из соображений минимального энергопотребления он реализован на КМОП-логических элементах (прототипная схема потребляет около 12 мА).

Задание скорости передачи и формата данных, обычно выполняемое в анализаторах последовательных данных с помощью

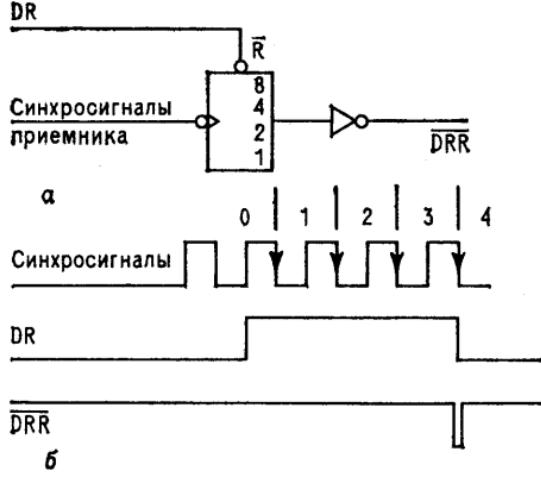


Рис. 13.5. Применение двоичного счетчика для выработки импульса сброса сигнала готовности данных в УАПП 6402: а — схема, б — временные диаграммы.

клавиш и индикатора, в данном случае осуществляется переключателями. Наличие генератора управления скоростью передачи, который вырабатывает все необходимые частоты одновременно, является достоинством схемы, так как для управления его выходами нужен лишь задающий переключатель.

Универсальный асинхронный приемопередатчик 6402 является автономным устройством, и поэтому его легче стыковать с логическими устройствами на дискретных компонентах, чем шинно-ориентированные программируемые приборы. Этот УАПП выдает сигнал готовности данных (DR) при получении последовательного кода символа. Сигнал может быть использован для синхронизации ввода данных в логический анализатор. Он должен быть сброшен отрицательным импульсом \overline{DDR} (сброс сигнала готовности данных), вырабатываемым пересчетной схемой, которая удерживается в сброшенном состоянии, пока сигнал DR имеет низкий уровень. Пересчетная схема показана на рис. 13.5; ее использование для задержки сигнала более предпочтительно, чем схемы на емкостях, потому что импульс \overline{DDR} всегда появ-

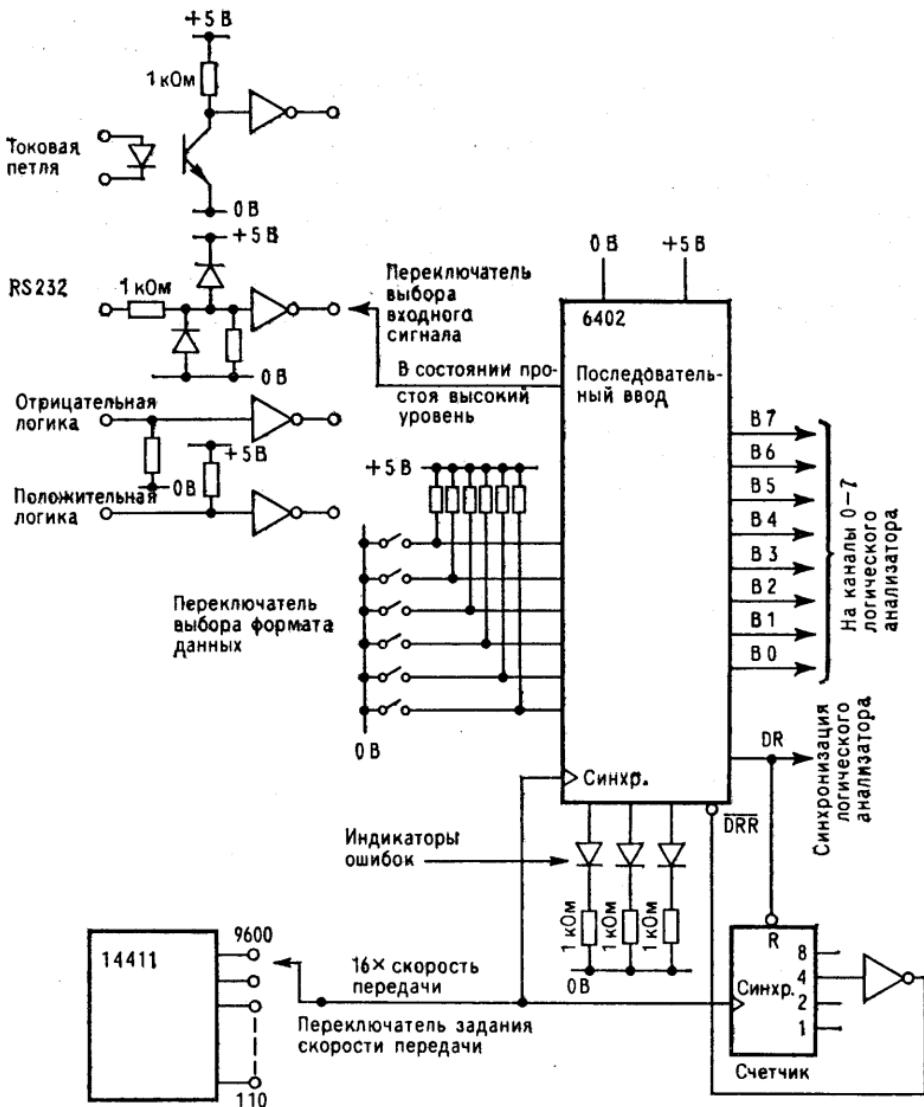


Рис. 13.6. Приставка ввода последовательных данных в логический анализатор.

ляется на ее выходе после прохождения строго определенного числа синхросигналов, определяющих скорость передачи. В качестве генератора синхросигналов используется MC14411, который настроен на генерацию синхросигналов с частотой, в 16 раз превышающей необходимую частоту тактирования передачи, как это и требуется для УАПП 6402. Полная функциональная схема приставки последовательной передачи приведена на рис. 13.6. Входной задающий переключатель позволяет использовать анализатор для обработки сигналов четырех различных уровней.

Настройка логического анализатора производится путем задания необходимого запускающего слова, т. е. шестнадцатеричного значения некоторого символа ASCII. Переключатели всех квалифицикаторов запуска и синхронизации находятся в выключенном состоянии; загрузка данных в анализатор производится после того, как вход последовательного ввода подключается к исследуемой точке и заданы ожидаемый формат слова и скорость передачи. Наблюдаемый последовательный выход приводится в активное состояние, и после занесения в память требуемого количества слов на индикацию выводятся первые зарегистрированные данные. Этот процесс может воспроизводиться повторно; последовательно просматривая при его повторении тестируемую схему по направлению от источника сигнала к интерфейсу, можно выяснить, где имеет место неисправность.

Описанное устройство, конечно, может быть использовано непосредственно с микроЭВМ при наличии соответствующей программы для анализа битов данных, вырабатываемых приемопередатчиком, выполнения необходимых проверок и записи данных в память в интервале между поступлениями соседних импульсов. При скорости передачи 1200 бод на один бит приходится примерно 830 мкс, и если для представления символа используется 10 бит, то времени оказывается вполне достаточно, чтобы обработка данных могла быть реализована с помощью программы на машинном языке. После ввода необходимого количества символов микроЭВМ может вывести информацию, находящуюся в буферной области, на индикацию программой на языке высокого уровня, так как время выполнения программы в данном случае значения не имеет. Наиболее подходящим набором символов для индикации является в данном случае совокупность принятых символов ASCII, дополненная некоторыми мнемоническими обозначениями для представления управляющих символов. Причинами значительной части «неисправностей» систем последовательной передачи данных являются ошибки, скрытые от глаза при передаче управляющих символов, что делает весьма целесообразным применение логических анализаторов последовательных данных.

Генератор последовательных символов

Описываемая ниже система построена на базе рассмотренного в этой книге одноплатного контроллера Z80. Основные характеристики системы определяются программными средствами, необходимость в дополнительном оборудовании сведена при этом к минимуму.

Рассматриваемая здесь программа написана применительно к скорости передачи 1200 бод с целью упрощения ее представле-

Глава 13. Средства обмена информацией

ния и может быть легко приспособлена к любой другой скорости передачи. В порты ввода-вывода считывается информация, задаваемая переключателями, и из них выводятся на описанную выше интерфейсную плату последовательные данные. Имеется возможность реализации четырех функций, задаваемых с помощью четырехпозиционного переключателя:

- 1) тест U^*U^* ;
- 2) подача набора символов в возрастающей последовательности;
- 3) тест «пробел!» (короткий тест линии);
- 4) тест с выборочной подачей символов.

Структурная схема системы представлена на рис. 13.7. Восьмь переключателей, подключенных к порту В, используются

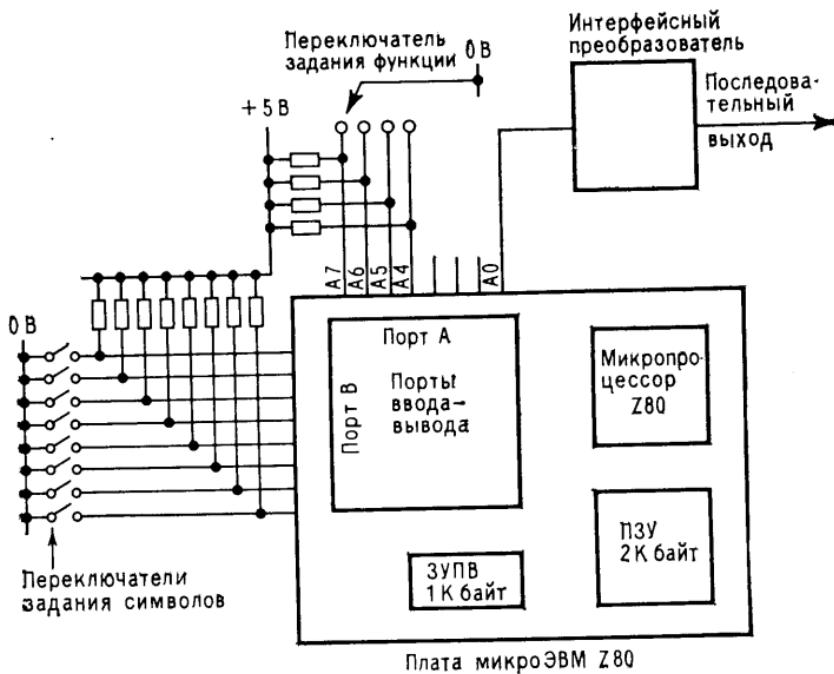


Рис. 13.7. Генератор последовательных символов для контроля интерфейса.

при реализации функции 4. Программа осуществляет настройку портов, опрашивает состояние переключателей и производит вызов подпрограммы в соответствии с заданной переключателями функцией, которая и реализуется, пока не изменится состояние переключателей. Схемы алгоритмов основной программы и используемых при ее работе подпрограмм представлены на рис. 13.8. Листинг приведен в виде программы 13.1. Функциональные возможности системы могут быть расширены за счет ее

ФАЙЛ: SERTSTZ: pLM
ОБЛАСТЬ ОБЪЕКТ-
ПАМЯТИ НОМЕР
КОД СТРОКИ

HEWLETT-PACKARD: АССЕМБЛЕР 780
ИСХОДНЫЙ ТЕКСТ

1 "Z80" 2 EQU ; MHZ
(0002) 2 CLK EQU ; ((10000 * CLK)/12 - 125)/24 + 1
(0041) 3 OBTIM EQU
(0078) 4 STPTIM EQU
(0000) 5 PORTA EQU
(0001) 6 PORTB EQU
(0002) 7 CNTRLA EQU
(0003) 8 CNTRLB EQU
(00CF) 9 MODE3 EQU
(83FF) 10 SWSAV EQU
11
MAIN:
12 LD SP,83FFH
0000 3183FF 13 LD A,MODE3
0003 3ECF 14 OUT [CNTRLA],A
0005 D302 15 LD A,11110000B
0007 3EF0 16 OUT [CNTRLA],A
0009 D302 17
18
000B 3E01 19 LD A,01
000D D300 20 OUT [PORTA],A
21
000F 3ECF 22 LD A,MODE3
0011 D303 23 OUT [CNTRLB],A
0013 3EFF 24 LD A,1111111B
0015 D303 25 OUT [CNTRLB],A
26
START: 27

0017	1600	28	LD D,00
0019	CD00BE	29	CALL SWCHK
9*	001C	30	B4:
001C	CB62	31	BIT 4,D
001E	2003	32	JR NZ,B5
0020	C30038	33	JP ROUT1
0023		34	
0023	CB6A	35	BIT 5,D
0025	2003	36	JR NZ,B6
0027	C3004E	37	JP ROUT2
002A		38	
002A	CB72	39	BIT 6,D
002C	2003	40	JR NZ,B7
002E	C30067	41	JR ROUT3
0031		42	
0031	CB7A	43	BIT 7,D
0033	20E2	44	JR NZ,START
0035	C30079	45	JP ROUT4
0038		46	
0038	CD00E9	47	ROUT1:
003B	0627	48	CALL CRLF
003D		49	LD B,39
003D		50	R11:
003D	2100F0	51	LD HL,MESS1
0040	CD00A0	52	CALL PRMESS
0043	10F8	53	DJNZ R11
0045	CD008E	54	CALL SWCHK
0048	DA0038	55	JP C,ROUT1
004B	C30017	56	JP START
		57	U*.....БЫВОД

004E	CD00E9	58	ROUT2:	CALL CRLF
0045	1E20	59		LD E,20H
0051	4B	60	R21:	LD C,E
0053	C000AC	61		CALL SEROP
0054	1C	62		INC E
0057	7B	63		LD A,E
0058	FE6E	64		CP 6EH
0059	C20053	65		JP NZ,R21
005B	CD008E	66		CALL SWCHK
005E	DA004E	67		JP C,ROUT2
0061	C30017	68		JP START
0064		69		
0067		70		
0067	CD00E9	71	ROUT3:	CALL CRLF
006A	2100F6	72		LD HL,MESS
006A	CD00A0	73		CALL PRMESS
006D	CD008E	74	R31:	CALL SWCHK
0070	DA006A	75		JP C,R31
0073	C30017	76		JP START
0076		77		
0079		78		
0079	CD00E9	79	ROUT4:	CALL CRLF
007C	DB01	80		IN A,[PORTB]
007C	4F	81		LD C,A
007E	CD00AC	82		CALL SEROP
007F	CD008E	83		CALL SWCHK
0082	DA007C	84	R41:	JP C,R41
0085	C30017	85		CALL CRLF
0088	CD00E9	86		JP START
008B		87		
008B		88		
008B		89		
008B		90		

ВЫВОД НАБОРА СИМВОЛОВ
НА ПЕРЕКЛЮЧАТЕЛИ ПОРТА В

92 ; ЭТА ПРОГРАММА СЧИТЫВАЕТ СОСТОЯНИЕ ВЫКЛЮЧАТЕЛЕЙ И,
93 ; ЕСЛИ ОНО НЕИЗМЕННО, ПРОИСХОДИТ ВОЗВРАТ С УСТАНОВКОЙ ФЛАГА
94 ; ПЕРЕНОСА. ЕСЛИ ИЗМЕНЕНИЕ — СБРОС ФЛАГА ПЕРЕНОСА

95
96 SWCHK:
97 IN A,[PORTA] ; ЧТЕНИЕ ПЕРЕКЛЮЧАТЕЛЕЙ
98 AND 11110000B ; ТОЛЬКО СТАРШИЙ ПОЛУБАЙТ
99 CP D
100 SCF ; УСТАНОВКА ПЕРЕНОСА
101 RET Z
102
103 SWCHK0:
104 LD D,A ; ПЕРЕКЛЮЧАТЕЛИ — В D
105 IN A,[PORTA] ; ЧТЕНИЕ ПЕРЕКЛЮЧАТЕЛЕЙ
106 AND 11110000B
107 CP D
108 JP NZ,SWCHK0
109 OR A
110 RET ; СБРОС ПЕРЕНОСА
111
112
113
114 PRMESS:
115 LD A,[HL]
116 CP EM
117 RET Z
118 LD C,A
119 CALL SEROP
120 INC HL
121 JP PRMESS
122
123
00A0
00A0 7E
00A1 FE04
00A3 C8
00A4 4F
00A5 CD00AC
00A8 23
00A9 C300A0

ПРОГРАММА ПЕЧАТИ
СООБЩЕНИЯ

00CO 3807
00C2 00C2 DB00
00C2 DB00
00C4 E6FE
00C6 C300CE
00C9 00
00C9 00
00CA DB00
00CC F601
00CE 05
00CF F200B5
00D2 DB00
00D4 F601
00D6 D300
00D8 110078
00DB CD00E2
00DE D1
00DF C1
00E0 FB
00E1 C9
00E2 00E2
00E3 00E3
00E4 00E4
00E5 00E5
00E6 00E6
00E7 00E7
00E8 00E8
00E9 00E9
00EA 00EA
00EB 00EB
00EC 00EC
00ED 00ED
00EE 00EE
00EF 00EF
00F0 00F0
00F1 00F1
00F2 00F2
00F3 00F3
00F4 00F4
00F5 00F5
00F6 00F6
00F7 00F7
00F8 00F8
00F9 00F9
00FA 00FA
00FB 00FB
00FC 00FC
00FD 00FD
00FE 00FE
00FF 00FF
147 SET0:
148 SET0:
149 SET0:
150 SET0:
151 SET0:
152 SET0:
153 SET0:
154 SET0:
155 SET0:
156 SET0:
157 SET0:
158 SET0:
159 SET0:
160 SET0:
161 SET0:
162 SET0:
163 SET0:
164 SET0:
165 SET0:
166 SET0:
167 SET0:
168 SET0:
169 SET0:
170 SET0:
J# C, SET1
IN A,[PORTA]
AND 1111110B ; УСТАНОВКА ВО В 0
JP CONT
NOP ; ДЛЯ ВЫРАВНИВАНИЯ ВРЕМЕНИ
; В ДВУХ ТРАКТАХ
IN A,[PORTA]
OR 000000001B ; УСТАНОВКА ВО В 1
DEC B
JP P,LOOP
IN A,[PORTA]
OR 000000001B ; УСТАНОВКА СТОПОВОГО БИТА
OUT [PORTA],A ; ВЫВОД БИТА
LD DE,STPTIM
CALL DELAY ; ОЖИДАНИЕ
POP DE
POP BC
EI
RET

124 : ПРОГРАММА ПОСЛЕДОВАТЕЛЬНОГО ВЫВОДА Z80

125 ; ИСПОЛЬЗОВАНИЕ БИТА 0 ПОРТА 0

126 ; СОСТОЯНИЕ ПРОСТОЯ = 1

127 ; СИМВОЛ, ПОМЕЩАЕМЫЙ В С ИСТОЧНИКОМ ЗАПРОСА

128 ; ЛИНИЯ ДОЛЖНА БЫТЬ УСТАНОВЛЕНА В 1, КОГДА ПРОИСХОДИТ ВЫЗОВ

129 ; ЗАНЕСЕНИЕ СИМВОЛА В РЕГИСТР С ИСТОЧНИКОМ ЗАПРОСА

130

00AC 131 SEROP: DI

00AC F3 132 PUSH BC

00AD C5 133 PUSH DE

00AE D5 134

135

00AF 0608 136 LD B,08

00B1 DB00 137 IN A,[PORTA] ; ЧТЕНИЕ ПОРТА ДЛЯ СОХРАНЕНИЯ

00B3 E6FE 138 AND 1111110B ; ПРИСВОЕНИЕ О СТАРТОВОМУ БИТУ

139

LOOP: OUT [PORTA],A ; ВЫВОД БИТА

140 LD DE,OBTIM ; ЗАДЕРЖКА БИТА В DE

141 CALL DELAY ; ОЖИДАНИЕ НА ЭТО ВРЕМЯ

142

143

00BD 79 144 LD A,C ; ВЗЯТИЕ СИМВОЛА ИЗ С

00BE 0F 145 RRCA ; МЛАДШИЙ БИТ В ТРИГГЕР ПЕРЕНОСА

00BF 4F 146 LD C,A ; ЗАПОМИНАНИЕ СДВИНУТОГО

; СИМВОЛА В С

171

; ПРОГРАММА ЗАДЕРЖКИ

172

; ДЕКРЕМЕНТ DE ДО НУЛЯ

173

174 ; В DE ДОЛЖНА БЫТЬ ЗАГРУЖЕНА НЕОБХОДИМНАЯ ЗАДЕРЖКА

175 **DELAY:**

00E2	1B	DEC DE
00E3	7A	LD A,D
00E4	B3	OR E
00E5	C200E2	JP NZ,DELAY
00E8	C9	RET

181

182 CRLF:

00E9	2100F3	LD HL,MESS2
00EC	CD00A0	CALL PRMESS
00EF	C9	RET

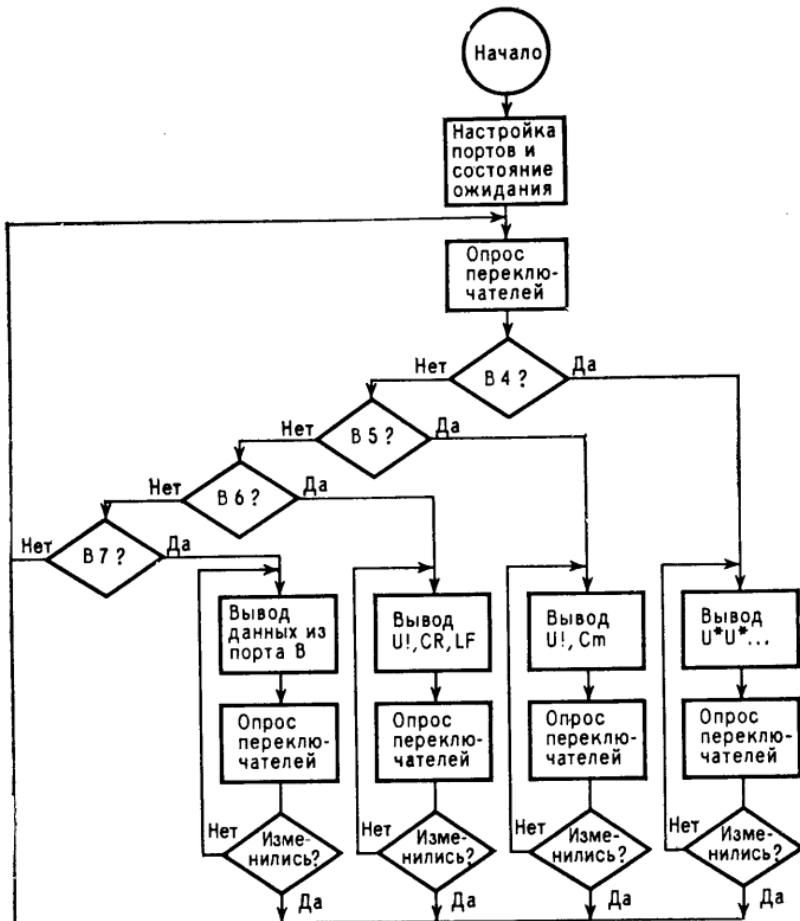
186

187 CR

000D	EQU 0DH
000A	EQU 0AH
0004	EQU 04
00F0 552A04	MESS1: DEF B 'U*', EM
00F3 0D0A04	MESS2: DEF B CR, LF, EM
00F6 20210D0A04	MESS4: DEF B 'P', CR, LF, EM

ПРОГРАММА
ЗАДЕРЖКИ

Программа 13.1. Программа генерации последовательных символов для однолатной микроЭВМ Z80.



6



8

Рис. 13.8. Схема алгоритма программы генерации символов.

адаптации к различным скоростям передачи данных путем введения переключателя, состояние которого считывается так же, как и состояние переключателей задания функций.

Заключение

Логический анализатор предназначен для работы с параллельными данными, в то время как обмен информацией между микроЭВМ и периферийными устройствами часто осуществляется в последовательной форме. В связи с этим были рассмотрены средства взаимного преобразования различных видов представления последовательных данных, перевода данных из последовательной формы в параллельную для индикации их логическим анализатором, генерации тестовых наборов данных в последовательной форме.

Приложения

Приложение 1 Сравнительные характеристики некоторых микропроцессоров

Сведения о микропроцессорах

Здесь приводится описание микропроцессоров 6502, 8085 и Z80, которые фигурируют в рассматриваемых в книге примерах. Приводятся некоторые данные и о микропроцессорах 6800, NSC800 и 8086.

Приводимые сведения почерпнуты из паспортов соответствующих изделий, но упорядочены по определенной схеме. Входные и выходные сигналы сгруппированы по категориям в соответствии с их назначением, причем подчеркнуты общие свойства различных микропроцессоров. Этот подход отличается от обычно используемого изготовителями, каждый из которых стремится подчеркнуть специфические особенности своего микропроцессора.

Сигналы объединены в следующие группы:

1. Сигналы управления процессором.
2. Сигналы управления пересылкой данных.
3. Сигналы шин адресов и данных.
4. Сигналы состояния процессора.
5. Сигналы управления прерываниями.
6. Напряжение питания.
7. Сигналы тактирования.
8. Другие сигналы, служащие, как правило, для реализации специфичных функций.

1. Сигналы управления процессором. Служат для останова или снижения скорости работы процессора. К их числу относится, например, сигнал RESET (Начальная установка), который обеспечивает перевод процессора в некоторое известное состояние.

2. Сигналы управления пересылкой данных. Управляют передачей данных между процессором, с одной стороны, и памятью и устройствами ввода-вывода — с другой. Это, например, сигналы READ и WRITE.

3. Сигналы шин адресов и данных. На адресные линии по даются значения адресов, задаваемые процессором в качестве указания устройств, являющихся источниками или местами на-

значения передачи данных. Линии адресов являются выходными линиями микропроцессора и, значит, не должны управляться от других устройств, если процессор не имеет возможности отключиться от них, приведя свои соответствующие выходы в состояние высокого импеданса («третье состояние»).

Линии данных являются двунаправленными и служат для передачи данных в процессор и из него. Линии адресов и данных отнесены к одной группе, потому что во многих процессорах шина адреса используется для пересылки данных путем временного мультиплексирования, как это делается в 8085, NSC800 и в большинстве 16-разрядных процессоров. В небольших микропроцессорных системах, где линии передачи старших разрядов адреса не используются полностью, неисправности этих линий не сказываются на работоспособности системы.

4. Сигналы состояния процессора. Они представляют собой информацию о внутреннем функционировании микропроцессора и иногда могут использоваться в качестве сигналов управления устройствами системы. Они позволяют разработчику микропроцессорной системы связать ее с контрольной аппаратурой для получения более полного представления о работе системы, как это, например, имеет место при сигнатурном анализе. Сигналы этой группы, которые в пределах конкретной системы не используются, могут в определенных ситуациях быть некорректными, не приводя к неправильному функционированию системы в целом; например, выходная цепь схемы выработки какого-то сигнала может выйти из строя и соответствующий сигнал состояния будет иметь при этом постоянный низкий уровень, но микропроцессор по-прежнему будет выполнять свои функции.

5. Сигналы управления прерываниями. Они используются для прерывания выполнения микропроцессором обычной последовательности команд, когда этого требуют какие-то внешние по отношению к микропроцессору события. Некоторые из сигналов этой группы несут информацию для устройства, инициирующего прерывание, о том, как воспринять его запрос. Если действие сигнала прерывания не разрешено программой, он вообще не оказывает никакого действия и, значит, будучи заблокирован, может иметь любое значение, не сказываясь на работоспособности системы. Другие сигналы прерываний всегда пребывают в рабочем состоянии, и если микропроцессор постоянно должен не реагировать на них, чтобы не прерывать исполнение необходимой программы, им необходимо придать пассивные значения; примером таких сигналов может служить NMI («Немаскируемое прерывание»), который всегда воспринимается процессором.

6. Напряжения питания. Для большинства современных микропроцессоров требуется одно питающее напряжение, но для некоторых известных типов аппаратуры, в которых, например,

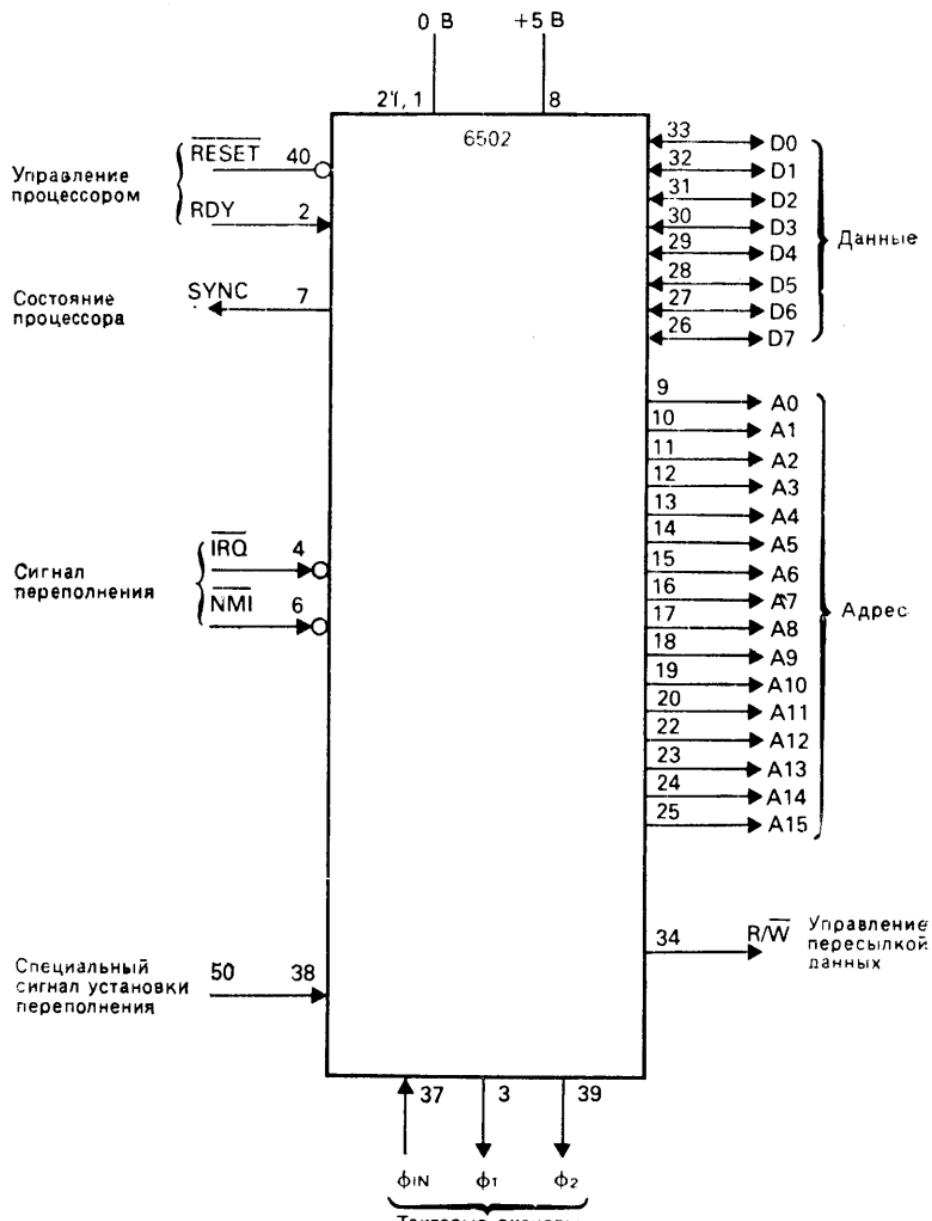


Рис. П1.1. Внешние сигналы и выводы микропроцессора 6502.

используется микропроцессор 8080, необходимо подавать на процессор более одного напряжения питания. Очевидно, что неисправности в схемах питания приводят к некорректной работе микропроцессоров и систем, построенных на их основе.

7. Сигналы тактирования. Они включают входные и выходные сигналы синхронизации. В связи с тем что работа микропроцессора протекает под действием синхросигналов, неисправности тактирования приводят к отказам системы. Если выходной синхросигнал буферизован и не подан на другие компоненты системы, его некорректность не обязательно является причиной отказа. В некоторых микропроцессорах внутренняя память построена на динамических регистрах, и уменьшение частоты тактирования ниже определенного уровня приводит к потере информации.

8. Другие сигналы. Это сигналы, не входящие в охарактеризованные выше рубрики и, как правило, отражающие специфичные функции конкретных микропроцессоров.

Микропроцессор 6502

Этот 8-разрядный микропроцессор по структуре аналогичен 6800, т. е. имеет такие же наборы команд и регистров. Сигналы микропроцессора 6502 и соответствующие им номера внешних выводов, показанные на рис. П1.1, кратко описаны ниже.

Сигналы 6502

1. Управление процессором

RESET	Приводит процессор в начальное состояние, после прекращения этого сигнала в счетчик команд загружается содержимое областей памяти, имеющих адреса FFFC и FFFD
RDY	Приводит процессор в состояние ожидания

2. Управление пересылкой данных

R/W	Одна линия, служащая для указания, выполняет процессор чтение или запись данных (1 — чтение, 0 — запись). Процессор не может адресовать устройства ввода-вывода отдельно от памяти
-----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. Шины адресов и данных

D0—D7	Обычные немультиплексируемые шины
A0—A15	

4. Состояние процессора

SYNC	Информирует о том, что микропроцессор выполняет выборку команды
------	-----------------------------------------------------------------

5. Обработка прерываний

NMI

Сигнал немаскируемого запроса прерывания. При его поступлении в счетчик команд загружается содержимое областей FFFA и FFB и выполнение программы продолжается, начиная с этого адреса

IRQ

Сигнал маскируемого запроса прерывания. В счетчик команд загружается содержимое областей FFFE и FFFF

6. Питание

0 В, 5 В

Необходимо лишь одно напряжение питания +5 В

7. Тактирование

ϕ_{IN}
 ϕ_1, ϕ_2

Входной тактовый сигнал

Двухфазный выходной тактовый сигнал

8. Другие сигналы

SO

Устанавливает флаг переполнения

На рис. П1.2 показан набор внутренних регистров процессора. Их количество меньше, чем во многих 8-разрядных, но этот недостаток компенсируется наличием команд, с помощью которых осуществляется быстрый доступ к адресам памяти 00_{16} — FF_{16} , относящимся к странице 0. Регистр указателя стека также 8-разрядный, предполагается, что стек размещается на странице 1.

Процессор располагает рядом способов адресации, некоторые из которых подразумевают пересылку данных между процессором и двумя смежными областями памяти. При этом всегда первым передается младший байт, а затем старший. К числу этих способов адресации относятся следующие.

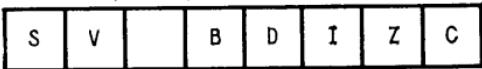
- Аккумуляторная.** Операция выполняется над данными, находящимися в аккумуляторе.
- Непосредственная.** В операции участвуют данные, которые берутся из области памяти, следующей за той, из которой выбрана команда.
- Абсолютная.** Используется в трехбайтовых командах; адрес, по которому должно производиться обращение, находится во втором и третьем байтах.
- В нулевую страницу.** Адрес определяется, как и при абсолютной адресации. Различие состоит в том, что команда имеет двухбайтовую длину и адрес, находящийся во втором байте, считается относящимся к странице 0.

5. В нулевую страницу с индексацией. Для вычисления адреса используются регистры X и Y. Адрес определяется путем сложения второго байта команды и содержимого соответствующего индексного регистра.

6. Абсолютная с индексацией. Для получения адреса используются регистры X и Y. Адрес образуется путем сложения 16-разрядного абсолютного адреса, находящегося во втором и третьем байтах команды, с содержимым соответствующего индексного регистра.



Флаговый регистр



S – Знак

V – Переполнение

B – Прерывание приема

D – Десятичная обработка

I – Блокировка прерываний

Z – Нуль

C – Перенос

Рис. П1.2. Регистры микропроцессора 6502.

щему за адресом команды, образуют двубайтовый исполнительный адрес.

10. Индексная косвенная. Используется в двубайтовых командах. Второй байт команды складывается с содержимым регистра X, и результат используется как адрес, относящийся к странице 0. По этому и следующему за ним адресам находится исполнительный адрес.

11. Косвенная с индексацией. Применяется в двубайтовых командах. Второй байт команды адресует область, принад-

7. Неявная. В команде подразумевается использование определенного регистра.

8. Относительная. В командах условного перехода второй байт содержит величину смещения адреса команды, которая должна выполняться следующей, по отношению к очередному адресу. Таким образом, переход может осуществляться не более чем на 128 областей назад и не более чем на 127 вперед.

9. Абсолютная косвенная. Во втором и третьем байтах команды находится адрес области памяти, содержимое которой вместе с содержимым области, находящейся по адресу, следую-

лежащую странице 0. Содержимое этой и следующей за ней областей рассматривается как 16-разрядный адрес, к которому для формирования исполнительного адреса прибавляется содержимое регистра Y.

Команды микропроцессора 6502

Команды арифметических и логических операций

ADC	Сложение с содержимым регистра A с переносом
SBC	Вычитание с переносом
AND	Логическое И
ORA	Логическое ИЛИ
EOR	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ
CMP	Сравнение с аккумулятором
CPX	Сравнение с регистром X
CPY	Сравнение с регистром Y
INC	Инкремент содержимого памяти
DEC	Декремент содержимого памяти
INX/INY	Инкремент содержимого регистра X/Y
DEX/DEY	Декремент содержимого регистра X/Y
ROL/ROR	Циклический сдвиг влево/вправо с переносом
ASL/LSR	Сдвиг влево/вправо

Команды пересылки данных

LDA	Загрузка в аккумулятор
STA	Запись в память содержимого аккумулятора
LDX/LDY	Загрузка в регистр X/Y
STX/STY	Запись в память содержимого регистра X/Y
TAX/TAY	Пересылка из аккумулятора в регистр X/Y
TXA/TYA	Пересылка из регистра X/Y в аккумулятор
TSX	Пересылка из регистра указателя стека в регистр X
TXS	Пересылка из регистра X в регистр указателя стека

Команды переходов

JMP	Переход
JSR	Переход к подпрограмме
RTI	Возврат из подпрограммы обработки прерывания
RTS	Возврат из подпрограммы
BCC	Переход по отсутствию переноса

BCS	Переход по наличию переноса
BEQ	Переход по нулевому результату
BMI	Переход по отрицательному результату
BPL	Переход по положительному результату
BVC	Переход по отсутствию переполнения
BVS	Переход по наличию переполнения

Команды стековых операций

PHA/PHP	Занесение в стек содержимого регистра А/флагов
PLA/PLP	Загрузка регистра А/флагов из стека

Команды обработки прерываний

CLI	Сброс бита блокировки прерываний в регистре флагов
SEI	Установка бита блокировки прерываний
BRK	Прерывание приема

Другие команды

SEC	Установка флага переноса
SED	Установка флага десятичной обработки
NOP	Нет операции
CLC	Сброс флага переноса
CLD	Сброс флага десятичной обработки
CLV	Сброс флага переполнения
BIT	Битовая проверка

Микропроцессор 8085

Это усовершенствованная версия микропроцессора 8080, имеющая две дополнительные команды: RIM и SIM, одно питательное напряжение, мультиплексированную шину адреса/данных, расширенные возможности обработки прерываний. На рис. П1.3 показаны внешние сигналы и выводы микропроцессора 6502.

Сигналы 8085**1. Управление центральным процессором**

RESET IN	Сигнал начальной установки процессора. Счетчик команд устанавливается в состояние 0000
READY	Указывает, что память или система ввода-вывода готовы к посылке или приему данных. Если сигнал имеет низкий уровень, процессор ожидает появления высокого уровня, для того чтобы завершить цикл чтения или записи

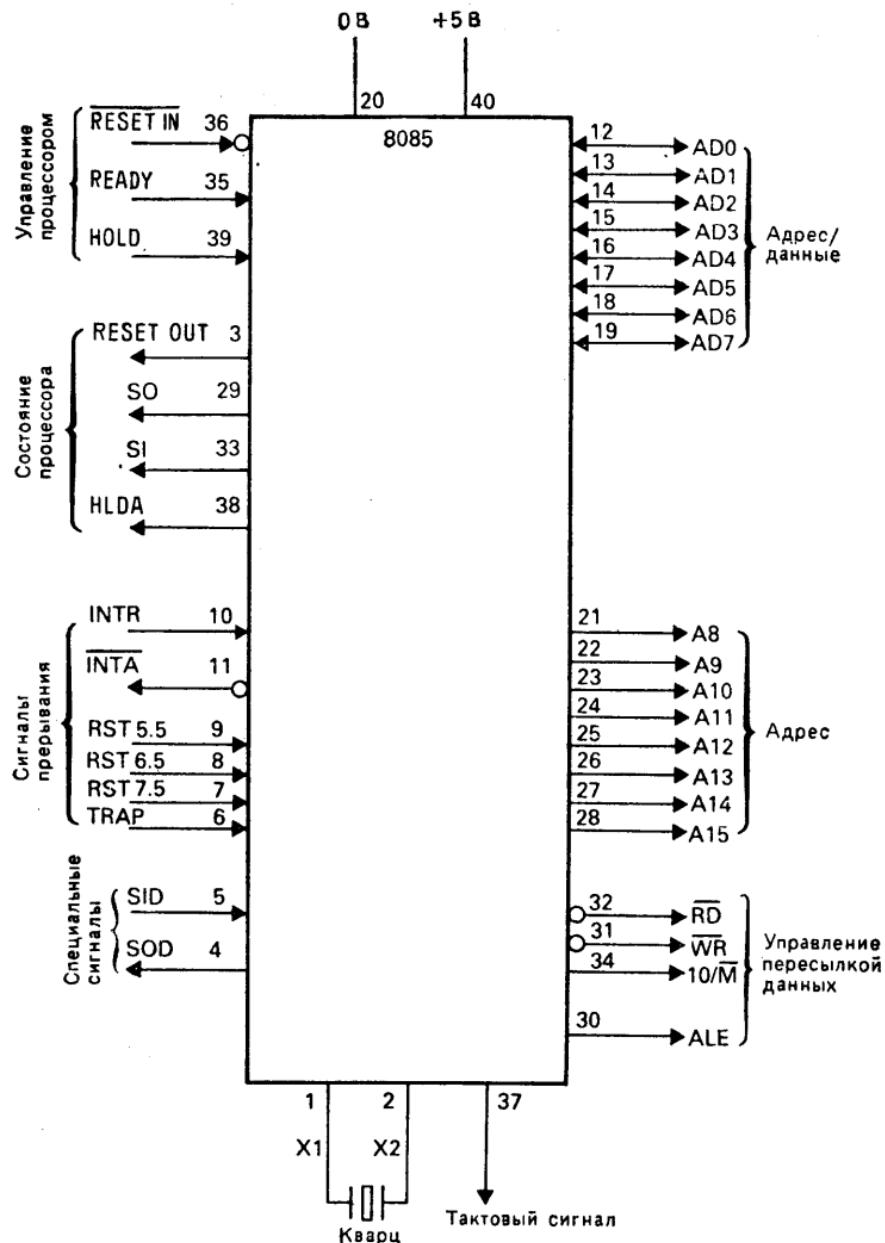


Рис. П1.3. Внешние сигналы и выводы микропроцессора 8085.

HOLD

Сигнал подается другим устройством и является запросом на использование шин адресов и данных. Процессор предоставляет шины в распоряжение запрашивающего устройства по окончании текущей пересылки данных

2. Управление пересылкой данных

<u>RD</u>	Сигнал управления чтением
<u>WR</u>	Сигнал управления записью
<u>ALE</u>	Сигнал демультиплексирования адресной шины. Он имеет высокий уровень, когда младшие 8 линий адресной шины используются для передачи адресной информации
<u>IO/M</u>	Сигнал имеет низкий уровень при обмене информацией с памятью и высокий при пересылках ввода-вывода

3. Шины адреса и данных

AD0—AD7	Адреса и данные передаются с использованием мультиплексирования. В первом цикле любой пересылки с участием памяти или устройств ввода-вывода по линиям AD0—AD7 передается адресная информация, а в течение оставшегося времени — данные
AD8—AD15	

4. Состояние центрального процессора

RESET OUT	Информирует о том, что процессор подвергается начальной установке. Этот сигнал стробируется тактовыми сигналами
HLDA	Указывает, что сигнал запроса шин HOLD получен и шины будут освобождены в следующем цикле. После снятия сигнала HOLD этот сигнал тоже приобретает низкий уровень, и шины отдаются в распоряжение запрашивавшего их устройства через половину цикла синхронизации

5. Обработка прерываний

INTR	Маскируемый сигнал прерывания общего назначения. Его получение подтверждается сигналом INTA, который может быть использован прерывающим устройством для подачи команды на шину данных. Эта команда, обычно RST или CALL, считывается из памяти. По команде CALL дважды осуществляется чтение информации с шины данных для получения двух байтов адреса, которые также должны быть переданы прерывающим устройством
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

RST 5.5	Сигналы аппаратных прерываний, которые могут быть замаскированы с помощью команды SIM. По этим сигналам, если они не заблокированы и не замаскированы, процессор переходит к выполнению программ, начинающихся соответственно с адресов 2C, 34 и 3C
TRAP	Немаскируемый сигнал прерывания. Программа обработки прерываний этого типа должна начинаться с адреса 0024 ₁₆
INTA	Сигнал, подаваемый процессором для подтверждения получения запроса на прерывание INTR; он может быть использован аппаратурой обработки прерываний для вывода команд на шину данных

6. Питание

0B, 5В	Необходимо лишь одно напряжение питания +5 В
--------	----------------------------------------------

7. Тактирование

X1, X2	Непосредственно к этим внешним выводам подключается кварцевый кристалл с резонансной частотой до 6 МГц для микропроцессора 8085А (до 10 МГц для 8085А—1). В составе микропроцессора имеется тактовый генератор и делитель, один из выходов которого буферизирован и выведен в виде сигнала CLK
--------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

8. Другие сигналы

SID	Одиночная входная линия данных, информация с которой может быть считана в разряд 7 регистра А с помощью команды RIM
SOD	Одиночная выходная линия данных, по которой выводится состояние бита 7 регистра А, когда при единичном состоянии бита 6 регистра А выполняется команда SIM

Эти линии могут быть использованы для последовательного ввода и вывода информации под управлением программы.

На рис. П1.4 показан набор внутренних регистров процессора, в том числе аккумулятор и шесть 8-разрядных регистров общего назначения, причем последние могут объединяться в пары, образуя 16-разрядные регистры. Указатель стека и счетчик команд — это 16-разрядные регистры.

При выполнении операций данные могут быть адресованы различными способами, рассмотренными ниже. При занесении в память 16-разрядных данных первым всегда записывается

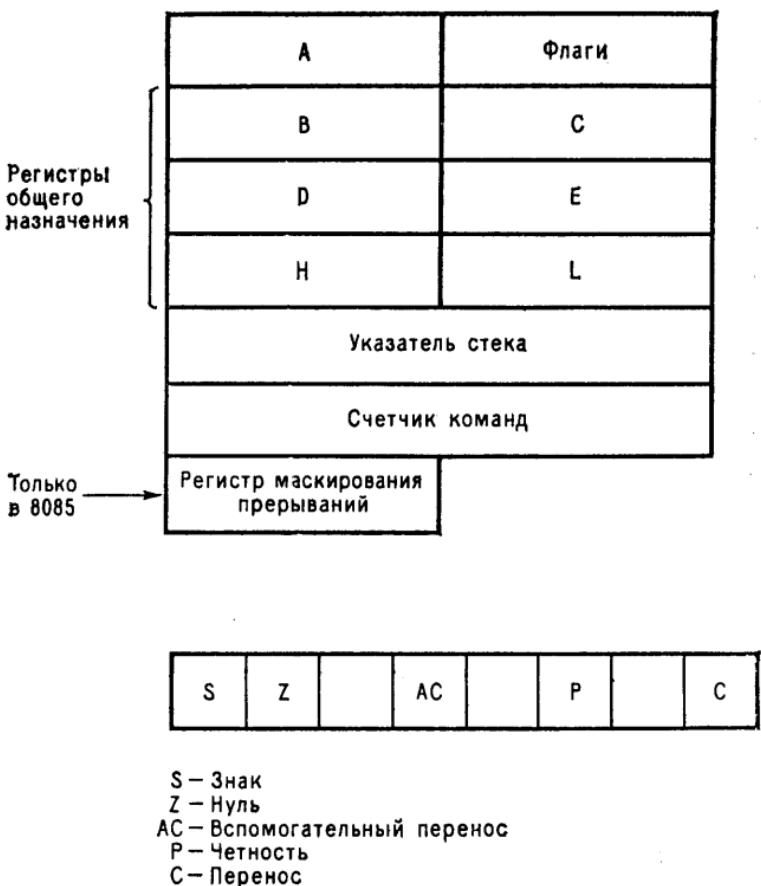


Рис. П1.4. Регистры микропроцессоров 8080 и 8085.

младший байт. В 8085 используются следующие виды адресации.

- Регистровая.** В однобайтовой команде неявным образом подразумевается обращение к одному из двух регистров.
- Непосредственная.** Операция выполняется над данными, содержащимися во втором байте двубайтовой команды.
- Расширенная непосредственная.** Второй и третий байты трехбайтовой команды используются в качестве 16-разрядных данных.

- 4. Абсолютная.** Адрес, по которому происходит обращение к памяти, содержится во втором и третьем байтах трехбайтовой команды.
- 5. Регистровая косвенная.** Адрес, по которому происходит обращение к памяти, находится в регистровой паре. В качестве регистровой пары, содержащей исполнительный адрес, используется пара HL.
- 6. Двойная регистровая.** В однобайтовой команде неявным образом задается обращение к одной или двум регистровым парам.
- 7. Портовая.** При выполнении двух команд адресная шина и управляющий сигнал $10/\bar{M}$ используются таким образом, что не происходит выборки памяти.

Ниже приводится обобщенная сводка команд микропроцессоров 8080 и 8085. Отметим, что во всех командах переходов используется абсолютная адресация и, следовательно, переход может быть произведен к любой области памяти.

Команды микропроцессоров 8080/8085

Команды арифметических и логических операций

ACI	Сложение с переносом непосредственное
ADC	Сложение с переносом
ADD	Сложение
ADI	Сложение непосредственное
DAD	Сложение двойное регистровое (с HL)
SBB	Вычитание с заемом
SBI	Вычитание с заемом непосредственное
SUB	Вычитание
SUI	Вычитание непосредственное
ANA	Логическое И
ANI	Логическое И непосредственное
ORA	Логическое ИЛИ
ORI	Логическое ИЛИ непосредственное
XRA	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ
XRI	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ непосредственное
CMP	Сравнение
CPI	Сравнение непосредственное
RAL/RAR	Циклический сдвиг левый/правый с переносом
RLC/RRC	Циклический сдвиг левый/правый
INR	Инкремент

DCR

Декремент

INX

Инкремент регистровой пары

DCX

Декремент регистровой пары

Команды пересылки данных

MOV

Пересылка данных

MVI

Загрузка непосредственная

LXI

Загрузка регистровой пары непосредственная

LDA

Загрузка аккумулятора прямая из памяти

LDAX

Загрузка аккумулятора косвенная из памяти

STA

Запись содержимого аккумулятора в память прямая

STAX

Запись содержимого аккумулятора в память косвенная

Команды переходов

CALL/RET/JMP

Вызов/Возврат/Переход абсолютный

CNZ/RNZ/JNZ

Вызов/Возврат/Переход, если не нуль

CZ/RZ/JZ

Вызов/Возврат/Переход, если нуль

CNC/RNC/JNC

Вызов/Возврат/Переход, если нет переноса

CC/RC/JC

Вызов/Возврат/Переход, если перенос

CPO/RPO/JPO

Вызов/Возврат/Переход, если нечетно

CPE/RPE/JPE

Вызов/Возврат/Переход, если четно

CP/RP/JP

Вызов/Возврат/Переход, если плюс

CM/RM/JM

Вызов/Возврат/Переход, если минус

RST

Повторный запуск

PCHL

Пересылка HL в счетчик команд

Команды стековых операций

PUSH

Занесение в стек

POP

Загрузка из стека

XTHL

Обмен вершины стека и HL

Команды обработки прерываний

EI

Разрешение прерываний

DI

Запрещение прерываний

Команды ввода-вывода

IN

Ввод из порта в регистр A

OUT

Вывод впорт из регистра A

Другие команды

NOP

Нет операции

HLT

Останов

CMA	Инверсия аккумулятора
CMC	Инверсия переноса
STC	Установка флага переноса
DAA	Десятичная коррекция аккумулятора
RIM	Чтение маски прерываний в
SIM	регистр А Установка маски прерываний в регистре А
	Только в 8085

Микропроцессор Z80

Этот микропроцессор представляет собой один из более совершенных вариантов 8080, с дополнительными регистрами, командами и некоторыми другими возможностями. Для него требуется лишь однофазное тактирование и один источник питания +5 В; микропроцессор, однако, не вырабатывает сигнала подтверждения прерывания, который в случае необходимости должен быть сформирован с помощью внешней по отношению к микропроцессору схемы. Для выбора памяти и системы ввода-вывода используются отдельные сигналы. Все входные и выходные управляющие сигналы в активном состоянии имеют низкий уровень. Внешние сигналы и выводы микропроцессора показаны на рис. П1.5 и описаны ниже.

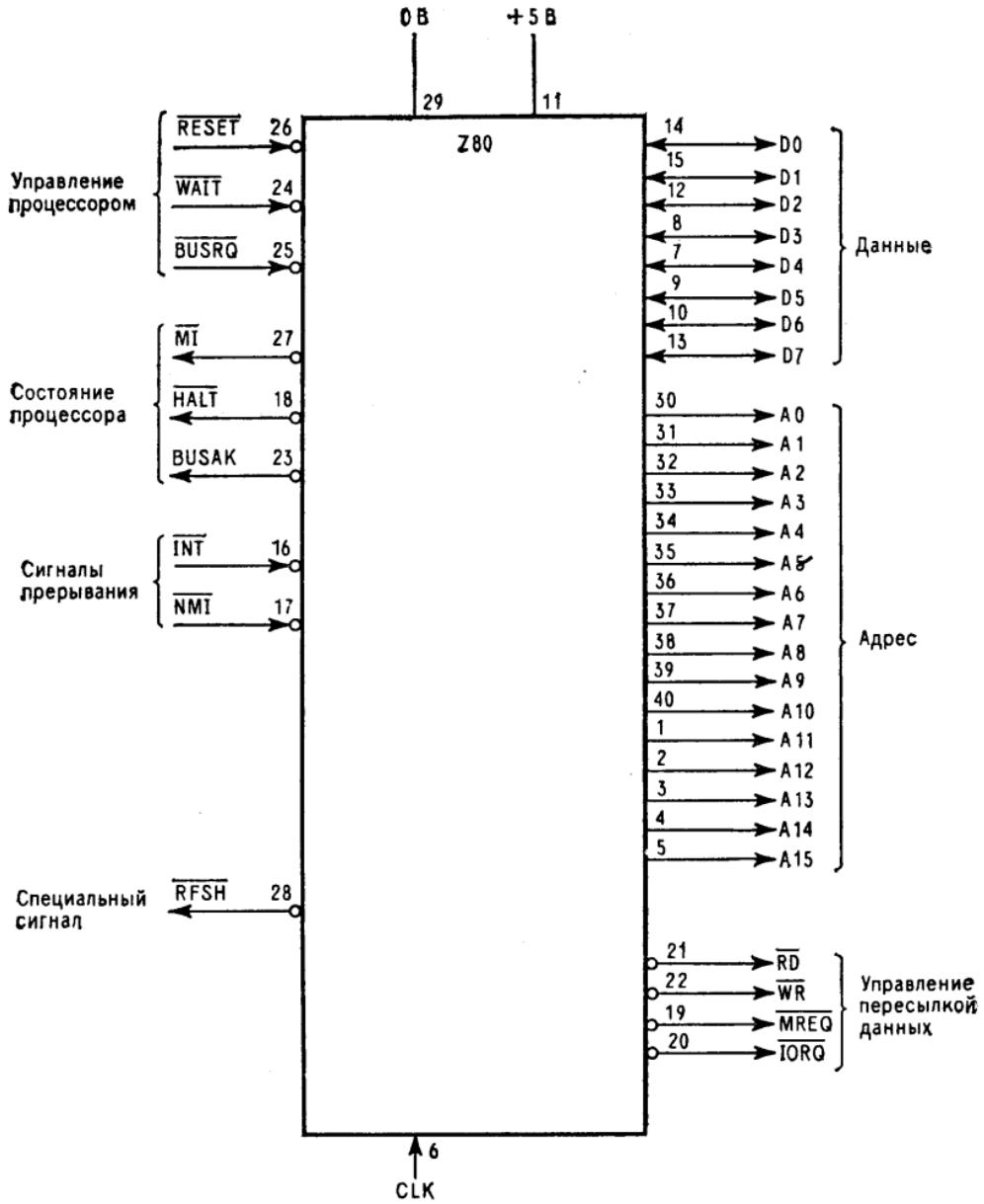
Сигналы микропроцессора Z80

1. Управление центральным процессором

<u>RESET</u>	Входной сигнал, имеющий в активном состоянии низкий уровень и служащий для начальной установки процессора. В счетчик команд заносится значение 0000
<u>WAIT</u>	Указывает процессору, что он должен перейти в состояние ожидания
<u>BUSRQ</u>	Входной сигнал, имеющий в активном состоянии низкий уровень, который может подаваться устройством, запрашивающим доступ к системным шинам

2. Управление пересылкой данных

<u>RD</u>	Используется, когда процессору необходимо прочитать данные из памяти или из системы ввода-вывода
<u>WR</u>	Сигнал управления записью
<u>MREQ</u>	Пребывает в активном состоянии, когда имеет место обмен информацией с памятью



Примечание: когда сигналы M1 и IORQ одновременно имеют низкий уровень, это является признаком подтверждения прерывания. Поэтому для обработки прерываний оба они должны быть поданы на интерфейсные ИС процессора Z80.

Схема формирования
сигнала INTA

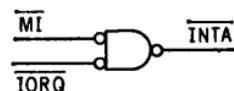


Рис. П1.5. Внешние сигналы и выводы микропроцессора Z80.

IORQ

Пребывает в активном состоянии, когда имеет место обмен информацией с системой ввода-вывода, т. е. при выполнении команд IN и OUT

3. Шины адресов и данныхD0—D7

Процессор располагает восемью линиями данных и 16-ю линиями адреса. При освобождении шин по запросу BUSRQ соответствующие выходные схемы процессора приводятся в состояние высокого импеданса

4. Состояние процессораMI

Выходной сигнал, имеющий в активном состоянии низкий уровень и информирующий о том, что происходит выборка команд

HALT

Выходной сигнал, имеющий в активном состоянии низкий уровень и информирующий о том, что была выполнена команда HLT и процессор соответственно остановлен

BUSA_K

Сигнал указывает, что процессор снял с себя управление шинами

5. Обработка прерыванийINT

Сигнал прерывания общего назначения, который используется одним из трех способов, может быть заблокирован командой DI и разрешен командой EI

NMI

Немаскируемый сигнал прерывания, по которому в программе происходит переход по адресу 0066₁₆ после запоминания текущего значения программного счетчика в стеке
Совместно действуют как сигнал подтверждения прерывания

6. Питание

0, 5 В

Необходимо одно напряжение питания +5 В

7. ТактированиеCLK

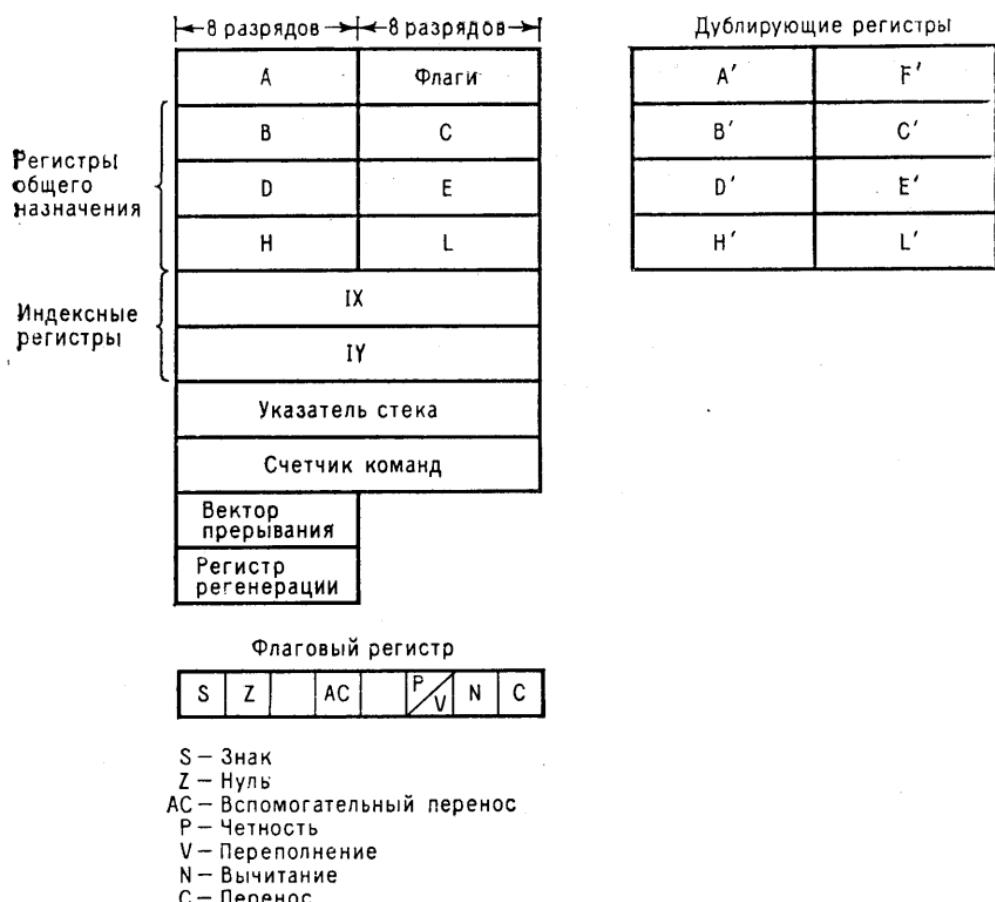
Требуется один тактирующий сигнал, частота которого не подвергается делению внутри процессора. Для других устройств, нуждающихся в тактировании, должен использоваться этот же сигнал, так как буферизированного выходного сигнала не предусмотрено

8. Другие сигналы

RFSH

Сигнал пребывает в активном состоянии в течение последнего периода тактирования цикла MI, содержимое регистра регенерации выводится на младшие линии адресной шины

Набор внутренних регистров микропроцессора Z80 показан на рис. П1.6. Видно, что он почти идентичен набору регистров 8085,



Дублирующие регистры

A'	F'
B'	C'
D'	E'
H'	L'

Рис. П1.6. Регистры микропроцессора Z80.

но включает в дополнение к нему два индексных регистра, регистр вектора прерывания и регистр регенерации. Кроме того, в числе регистров Z80 имеется дублирующий набор регистров, а среди его команд предусмотрены команды обмена содержимым между рабочими и дублирующими регистрами. Это позволяет быстро запоминать данные, находящиеся в рабочих регистрах, когда возникает необходимость занесения в них новой

информации, например, при обращениях к подпрограммам. Содержимое индексных регистров может быть использовано совместно со смещениями, указываемыми в командах, для определения значений адресов с помощью таблиц.

Набор команд Z80 включает все команды 8085, кроме RIM и SIM, поэтому в Z80 применяются все способы адресации, используемые в 8085; кроме них имеются дополнительные способы относительной адресации, применяемые в командах переходов.

Команды микропроцессоров Z80 и NSC800

Команды арифметических и логических операций

ADC	Сложение с переносом (8- и 16-разрядное)
ADD	Сложение (8- и 16-разрядное)
SBC	Вычитание с переносом (8- и 16-разрядное)
SUB	Вычитание
AND	Логическое И
OR	Логическое ИЛИ
XOR	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ
CP	Сравнение
INC	Инкремент
DEC	Декремент
RLA	Циклический сдвиг левый регистра А с переносом
RRA	Циклический сдвиг правый регистра А с переносом
RLCA	Циклический сдвиг левый регистра А
RRCA	Циклический сдвиг правый регистра А
RL	Циклический сдвиг левый с переносом
RLC	Циклический сдвиг левый
RR	Циклический сдвиг правый с переносом
RRC	Циклический сдвиг правый
RLD	Десятичный циклический сдвиг левый
RRD	Десятичный циклический сдвиг правый
SLA	Арифметический сдвиг левый
SRA	Арифметический сдвиг правый
SRL	Логический сдвиг правый

Команды пересылки данных

LD a, b Загрузка в а из b,

- где 1) а — любой из регистров A, B, C, D, E, H, L,
 b — любой из регистров A, B, C, D, E, H, L, (HL),
 (IX + смещение), (IY + смещение) или 8-разрядное
 число;
- 2) а — регистр A,
 b — (BC), (DE), (адрес), R или I;

- 3) a — (BC), (DE), (адрес), R или I,
b — регистр A;
- 4) a — (HL), (IX + смещение), (IY + смещение),
b — A, B, C, D, E, H, L или 8-разрядное число;
- 5) a — (адрес),
b — A, BC, DE, HL, IX, IY, указатель стека;
- 6) a — BC, DE, HL,
b — (адрес) или 16-разрядное число;
- 7) a — указатель стека,
b — (адрес), HL, IX, IY или 16-разрядное число.

Команды переходов

JP/CALL/RET	Переход/Вызов/Возврат абсолютный
JP/CALL/RET NZ	Переход/Вызов/Возврат, если не нуль
JP/CALL/RET Z	Переход/Вызов/Возврат, если нуль
JP/CALL/RET NC	Переход/Вызов/Возврат, если нет переноса
JP/CALL/RET C	Переход/Вызов/Возврат, если перенос
JP/CALL/RET PE	Переход/Вызов/Возврат, если четно
JP/CALL/RET PO	Переход/Вызов/Возврат, если нечетно
JR, смещение	Переход относительный
JR C, смещение	Переход относительный, если перенос
JR NC, смещение	Переход относительный, если нет переноса
JR NZ, смещение	Переход относительный, если не нуль
JR Z, смещение	Переход относительный, если нуль
DJNZ	Декремент В и переход относительный, если не нуль
RETI	Возврат из прерывания
RETN	Возврат из немаскируемого прерывания

Команды стековых операций

PUSH	Занесение в стек } AF, BC, DE, HL,
POP	Загрузка из стека } IX, IY

Команды побитовых операций

BIT	Проверка бита	Любой бит из A, B, C, D, E, H, L, (HL),(IX + смещение), (IY + смещение)
SET	Установка бита	
RES	Сброс бита	

Команды блочных операций

IND	Блочная загрузка с декрементом из порта (C)
INI	Блочная загрузка с инкрементом из порта (C)

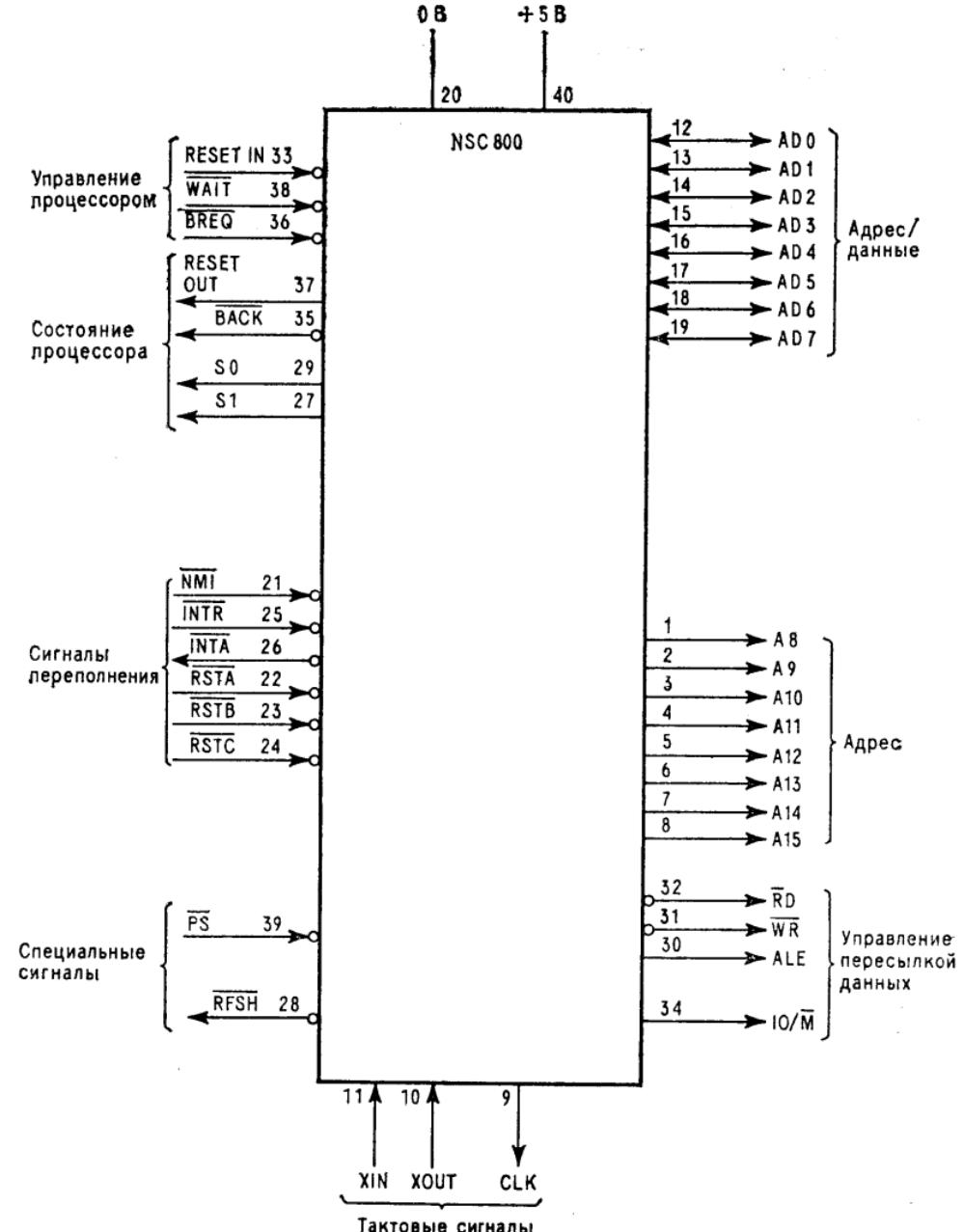


Рис. П1.7. Внешние сигналы и выводы микропроцессора NSC800.

INDR

Блочная загрузка с автодекрементом из порта (C)

INIR

Блочная загрузка с автоинкрементом из порта (C)

OTDR

Блочная загрузка с автодекрементом в порт (C)

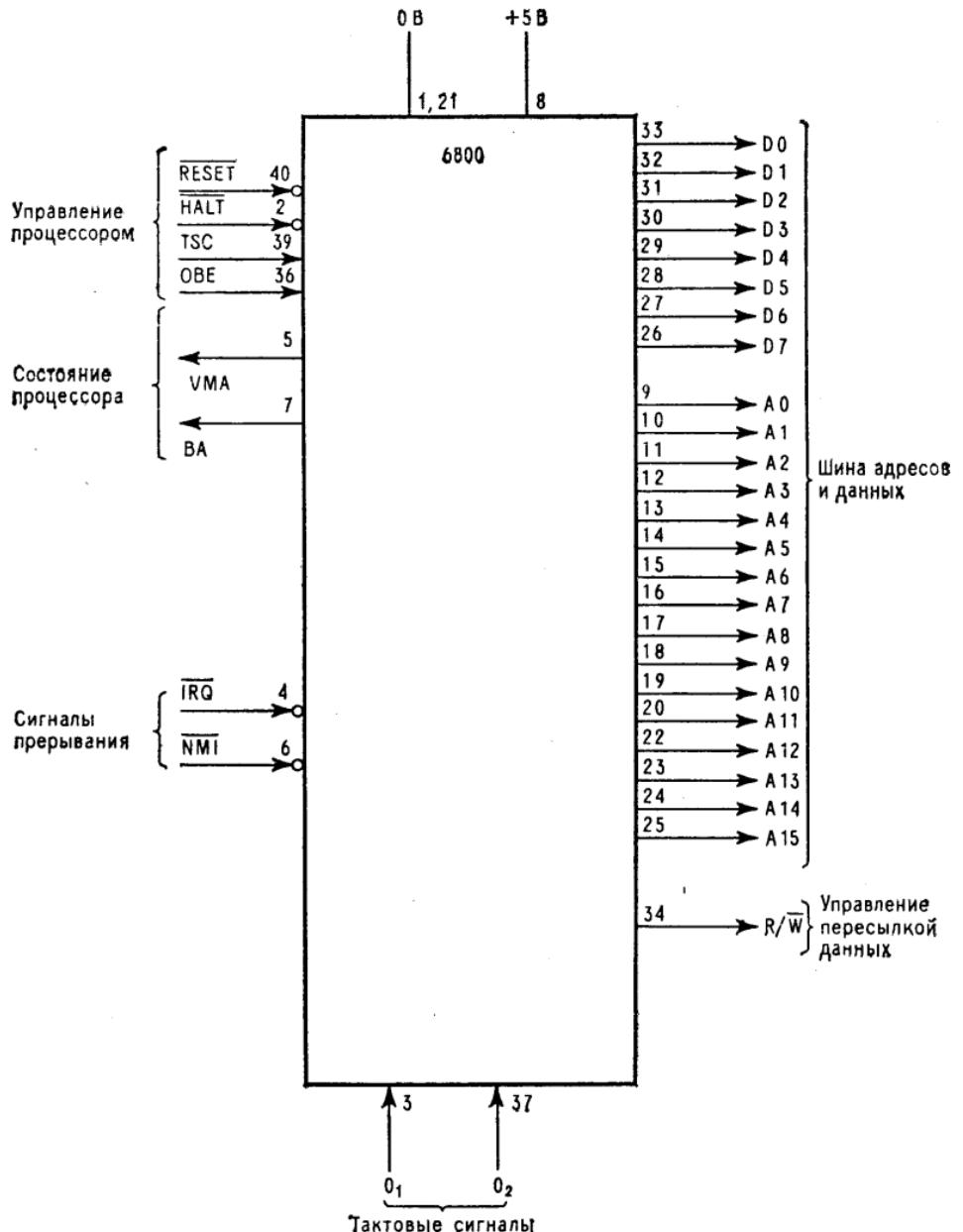


Рис. П1.8. Внешние сигналы и выводы микропроцессора 6800.

OTIR

Блочная загрузка с автоинкрементом в порт (C)

OUTD

Блочная загрузка с декрементом в порт (C)

OUTI

Блочная загрузка с инкрементом в порт (C)

LDI	Блочная пересылка с декрементом
LDI	Блочная пересылка с автодекрементом
LDI	Блочная пересылка с инкрементом
LDI	Блочная пересылка с автоинкрементом

Другие команды

DAA	Десятичная коррекция аккумулятора
CFF	Инверсия флага переноса
CPD	Сравнение с декрементом памяти
CPL	Инверсия регистра А
CPDR	Поиск с декрементом памяти
CPIR	Сравнение с инкрементом памяти
NEG	Формирование дополнительного кода регистра А
NOP	Нет операции
SCF	Установка флага переноса

Примечание: в мнемонических обозначениях команд языка ассемблера Z80 выражение, заключенное в скобки, означает, что обращение происходит к области памяти, адрес которой определяется путем вычисления значения этого выражения. Например, при выполнении команды LDA, (IX + 5) производится загрузка в регистр А содержимого области памяти, адрес которой образуется прибавлением смещения 5 к содержимому индексного регистра.

На рис. П1.7 и П1.8 показаны сигналы и назначение внешних выводов микропроцессоров NSC800 и 6800.

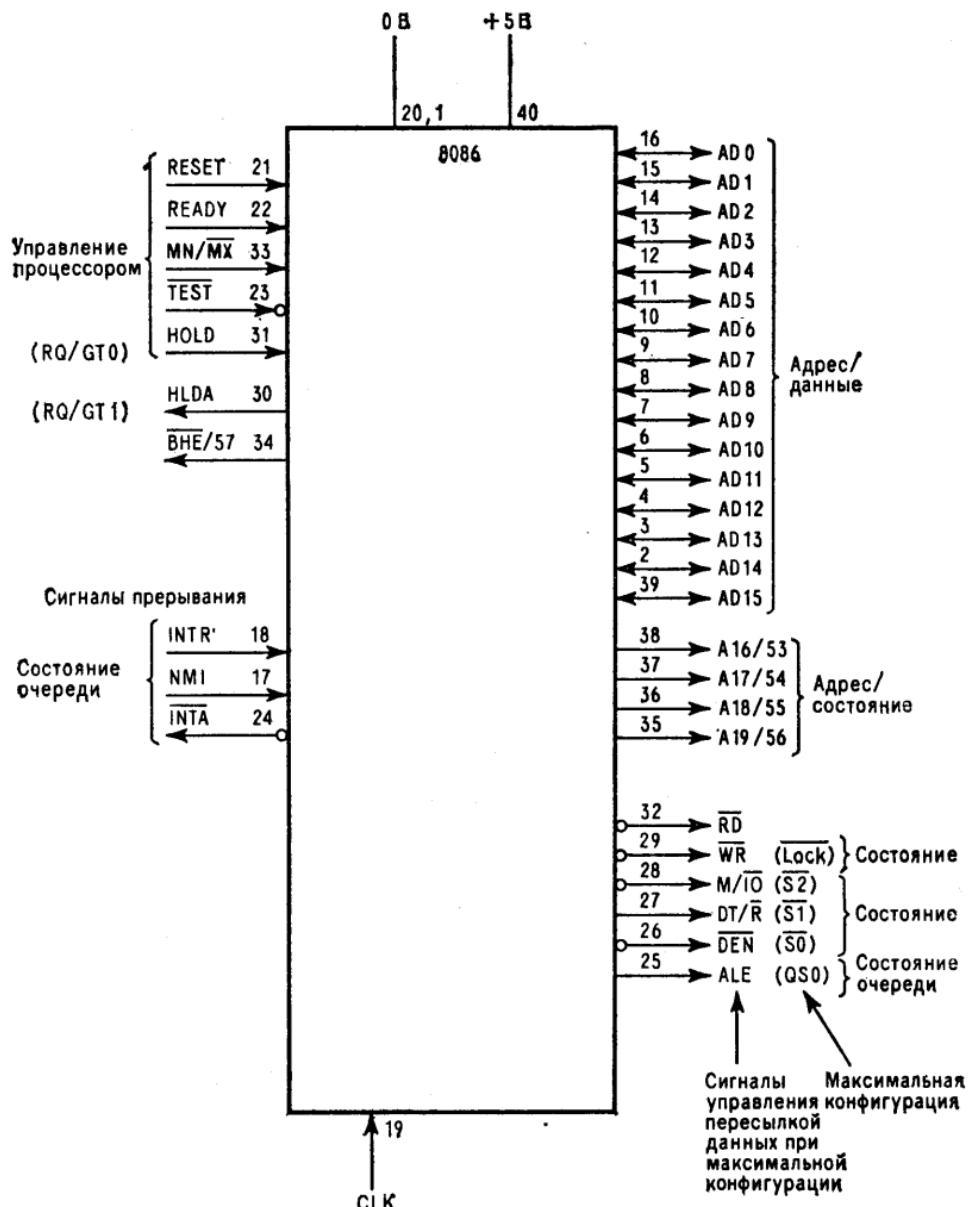
Микропроцессоры 8088 и 8086

Выпущенный в 1978 г. микропроцессор 8086 был первым 16-разрядным микропроцессором фирмы Intel. Он был спроектирован как совместимый «снизу-вверх» с 8080 на уровне языка ассемблера (т. е. набор ассемблерных команд 8080 является подмножеством набора ассемблерных команд 8086), но не совместимый на уровне машинного языка.

Как и 8080, микропроцессор 8086 является лишь одной из БИС в микропроцессорной системе. В дополнение к 8086 обычно применяются следующие БИС:

- 8284 — генератор тактовых импульсов с делителем частоты;
- 8288 — шинный контроллер, применяемый в ситуациях, когда процессор работает в так называемой максимальной конфигурации (MAXIMUM MCDE), т. е. со-вместно с сопроцессорами.

Микропроцессор 8088 — это то же, что 8086, но с 8-разрядной шиной данных. Эти процессы идентичны во многих



Примечания:

- Названия сигналов, заключенные в скобки, например (\overline{LOCK}) , относятся к максимальной конфигурации.
- Сигналы состояния S0 – S7 мультиплексируются.

Рис. П1.9. Внешние выводы и сигналы микропроцессора 8086.

отношениях (рис. П1.9). Имеющие место различия во внешних связях отмечены в следующем ниже описании.

Сигналы микропроцессоров 8088 и 8086

Примечание: сигналы, используемые при минимальной конфигурации, выделены с помощью скобок, как, например, (HOLD).

1. Управление центральным процессором

RESET

Начальная установка системы осуществляется при поступлении положительного фронта этого сигнала. Блокируются прерывания и пошаговый режим работы. Выполнение программы после начальной установки начинается с области памяти с адресом $FFFF0_{16}$, причем содержимое регистров DS, ES, SS и PC равно 00_{16} . Служит для указания процессору, что память или устройство ввода-вывода готовы к посылке или приему данных. Если этот сигнал имеет низкий уровень, 8086 находится в состоянии ожидания, пока READY не приобретет высокого уровня. Этот сигнал, как и тактовые сигналы, поступает в процессор через 8284.

(HOLD)

Другие устройства пользуются этим сигналом для запроса на использование системных шин. Когда он приобретает высокий уровень, процессор 8086 по завершении текущего шинного цикла переходит в состояние отключения шин (захвата их другим устройством) и сообщает об этом, придавая высокий уровень сигналу HOLD.

TEST

Когда в программе процессора встречается команда WAIT, он переходит в состояние простоя, если сигнал TEST имеет высокий уровень.

MN/MX

Служит для указания одного из режимов работы процессора:

(а) Минимальная конфигурация. При подаче на контакт MN/MX напряжения питания процессор 8086 устанавливается в режим минимальной конфигурации; выводам 24—31 присваиваются следующие функции: INTA, ALE, DEN, DT/R, IO/M, WR, HLDA, HOLD;

(б) Максимальная конфигурация. Этот режим задается подачей на контакт MN/MX потенциала "земли". В максимальной конфигурации процессор обычно используется совместно с шинным контроллером 8288, который обеспечивает взаимодействие 8086 с сопроцессорами. Контакты 24—31 служат при этом для передачи следующих сигналов: QS1, QS0, S₀, S₁, S₂, LOCK, RQ/GT0, RQ/GT1.

2. Управление пересылкой данных

<u>RD</u>	Сигнал управления чтением
<u>(WR)</u>	Сигнал управления записью
<u>(ALE)</u>	Положительный импульс ALE означает, что на шине адреса/данных находится достоверный адрес памяти
<u>(DEN)</u>	Служит разрешающим сигналом для приемо-передатчиков шины данных
<u>(DT/R)</u>	Управляет направлением передачи через приемо-передатчики шины данных. Выполняет те же функции, что S1 в режиме максимальной конфигурации. При высоком уровне сигнала данные передаются из процессора на системную шину, при низком — с шины в процессор
<u>M/IO</u>	Имеет низкий уровень при выполнении команд с участием устройств ввода-вывода, высокий уровень — при обращениях к памяти (при выводе информации). <i>Примечание:</i> в микропроцессоре 8088 сигнал имеет инверсное значение, т. е. IO/M из соображений совместимости с 8085

Большая часть описанных выше сигналов вырабатывается непосредственно процессором, когда он функционирует в минимальной конфигурации. В системе с максимальной конфигурацией шинный контроллер 8288, используя сигналы состояния S0, S1 и S2, вырабатывает следующую управляющую информацию:

<u>MRDC</u>	Команда чтения из памяти
<u>MWTC</u>	Команда записи в память
<u>AMWC</u>	Упреждающая команда записи в память. Сигнал предупреждает память о предстоящей записи раньше, чем MWTC
<u>IORC</u>	Команда чтения ввода-вывода (ввод)
<u>IOWC</u>	Команда записи ввода-вывода (вывод)
<u>AIOWC</u>	Упреждающая команда записи ввода-вывода (вывод)

Примечание: 8288 для максимальной конфигурации вырабатывает еще следующие сигналы:

DEN	Разрешение передачи данных. Сигнал имеет тот же смысл, что <u>DEN</u> , вырабатываемый
-----	----------------------------------------------------------------------------------------

DT/R
ALE
INTA

процессором, но закодирован инверсным образом (активным уровнем является высокий)
Передача/прием данных
Разрешение регистра адреса
Подтверждение прерывания

3. Шина адреса и данных

8086

AD0—AD15

A16—A19

Мультиплексируемая шина адреса/данных
Старшие четыре линии 20-разрядной шины
данных

8088

AD0—AD7

A8—A19

Мультиплексируемая шина адреса/данных
Старшие 12 линий 20-разрядной шины данных

4. Состояние процессора

(HLDA)

Высокий уровень этого выходного сигнала является подтверждением получения входного сигнала запроса на захват шин HOLD. Подавая сигнал HLDA, процессор приводит соответствующие выходы в состояние высокого импеданса

A19/S6

A18/S5

A17/S4

A16/S3

Через эти выполняющие двойную функцию внешние выводы в течение первого периода синхронизации командного цикла выводятся старшие четыре разряда адреса. Во время остальных периодов цикла выполнения команды эти контакты служат для вывода информации о состоянии.

S4 S3

0 0 Дополнительный сегмент

0 1 Стековый сегмент

1 0 Сегмент программы или несегмент

1 1 Сегмент данных

S5 отражает состояние флага разрешения прерываний

S6 имеет низкий уровень, когда системнойшиной управляет микропроцессор

Когда процессор работает в максимальной конфигурации, выводы S0, S1 и S2 служат для передачи следующих сведений о состоянии процессора в шинный контроллер 8288:

<u>S2</u>	<u>S1</u>	<u>S0</u>	
0	0	0	Подтверждение прерывания
0	0	1	Чтение ввода-вывода
0	1	0	Запись ввода-вывода
0	1	1	Останов
1	0	0	Выборка команды
1	0	1	Чтение памяти
1	1	0	Запись в память
1	1	1	Неактивен

Далее, сигналы QS0 и QS1 несут подробную информацию о состоянии очереди команд процессора.

QS0 QS1

0	0	Нет операции
0	1	Исполняется первый байт команды
1	0	Очередь ликвидирована
1	1	Принят байт следующей команды из очереди

RQ/GT0

Линия запроса/разрешения служит для передачи управления шинами с тремя состояниями. Микропроцессор 8088/86 подтверждает получение запроса, подавая отрицательный импульс по этой линии, по которой и был получен запрос. Когда устройство, получившее шины в распоряжение, заканчивает их использование, оно возвращает управление ими процессору, посыпая ему второй отрицательный импульс

RQ/GT1

Сигнал, аналогичный RQ/GT0, но с низким приоритетом (т. е. процедура предоставления шин по запросу, поступившему по этой линии, реализуется лишь в том случае, если это уже не делается с линией RQ/GT0)

LOCK

Служит для указания, что управление системной шиной другим устройствам передаваться не должно. Установка этого сигнала производится программным путем

5. Обработка прерываний

INTR

Входной сигнал запроса прерывания, запускаемый уровнем. Если бит разрешения прерывания установлен (значение 1) и сигнал

INTR имеет высокий уровень, процессор 8088/86 сначала выполняет процедуру подтверждения прерывания, а затем передает управление программе его обработки

(INTA)

Это выходной сигнал, имеющий низкий уровень, пока 8088/86 выполняет процедуру подтверждения прерывания. В системе с максимальной конфигурацией INTA вырабатывается шинным контроллером 8288

NMI

Входной сигнал немаскируемого запроса прерывания, запускаемый положительным фронтом. Управление передается программе обработки прерываний SERVICE с начальным адресом 00008_{16}

6. Питание

0 В, 5 В

Одно напряжение питания +5 В (+10 %)

7. Тактирование

CLK

Тактовый сигнал, обычно вырабатываемый БИС 8284 (2—5 МГц); резонансная частота кварцевого кристалла в три раза больше тактовой частоты процессора (так, например, для тактирования с частотой 5 МГц используется кварц с резонансной частотой 15 МГц)

Программная модель микропроцессоров 8088/86

На рис. П1.10 представлена программная модель процессоров 8088/86. Набор регистров включает четыре 16-разрядных регистра общего назначения, два 16-разрядных регистра-указателя, два 16-разрядных индексных регистра, 16-разрядный счетчик команд, четыре 16-разрядных регистра сегментов и 16-разрядный регистр состояния процессора.

Адреса и способы адресации

Адрес в микропроцессорах 8088/86 вычисляется путем суммирования содержимого регистра сегмента и действительного адреса, или смещения. К адресу сегмента добавляются четыре нуля, т. е. он всегда является 20-разрядным. Пример:

$$\begin{array}{r}
 \text{содержимое регистра сегмента} \\
 \text{действительный адрес, или смещение} \\
 \text{исполнительный адрес}
 \end{array}
 \quad
 \begin{array}{r}
 20\ 000_{16} \\
 +\ 2\ 800_{16} \\
 \hline
 22\ 800_{16}
 \end{array}$$

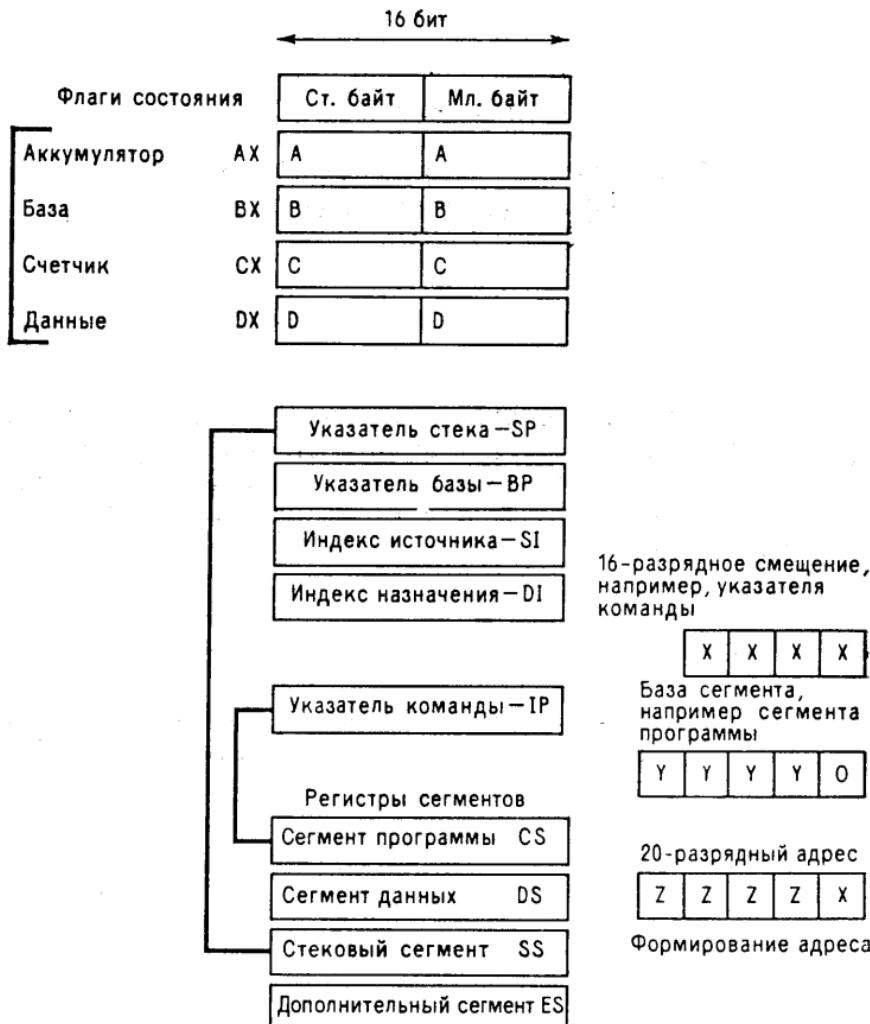


Рис. П1.10. Программная модель микропроцессоров 8088 и 8086.

В микропроцессорах 8086/8088 реализуются следующие способы адресации:

1. Регистровая
2. Непосредственная
3. Прямая
4. Косвенная
5. Базовая
6. Индексная
7. Базово-индексная
8. Цепочная
9. Портовая

Набор команд включает в качестве подмножества все ассемблерные команды 8080 и большинство команд 8085. Для ознакомления с полным набором команд 8088/86 следует обратиться к технической документации фирмы Intel или к соответствующей книге (см., например, R. Rector & G. Alexy, The 8086 Book, Osborne/McGrawHill, 1980).

Приложение 2

Технические характеристики трех одноплатных микроЭВМ

В данный раздел включены основные сведения о трех широко распространенных одноплатных микроЭВМ, упоминавшихся в основных главах книги. Эти микроЭВМ иллюстрируют применение трех основных микропроцессорных семейств: 6502, Intel 8085 и Z80 фирмы Zilog.

Описание микроЭВМ AIM 65

(Взято из руководства для пользователя AIM 65 User Guide с любезного разрешения фирмы Rockwell International Corporation.)

В этом подразделе рассматриваются аппаратные и программные средства AIM 65. Аппаратура из соображений простоты изложения подразделяется на функциональные блоки. Описываются монитор, редактор и ассемблер AIM 65, а также пользовательские подпрограммы, процедуры и условия обращения к ним.

Общие сведения

МикроЭВМ AIM 65 является законченной микропроцессорной системой. В ее составе: центральный процессор R6502, ПЗУ для хранения стандартных программ, ЗУПВ и периферийное оборудование, включающее индикатор, принтер и клавиатуру. Возможность наращивания аппаратных средств путем установки их на плату микроЭВМ или за счет добавления других плат определяет широкую применимость AIM 65. Прикладное применение микроЭВМ существенно облегчается при использовании универсального пользовательского интерфейсного адаптера R6522.

Основные функциональные компоненты AIM 65 показаны на структурной схеме, приведенной на рис. П2.1.

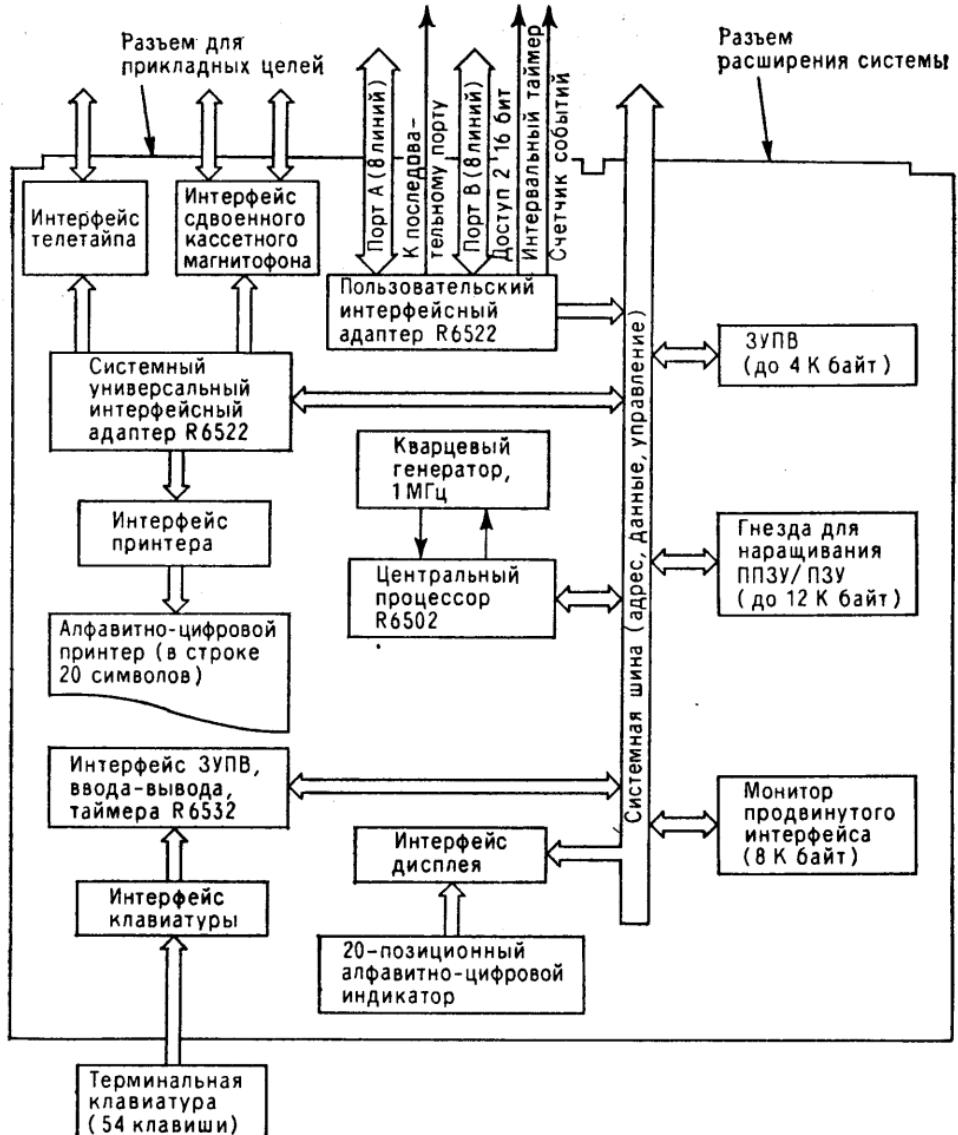


Рис. П2.1. Структурная схема микроЭВМ AIM65.

Функциональный состав

Основными функциональными блоками микроЭВМ являются следующие: система питания, схема выбора кристалла, ЗУПВ, ПЗУ, интерфейс принтера, интерфейс индикатора, интерфейс клавиатуры, пользовательский интерфейс R6522, интерфейс кассетного магнитофона, интерфейс телетайпа и последовательного ввода-вывода.

R6502

Восьмиразрядный микропроцессор R6502 — центральный процессор микроЭВМ AIM 65 — осуществляет общее и текущее управление всеми операциями. Он обменивается информацией с другими устройствами машины по трем отдельным шинам. С помощью 16-разрядной адресной шины процессор может прямо обращаться к 65 536 областям памяти. Посредством двухнаправленной 8-разрядной шины данных происходит пересылка данных между R6502, с одной стороны, памятью и интерфейсными устройствами — с другой. Управляющая шина служит для передачи сигналов синхронизации и управления от процессора R6502 к интерфейсным и периферийным устройствам, компонентам системы, находящимся вне платы микроЭВМ.

Тактирование R6502

Рабочая частота R6502 составляет 1 МГц. Опорный сигнал 4 МГц вырабатывается кварцевым генератором. С помощью пересчетной схемы Z10, построенной на двух триггерах D-типа, эта частота снижается в четыре раза; полученный сигнал используется в качестве входного сигнала фазы тактирования ϕ_0 , имеющего частоту 1 МГц.

В процессоре на основе ϕ_1 вырабатываются фазовые сигналы 1 и 2 (ϕ_1 и ϕ_2). Сигнал ϕ_1 (OUT) выводится через J3-3 для тактирования внешних по отношению к микроЭВМ устройств.

Сигнал ϕ_2 (OUT) процессора R6502 подается на J1-C и на инвертор. Сигнал ϕ_2 , снимаемый с J16-8, подается на J3-Y и Z16-11. Буферизированный тактовый сигнал (SYS \emptyset), вырабатываемый инвертором Z16-10, используется в качестве опорного для системной синхронизации устройств, установленных на плате и вне ее (J3-U).

Сигнал чтения/записи

Сигнал чтения/записи R/W управляет направлением пересылки данных между R6502 и другими устройствами. Он подается на J1-D и инвертор Z16-3. После буферизации на Z16-6 он преобразуется в системный сигнал чтения/записи (SYS P/W), используемый на плате и за ее пределами (J3-V).

Управляющие переключатели

RESET

С помощью кнопки S1 вырабатывается сигнал начальной установки RESET для аппаратных и программных средств

микроЭВМ. Таймер Z4 удерживает низкий уровень сигнала RESET не менее 15 мс после отпускания кнопки. Сигнал RESET подается на процессор R6502, мониторы R6522(Z32) и R6532(Z33), пользовательский интерфейс R6522(Z1) и интерфейс индикатора R6520(UI). Для приведения в начальное состояние других устройств сигнал RESET подается также на разъем расширения системы. Начальную установку программных средств осуществляет монитор, когда линия RESET приобретает высокий уровень.

КВ/TTY

С помощью переключателя S3 (КВ/TTY) микроЭВМ AIM 65 выдается указание принимать управляющие сигналы либо от AIM 65, либо с клавиатуры. Состояние переключателя опрашивается монитором R6522.

Интерфейс принтера

Принтер воспроизводит информацию на термочувствительной рулонной бумаге с помощью десяти термоэлементов, каждый из которых осуществляет печать двух точечно-матричных символов (размер матрицы 5×7 точек). Термоэлементы жестко закреплены на подвижной термоголовке. В процессе печати термоголовка перемещается по горизонтали вперед и назад, причем при движении в каждом направлении печатается одна строка точек. Отдельные термоэлементы во время перемещения термоголовки включаются на дискретные интервалы времени, в результате чего осуществляется постепенное формирование символов. После печати каждой строки точек валик принтера перемещает бумагу на одну строку по вертикали. Печать строки символов завершается после печати семи строк точек. Формирование печатных символов осуществляется с помощью точечных образов, хранящихся в мониторе AIM 65. Инициирование цикла печати, управление и синхронизация его прохождения также выполняются монитором AIM 65.

Интерфейс индикатора

Индикатор AIM 65 состоит из пяти 4-символьных 16-сегментных алфавитно-цифровых индикаторных элементов. Каждый индикаторный элемент (их обозначения DS1—DS5) содержит внутреннюю память, дешифратор и схемы управления. Индикаторные элементы подключаются к шинам адресов, данных и управления микроЭВМ AIM 65 посредством параллельного интерфейсного адаптера R6520(UI), входящего в состав индика-

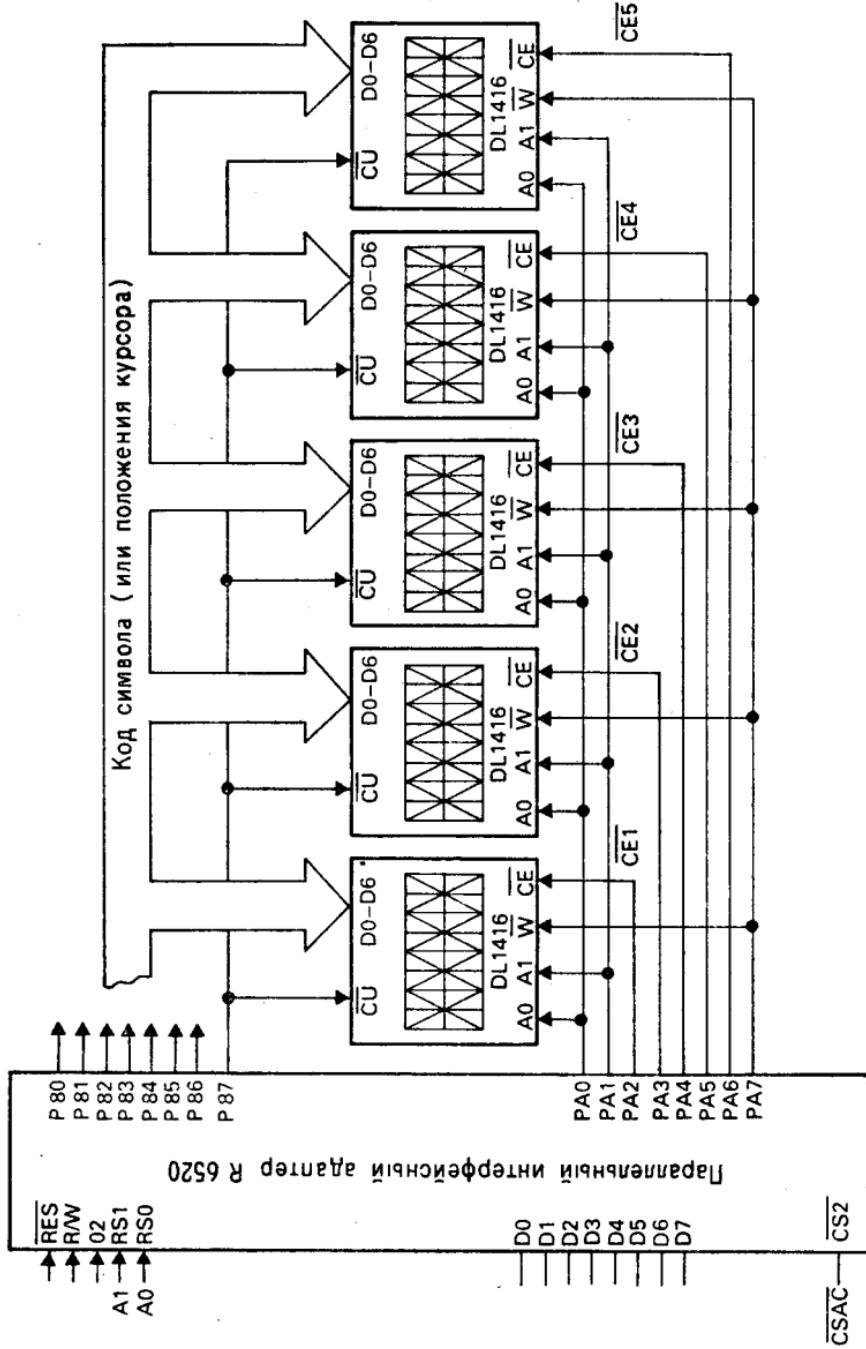


Рис. П2.2. Интерфейс индикатора.

торного модуля. Каждым индикаторным элементом связаны семь линий данных ($D_0 - D_6$), две адресные линии (A_0 и A_1), две линии управляющих сигналов (W и CW) и линия выбора кристалла (CE). Схема индикатора приведена на рис. П2.2.

Для каждого из пяти индикаторных элементов имеется отдельная линия выбора кристалла ($CE_1 - CE_5$). Для загрузки данных в индикаторный элемент соответствующему сигналу CE придается низкий уровень. На линии $D_0 - D_6$ подаются необходимые данные, а на линии адресов $A_0 - A_1$ подается адрес символа в индикаторном элементе. На линии сигнала курсора CU поддерживается высокий уровень. Для записи и индикации данных сигнал низкого уровня подается на линию сигнала записи W . После того как сигнал W снова приобретет высокий уровень, данные на индикаторе сохраняются, пока в индикаторный элемент не поступят новые данные или не будет выведен на индикацию курсор. Ввод данных в индикатор может производиться в произвольные моменты времени, асинхронно по отношению к другим процессам, протекающим в микроЭВМ.

Интерфейс клавиатуры

Подключение клавиатуры к AIM 65 осуществляется через интерфейс R6532. Линии ввода-вывода $PA_0 - PA_7$ интерфейса R6532(Z33) соединяются соответственно с входными линиями $K_{11} - K_{18}$ клавиатуры. Линии $PB_0 - PB_7$ подсоединяются соответственно к выходным линиям клавиатуры $K_{01} - K_{08}$ (рис. П2.3).

При сканировании клавиатуры в поисках нажатой клавиши в разряды выходного регистра А интерфейса R6532 поочередно заносится 0, поступающий в каждый момент времени на одну из линий $K_{11} - K_{18}$. Этот 0 поступает на клавишные переключатели, связанные с линией. Через все клавишные переключатели, подключенные к выбранной в данный момент линии, образуются замкнутые цепи к выходным линиям из числа $K_1 - K_8$, в результате чего возникают логические 0 на выходах соответствующих разрядов выходного регистра В интерфейса R6532. Каждая ненажатая клавиша означает наличие разрыва в цепях связи с выходами $PB_0 - PB_7$ и, как следствие, присутствие логических 1 на выходах соответствующих разрядов выходного регистра В интерфейса R6532.

Одноплатный набор проектировщика SDK-85

В состав этого одноплатного набора проектировщика входят микропроцессор Intel 8085 и ряд обеспечивающих БИС, специально предназначенных для совместного использования с этим

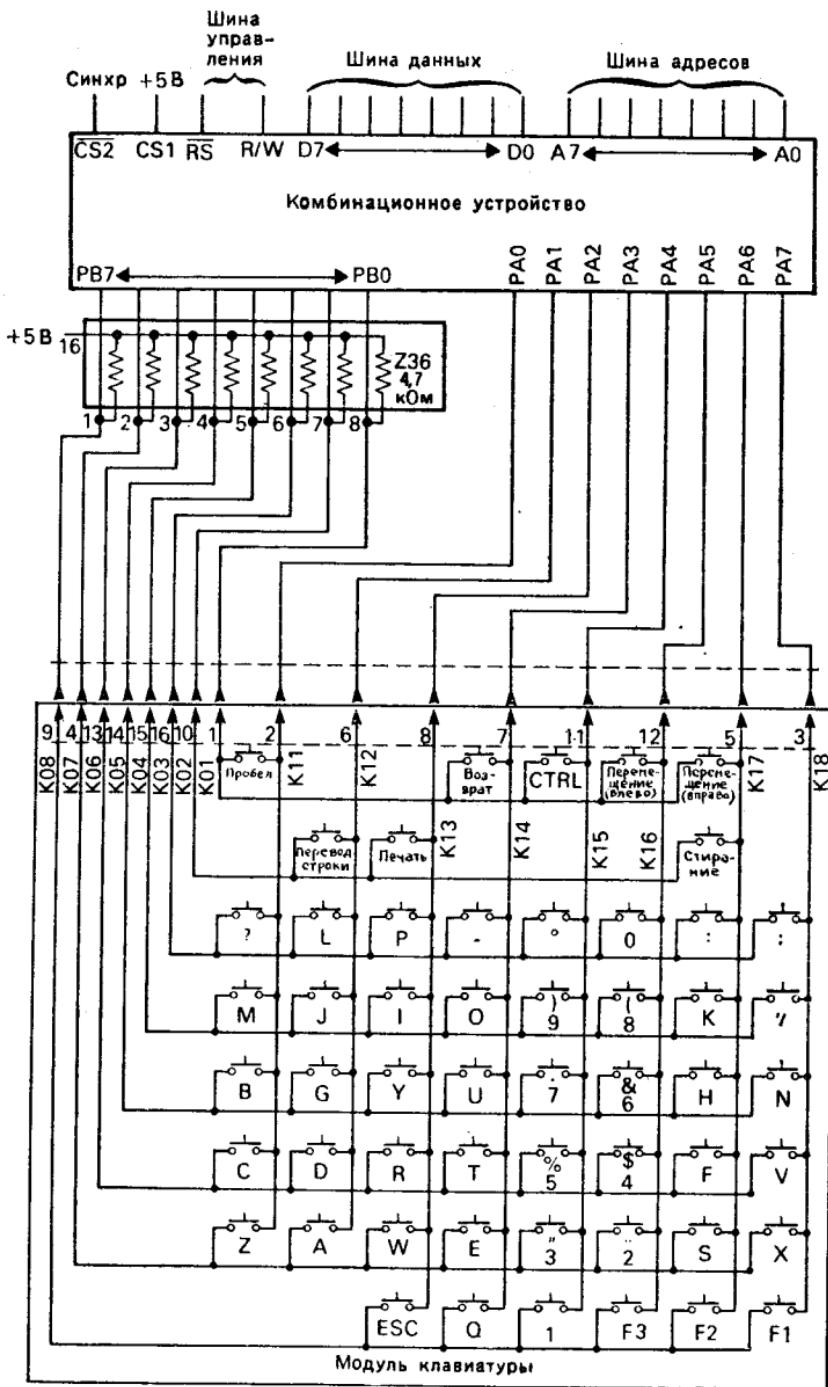


Рис. П2.3. Интерфейс клавиатуры.

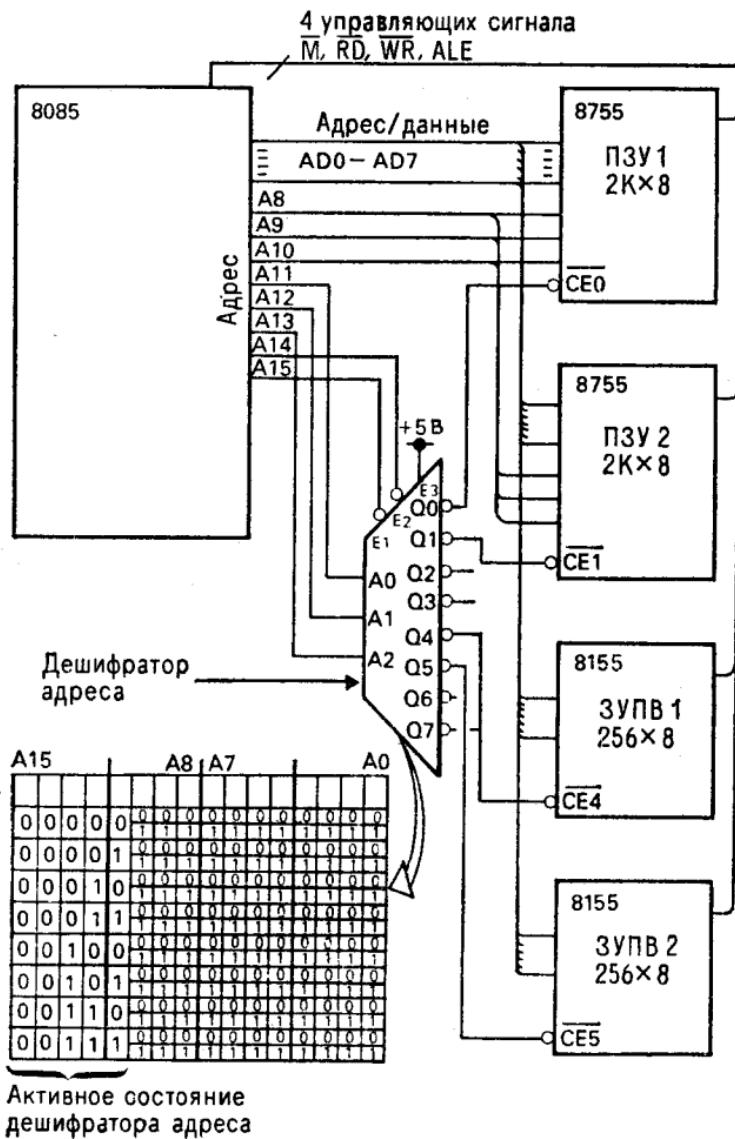


Рис. П2.4. Адресация памяти в SDK-85.

микропроцессором. Базовый набор включает процессор 8085, ПЗУ 8355 объемом 2К байт для размещения программы-монитора, снабженное двумя 8-разрядными портами ввода-вывода, и ЗУПВ с объемом 256 байт, имеющее два 8-разрядных и один 6-разрядный порты ввода-вывода.

Мониторная программа состоит из двух частей, одна из которых позволяет обмениваться информацией с системой посредством встроенной клавиатуры, а вторая предназначена для обмена информацией с устройствами последовательного ввода-

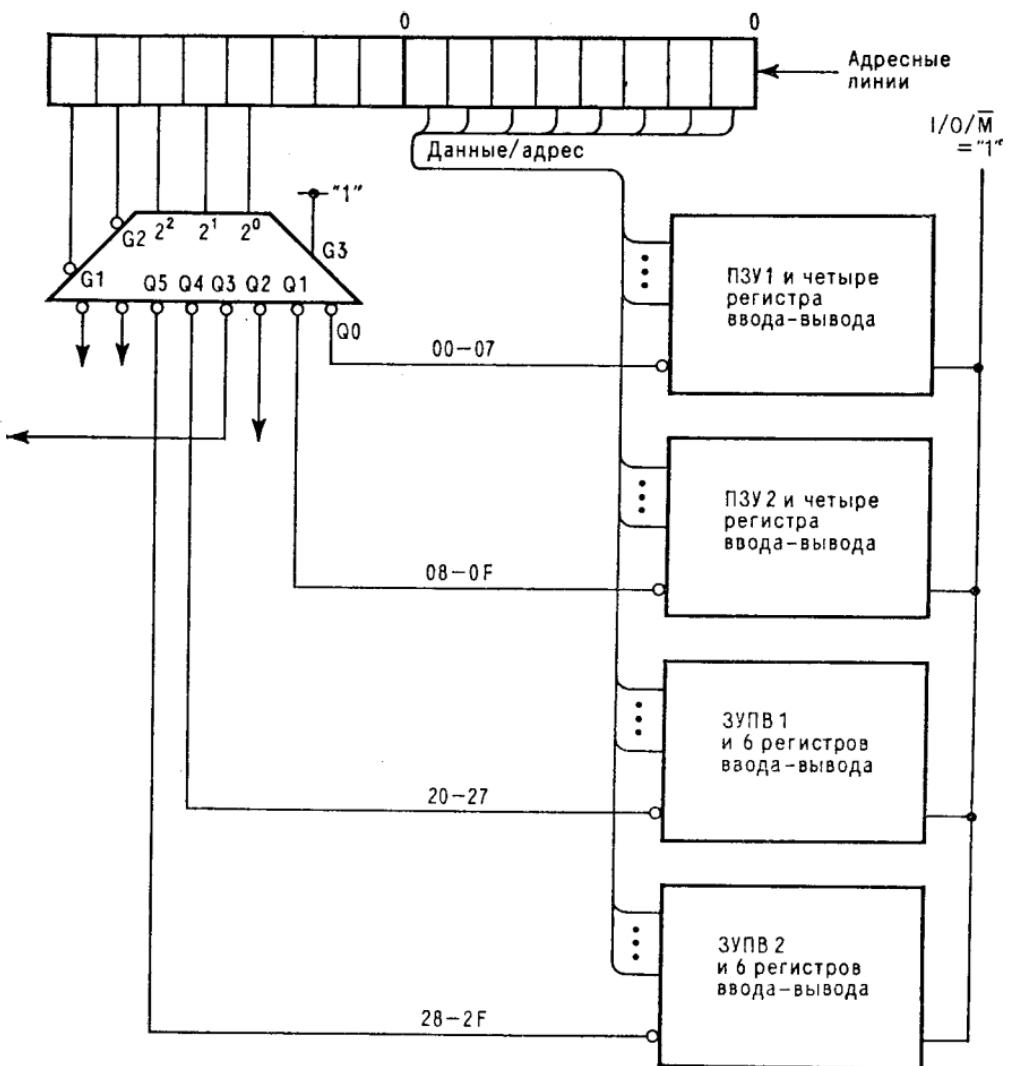


Рис. П2.5. Адресация ввода-вывода в SDK-85. Адреса ввода-вывода дублируются в обеих половинах шины адреса/данных. Линии A8—A15 используются в SDK-85 для обращения к устройствам ввода-вывода с помощью тех же дешифраторов, которые служат для обращения к памяти.

вывода (например, с телетайпом или дисплеем, снабженным клавиатурой) со скоростью 110 бод.

Набор SDK, организация которого иллюстрируется рис. П2.4, П2.5 и табл. П2.1, представляет собой расширенную версию базового набора. В его состав включено дополнительное ЗУПВ 8155, в результате чего общий объем ЗУПВ программ и данных достигает 512 байт, и дополнительное ПЗУ, служащее для размещения построчного ассемблера. Расширение памяти

обусловило увеличение количества портов до восьми 8-разрядных и двух 6-разрядных.

Для работы аппаратуры последовательной передачи данных базового набора необходимо дополнительное напряжение питания 10 В. Для набора, описываемого здесь, это напряжение не требуется, а входная цепь последовательной передачи представ-

Таблица П2.1. Карта распределения адресов памяти и ввода-вывода

(а) Распределение адресов памяти

Адреса	Устройства
0000-07FF 0800-0FFF	ПЗУ монитора ПЗУ ассемблера
1800-1900 2000-27FF 2000-20FF 2100-21FF · · 2700-27FF	Контроллер клавиатуры/индикатора Выбор ЗУПВ1 Первое появление ЗУПВ1 Второе появление ЗУПВ1
2800-2FFFF 2800-28FF 2900-29FF · · 2F00-2FFFF	Выбор ЗУПВ2 Первое появление ЗУПВ2 Второе появление ЗУПВ2
	Восьмое появление ЗУПВ1 Восьмое появление ЗУПВ2

(б) Распределение адресов ввода-вывода

Адреса ввода-вывода 8755	Регистр порта ввода-вывода
Мониторное ПЗУ { 00 Ассемблерное { 08 01 { 01 ПЗУ { 09 02 { 02 { 0A 03 { 03 { 0B	Порт A Порт B DDR A DDR B
Адреса ввода-вывода 8155	
ЗУПВ1 { 20 21 22 23 24 25 { 28 29 2A 2B 2C 2D	Управление/Состояние A B C Мл. разряды счетчика Ст. разряды счетчика

ляет собой пассивную 20-миллиамперную токовую петлю с оптической развязкой. Карта распределения памяти имеет следующий вид:

0000-07FF	ПЗУ монитора
0800-0FFF	ПЗУ ассемблера
1800-1FFF	Программы обслуживания клавиатуры и индикатора
2000-20FF	ЗУПВ
2800-28FF	ЗУПВ

Подробная информация о наборе проектировщика SDK-85 приведена в руководстве для пользователя.

Одноплатный контроллер MEDC Z80

Этот одноплатный контроллер реализован на основе микропроцессора Z80 и устройств ввода-вывода того же семейства БИС. На плате имеются все схемы, необходимые для построения небольших микропроцессорных систем, и разъем, позволяющий использовать плату контроллера в составе больших систем. Разработчики контроллера ставили перед собой задачу облегчить проектирование микропроцессорных систем малого масштаба и создать небольшую систему на базе Z80, которая могла бы применяться для изучения методов внутрисхемной эмуляции при работе с системами проектирования микропроцессорных систем.

Можно считать, что контроллер состоит из трех следующих подсистем:

1. Процессор Z80 и связанные с ним запоминающие устройства со схемами дешифрации и управления.
2. Подсистема ввода-вывода.
3. Интерфейс общего назначения.

Процессор и связанные с ним устройства

Микропроцессор Z80 представляет собой одну из усовершенствованных версий микропроцессора 8080 фирмы Intel; он имеет расширенную систему команд, а доработка аппаратных средств позволяет использовать всего один источник питания +5 В и однофазное тактирование. Машинные программы, написанные для 8080, могут выполняться на Z80, но программы 8080, написанные на уровне языка ассемблера для того, чтобы их можно было обрабатывать с помощью ассемблера Z80, необходимо предварительно перевести на язык ассемблера Z80.

Тактовый генератор состоит из двух инверторов и кварцевого стабилизатора; выходной сигнал генератора буферизируется посредством дополнительного инвертора перед тем, как подается на другие компоненты, установленные на плате (рис. П2.6).

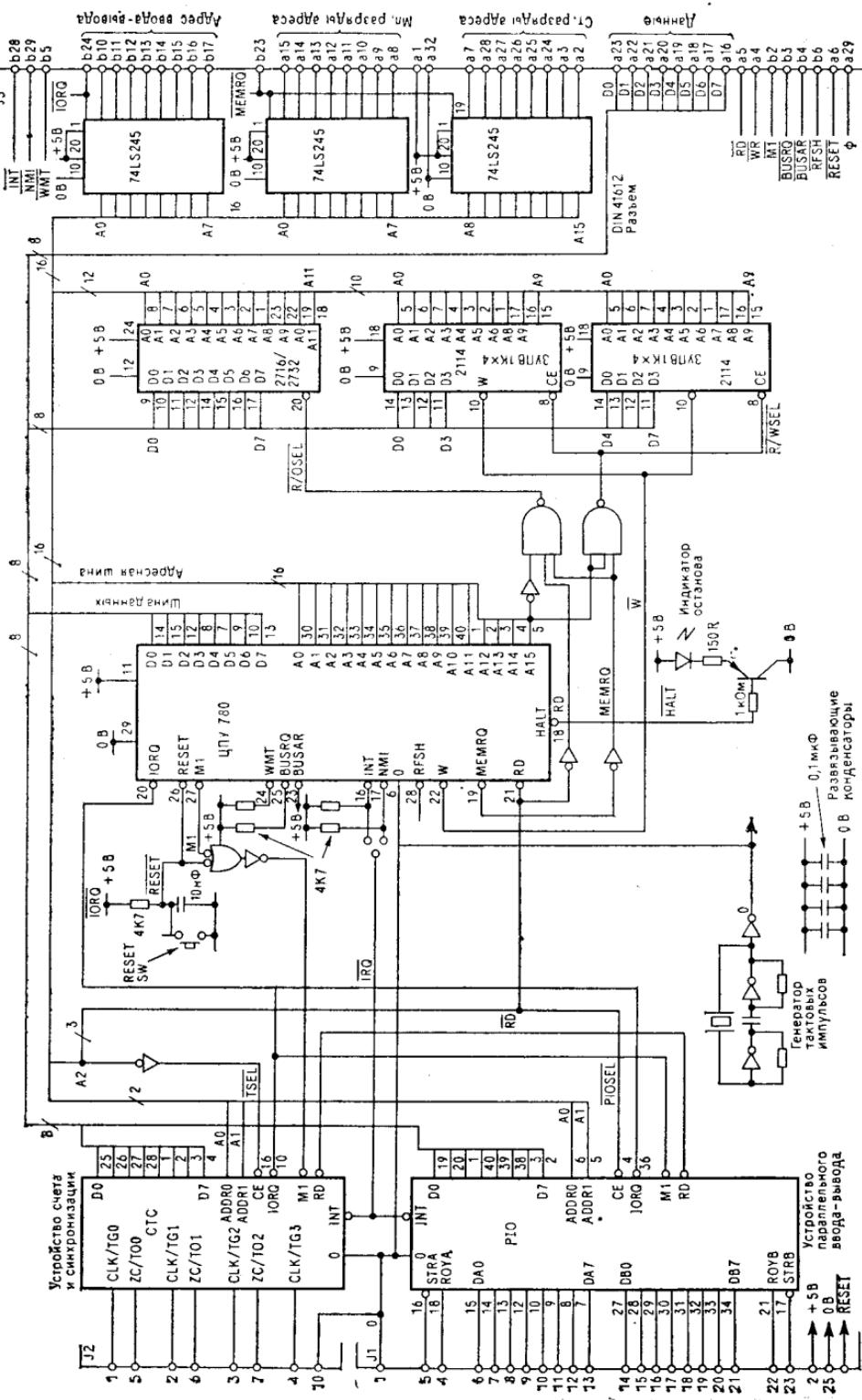


Рис. П2.6. Принципиальная электрическая схема одноплатового контроллера Z80.

Характеристики одноплатных микроЭВМ

На плате имеется кнопка начальной установки, которая приводит в начальное состояние не только процессор Z80, но и устройство ввода-вывода (активизируется их входной сигнал $M1$). ЗУПВ объемом $1K \times 8$ бит реализовано на двух ИС типа 2114.

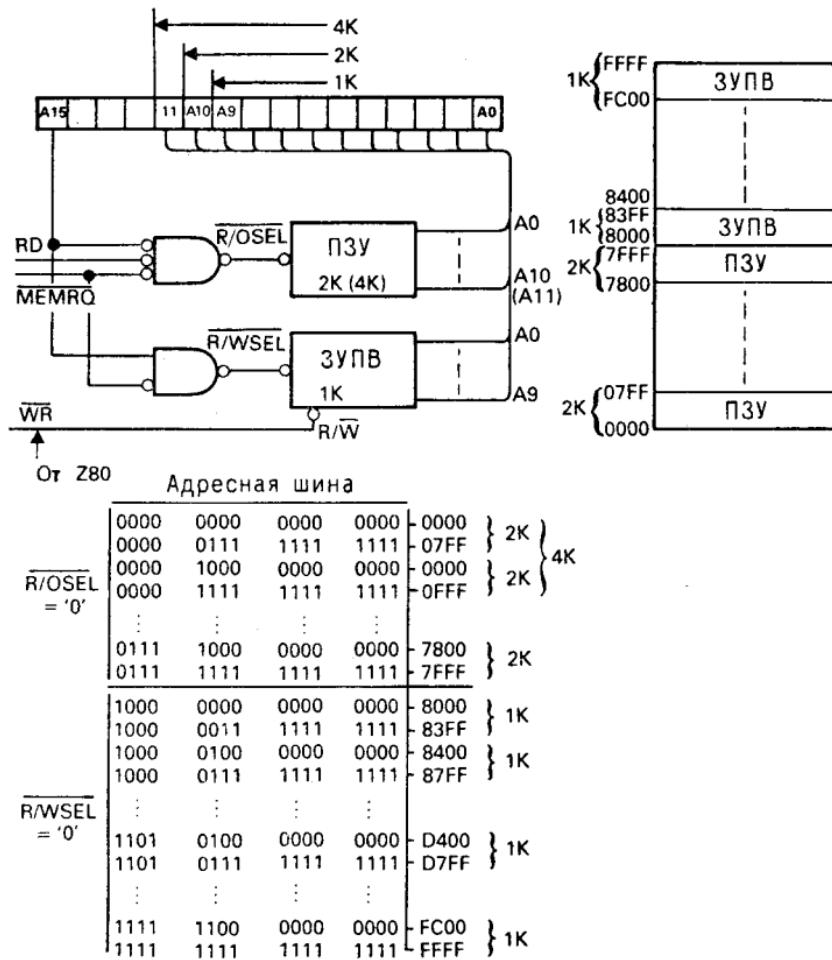


Рис. П2.7. Схема выбора и карта распределения адресов памяти.

В гнездо постоянной памяти могут быть установлены ИС СППЗУ типов либо 2716 ($2K \times 8$ бит), либо 2532 ($4K \times 8$ бит). Логика адресации памяти построена таким образом, что ПЗУ занимает нижнюю часть адресного пространства, а ЗУПВ — верхнюю (рис. П2.7).

При чтении информации из памяти или записи в память приходит в активное состояние процессорный сигнал MEMRQ , который служит для управления двумя вентилями И-НЕ, выходы которых соединены с линиями выбора памяти. Выходные

сигналы этих вентилей могут иметь активные значения, только тогда, когда активен (имеет низкий уровень) сигнал $\overline{\text{MEMRQ}}$.

Управляющий сигнал $\overline{\text{R/WSEL}}$ пребывает в активном состоянии, когда Z80 осуществляет чтение данных ($\overline{\text{RD}} = 0$) и на адресной линии A15 также имеет место логический 0 (что соответствует диапазону адресов 0000—7FFF). Адреса 0000—07FF (или 0000—0FFF при использовании ПЗУ 4Кбайт) охватывают все области ПЗУ. Для обращения по следующим 2К (4К) адресам опять должны быть перебраны все возможные комбинации сигналов на 11 (12) адресных линиях. Таким образом, пока $\text{A15} = 0$, в диапазоне адресов 0000—7FFF происходит повторяющаяся выборка адресов ПЗУ. Только когда процессор выбирает адрес, в котором $\text{A15} = 1$, прекращаются обращения к ПЗУ. Принято считать, что ПЗУ занимает адреса 0000—07FF (0FFF). Сигнал выбора ЗУПВ $\overline{\text{R/WSEL}}$ имеет активное значение (низкий уровень) при $\text{A15} = 1$ и $\overline{\text{MEMRQ}} = 0$ как для чтения, так и для записи. Сигнал W подается прямо на обе ИС ЗУПВ 2114 и управляет направлением передачи данных между Z80 и памятью.

Сигнал $\overline{\text{R/WSEL}}$ имеет активное значение в диапазоне адресов 8000—FFFF, но так как ЗУПВ имеет объем всего 1К байт, набор его адресов многократно (32 раза) укладывается в этот диапазон. Начальными адресами ЗУПВ можно считать 8000, 8400, 8800, ..., FC00, а при разработке программ полагать, что ЗУПВ размещается в диапазоне FC00—FFFF (или, если это удобнее, 8000—88FF).

Обращение к одним и тем же адресам памяти при выдаче микропроцессором различных адресов характерно для систем с простыми средствами дешифрации адресов, в которых сигнал выбора конкретного запоминающего устройства пребывает в активном состоянии в диапазоне адресов процессора, большем, чем количество ячеек устройства. Это грубое внутриплатное декодирование адресов при необходимости использования внешней памяти порождает трудности, связанные с предупреждением конфликтов при использовании шин.

Схемы ввода-вывода

Средства ввода-вывода контроллера включают БИС интерфейса параллельного ввода-вывода (PIO) и БИС счета и синхронизации (CTC). Они предназначены для работы в режиме портовой адресации микропроцессора Z80, при котором пересылка данных в процессор и из него осуществляется путем выполнения команд IN и OUT. При выполнении этих команд производится в активное состояние выходной сигнал $\overline{\text{IORQ}}$ и не по-

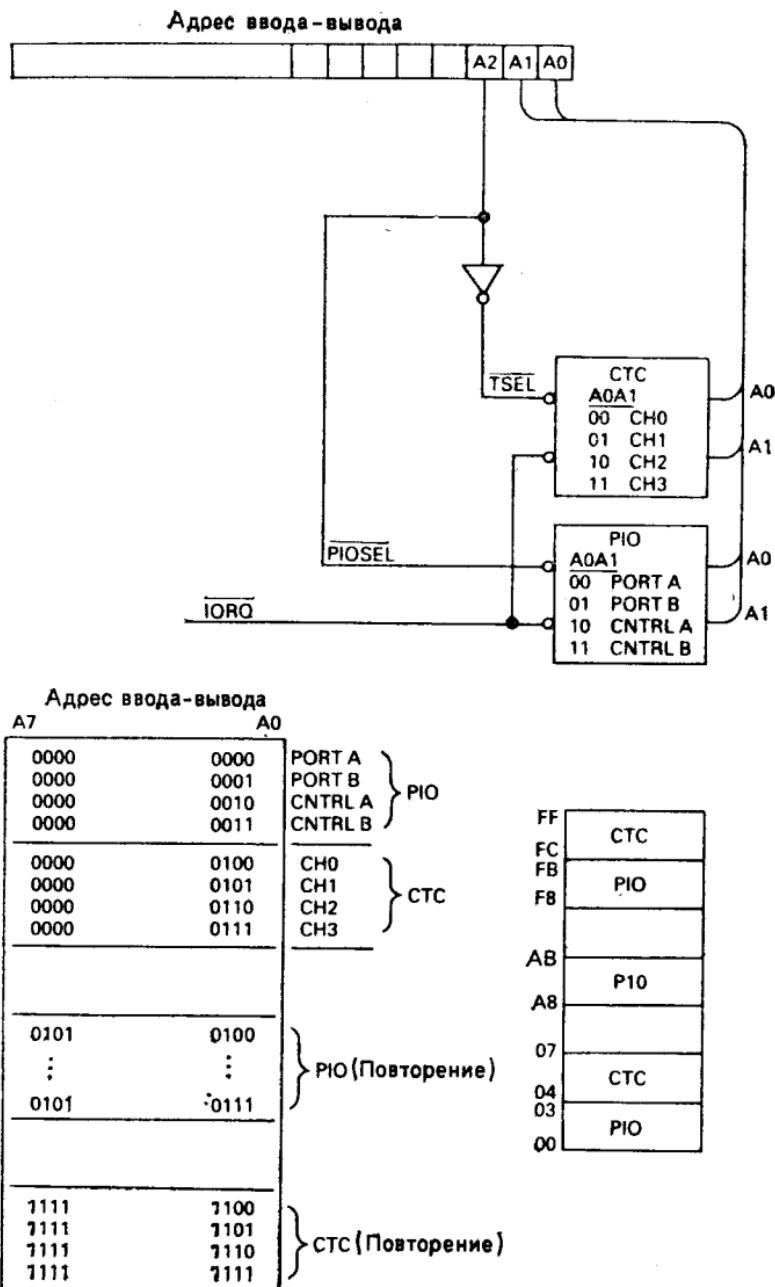


Рис. П2.8. Схема выбора и карта распределения адресов ввода-вывода.

дается сигнал $\overline{\text{MEMRQ}}$, т. е. обмен с памятью заблокирован. Восьмиразрядный адрес порта подается на восемь младших линий адресной шины. Устройства PIO и CTC содержат по четыре адресуемых регистра и поэтому снабжены двумя адрес-

ными линиями, которые соединены с выходами A0 и A1 процессора. Выполнение операций чтения и записи задается одним управляющим сигналом процессора \overline{RD} (рис. П2.8).

Адресная линия A2 используется в схеме адресации ввода-вывода для выбора PIO (при $A2 = 0$) или СТС (при $A2 = 1$). Адреса ввода-вывода 00, 01, 02, 03 и 08, 09, 0A, 0B, ..., F8, F9, FA, FB задают регистры PIO в качестве порта А, порта В, управляющих регистров А и В, т. е. в карте распределения адресов ввода-вывода повторно встречаются одни и те же регистры. Аналогичным образом каналы CH0, CH1, CH2 и CH3 устройства счета и синхронизации СТС отображаются сначала в адреса 04, 05, 06 и 07, затем в OC, OD, OE и OF и т. д. до FC, FD, FE и FF. Первое отображение портов в карте адресов ввода-вывода используется для адресации портов при написании программ.

Интерфейс общего назначения

Интерфейс общего назначения может быть приспособлен для использования в большинстве систем с шинами типа 6500/6800. Адресные сигналы буферизуются с помощью ИС 74LS245, адреса ввода-вывода формируются путем пропускания девяти младших разрядов адреса через вентили, управляемые сигналом IORQ.

При наращивании системы с использованием этого шинного разъема надо соблюдать осторожность, чтобы не возникло конфликтов между внешней памятью и запоминающими устройствами, имеющимися на плате. Так, например, при наращивании ИС ЗУПВ, установленные на плате, необходимо демонтировать.

Приложение 3

Источники и стабилизаторы напряжения питания

Источники питания

Большую часть микропроцессорной системы составляют цифровые схемы, рабочие параметры которых реализуются при колебаниях напряжения питания, не превышающих, скажем, 5 % от номинального значения. Кроме стандартного напряжения питания +5 В могут потребоваться другие, такие, например, как -5, ±12 В и т. д. Они нужны для работы некоторых интегральных схем, аппаратуры передачи данных (портов последовательного интерфейса RS232) или периферийных устройств, таких, как дисплеи и принтеры.

Необходимо каким-то образом стабилизировать напряжение питания, делая его устойчивым к медленным изменениям входного напряжения источника и любым колебаниям тока нагрузки,

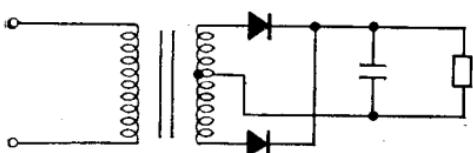


Рис. П3.1. Полнопериодный выпрямитель с трансформатором со средней точкой.

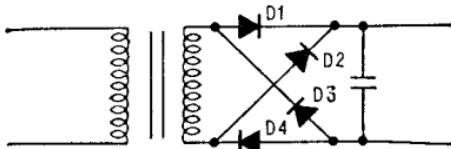


Рис. П3.2. Источник питания с выпрямительным мостом.

имеющим место в процессе функционирования микропроцессорной системы. Кроме того, должны иметься какие-либо средства поглощать или отсекать броски напряжения, появляющиеся при переходных процессах. Перед тем как говорить об обслуживании этой части микропроцессорной системы, кратко остановимся на принципах ее построения (возможно, опытные инженеры и техники захотят опустить этот раздел).

Простейший источник питания представляет собой трансформатор со средней точкой, управляющий полнопериодным выпрямителем (рис. П3.1). Следует заметить, что однополупериодный выпрямитель конструктивно более простой, так как вторичная обмотка трансформатора не имеет отводов и схема содержит всего один диод, но двукратное уменьшение количества пиков

напряжения существенно повышает его пульсацию. Наиболее предпочтительный вариант построения источника питания включает дешевый трансформатор с одной безотводной вторичной обмоткой и выпрямительный мост, реализующий полный период колебаний входного напряжения (рис. П3.2). Пиковое

обратное напряжение каждого диода в такой схеме снижено, но схема имеет и недостаток, заключающийся в том, что в каждом интервале заряда включены в прямом направлении два диода. В качестве напоминания о работе схемы на рис. П3.3 приведены некоторые временные диаграммы.

Выходное напряжение (рис. П3.3, а). Максимальная его величина лишь немного меньше максимального напряжения на вторичной обмотке; эта разница обусловлена падением напряжения на двух проводящих диодах. При небольших токах падение напряжения на диоде является логарифмической функцией тока, увеличиваясь всего на 18 мВ при каждом увеличении тока в два раза. Принято считать, что падение напряжения на кремниевом диоде составляет при малых токах примерно 0,7 В.

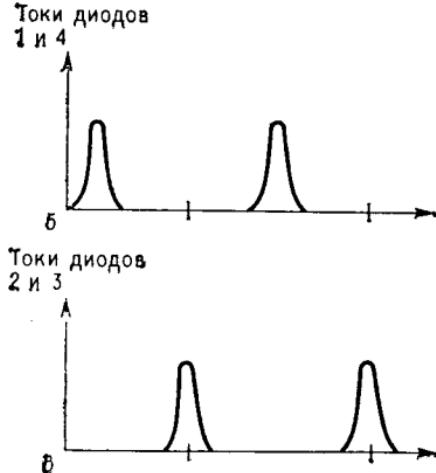


Рис. П3.3. Диаграммы работы источника питания: а — выходное напряжение имеет два пика в течение периода; б, в — каждая пара токов пропускает пиковые токи в один из полупериодов входного напряжения.

При больших токах, имеющих место, когда напряжение заряда близко к максимальному, возникает дополнительное падение напряжения на подложке диодов, прямо пропорциональное величине тока. Указанные обстоятельства затрудняют предварительное определение последствий увеличения тока без детального знания вольт-амперных характеристик диодов. В большинстве случаев удовлетворительные результаты дает принятие величины с запасом 1 В.

Диодные токи (рис. П3.3, б и в). Конденсатор заряжается в течение коротких интервалов пиковыми положительными значениями напряжения и равномерно разряжается через нагрузку в интервалах между этими пиковыми значениями. Чем короче время заряда, тем больше должны быть пиковые значения тока,

а ток этот пропорционален скорости нарастания входного напряжения. Увеличение емкости конденсатора приводит к увеличению тока, укорачиванию времени заряда и, как следствие, уменьшению колебаний напряжения. Задачей разработчика является нахождение компромиссного решения, при котором имеют место минимальные колебания напряжения и не превышаются максимально допустимые токи конденсатора и диодов. Ток протекает в первом полупериоде через диоды 1 и 4, а во втором — через диоды 2 и 3.

Если в каком-либо из диодов происходит обрыв, это может не очень сильно сказаться на среднем значении выходного напряжения, так как источник питания должен быть рассчитан на значительно большее потребление тока, чем то, которое обусловлено реальной нагрузкой. Однако заряд конденсатора будет происходить с вдвое меньшей частотой (в соответствии с диаграммой рис. П3.3, а или б), и для его разряда потребуется вдвое большее время.

Пульсация напряжения

При выходе из строя диода величина пульсации возрастает примерно в два раза. Стабилизатор, стоящий на выходе источника питания, может в значительной степени сгладить эту пульсацию, но необходимо упомянуть наличие минимального мгновенного напряжения, при котором стабилизатор выходит из линейного режима. Когда входное напряжение стабилизатора опускается ниже этого порога, все имеющиеся в нем провалы передаются на выход стабилизатора практически без сглаживания. Этот фактор сразу проявляется очень резко; так, например, пульсация напряжения на конденсаторе, имеющая величину 2 В, может быть сглажена до нескольких милливольт, а пульсация с амплитудой 3 В не может быть снижена до значения, меньшего 1 В (рис. П3.4). Критическим параметром является «дифференциальное напряжение между входом и выходом», которое для большинства стабилизаторов составляет около 2 В, т. е. при номинальном значении выходного напряжения 5 В входное напряжение не должно ни при каких обстоятельствах ни на один момент падать ниже 7 В.

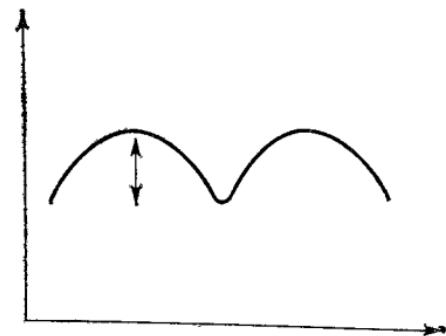


Рис. П3.4. Уменьшение пульсации напряжения за счет использования конденсатора большой емкости и повышения пиковых значений тока.

Для оценки величины пульсации напряжения необходимо сделать следующие допущения:

— ток нагрузки практически постоянен и равен частному от деления пикового значения напряжения на сопротивление нагрузки;

— провалы в напряжении сохраняются в течение целых полупериодов.

Эти допущения позволяют приблизительно рассчитать величину пульсации при плохом сочетании условий работы источника питания, причем сначала можно принять номинальное значение емкости конденсатора, а не минимальное.

$$\text{Ток разряда } (I) \approx \frac{\text{Пиковое напряжение}}{\text{Сопротивление нагрузки}} = \frac{V}{R}$$

$$\text{Скорость падения напряжения } \left(\frac{dV}{dt} \right) = \frac{\text{Потерянный заряд/Время}}{\text{Емкость}} = \frac{I}{C}$$

Процесс продолжается в течение времени t , причем

$$t \approx \begin{cases} 10 \text{ мс} & \text{при } 50 \text{ Гц} \\ 8 \text{ мс} & \text{при } 60 \text{ Гц} \end{cases}$$

$$\text{Величина пульсации напряжения} \approx \frac{V}{R \times C \times t}$$

Если величину пульсации представить в виде доли пикового напряжения, выражение имеет еще более простой вид:

$$\frac{\text{Амплитуда пульсации}}{\text{Пиковое напряжение}} \approx t,$$

где t — интервал между соседними пиками, который представляет собой постоянную времени, определяемую емкостью сглаживающего конденсатора и сопротивлением нагрузки. Приведенные зависимости не предназначены для расчета проектируемых схем, который при всех обстоятельствах должен опираться на тщательный статистический анализ; эти простые формулы должны лишь помочь грубо оценить возможные причины непорядков в системе питания.

Показанный на рис. П3.5 регулятор представляет собой трехполюсник с общей точкой между входом и выходом. Эта точка обычно соединяется с корпусом, который в свою очередь может быть заземлен. (Это делается не всегда, так как проблемы совместного использования устройств, каждое из которых имеет собственное заземление, общеизвестны и для их решения часто оказывается необходимым применение оптических линий передачи сигналов.) Довольно типичным приемом является установка конденсаторов на входе и выходе регулятора в дополнение к основному сглаживающему или накапливающему конденсатору, шунтирующему выпрямитель. Показанные на рис. П3.6 конденсаторы C_1 и C_2 должны быть конструктивно размещены

в непосредственной близости к регулятору. Они могут быть относительно небольшими по величине (меньше 1 мкФ) и служат для образования низкоимпедансных цепей для высокочастотных помех, т. е. шунтирования высокочастотных колебаний, которые могут возникнуть вследствие наличия индуктивностей в линиях связи и в нагрузке. Так, например, накапливающий

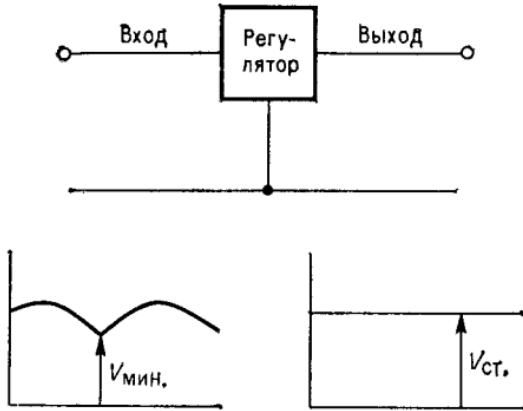


Рис. П3.5. Применение регулятора с тремя внешними выводами для стабилизации напряжения при условии $V_{\text{мин.}} > (V_{\text{ст.}} + 2)$.

конденсатор должен иметь большую емкость и обеспечивать сглаживание выпрямленного напряжения при минимальных физических размерах. На высоких частотах, при которых поведение регулятора сложно и имеется тенденция к возникновению высокочастотных колебаний, заметно проявляется индуктивная составляющая сопротивления этого конденсатора.

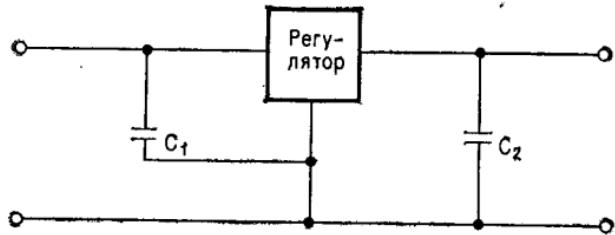


Рис. П3.6. Подавление высокочастотной нестабильности при помощи конденсатора с малой индуктивностью.

Если измерение постоянного напряжения на выходе стабилизированного источника показывает, что имеет место отклонение от номинала, и в особенности если заметна нестабильность постоянного напряжения, быстрая проверка с помощью осциллографа может выявить наличие колебаний мегагерцевого диапазона, являющихся следствием неисправностей высокочастотных развязок. Если осциллографа под рукой нет, то, возможно, установка конденсатора 0,1—1 мкФ в позиции C_1 или

C_2 приведет к видимым изменениям измеряемого напряжения за счет уменьшения или подавления этих высокочастотных колебаний. Любое подобное изменение свидетельствует о том, что возможной причиной искажений выходного напряжения является плохое качество высокочастотных развязок.

Трехполюсные регуляторы первоначально разрабатывались для получения фиксированных напряжений и предназначались главным образом для стабилизации напряжения питания +5 В внутри цифровых систем. Они почти вытеснили схемы, построенные на навесных транзисторах и операционных усилителях. Исключения составляют случаи, когда напряжения, токи или другие параметры схемы имеют очень большие величины.

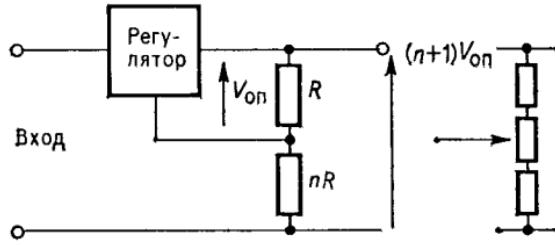


Рис. П3.7. Установка выходного напряжения с помощью резистивного делителя с переменным коэффициентом деления.

Когда простота трехполюсных регуляторов оказалась дополнена их небольшой стоимостью, определяемой массовостью производства, разработчики попытались расширить область их применения. Регулятор, показанный на рис. П3.7, позволяет получать уже не фиксированное напряжение, а напряжение любой величины, лежащей в диапазоне

$$V_{\text{вх.}} - V_{\text{диф.}} > V_{\text{вых.}} > V_{\text{оп.}},$$

где $V_{\text{диф.}}$ — это упоминавшееся выше минимальное значение разности входного и выходного напряжений. С точки зрения теории на работе самой схемы сказывается только разность между первичным и выходным напряжениями регулятора. Стабилизатор с выходным напряжением 5 В и типовым входным, имеющим величину 20 или 30 В, можно применить для получения регулируемого напряжения 100 В из входного, скажем 110—120 В. Это может оказаться рискованным, так как даже при кратковременном коротком замыкании на выходе все первичное напряжение окажется приложенным к стабилизатору. Принцип, иллюстрируемый рис. П3.7, широко используется для построения стабилизированных источников питания с фиксированным или изменяемым выходным напряжением, регулируемым в пределах нескольких десятков вольт. Благодаря появившимся позднее ИС, в которых для генерации малых, но стабильных

опорных напряжений используется так называемый метод энергетической щели, нижняя граница диапазона выходных напряжений опустилась до 1,2 В. При соответствующем первичном питании подобный трехполюсный стабилизатор может вырабатывать выходные напряжения от 1,2 до более чем 20 В.

Не менее важно, что в таких стабилизаторах предусмотрены средства защиты от некорректных условий эксплуатации. Это обстоятельство не следует рассматривать как стимул к их неправильному использованию, но нельзя не отметить, что они, будучи снажены как токовыми, так и температурными ограничителями, защищены гораздо лучше, чем многие устройства

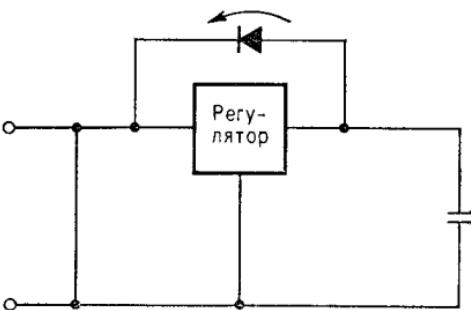


Рис. П3.8. Защита регулятора с помощью шунтирующего диода, через который происходит разряд выходного конденсатора, при коротком замыкании на входе при переходном процессе.

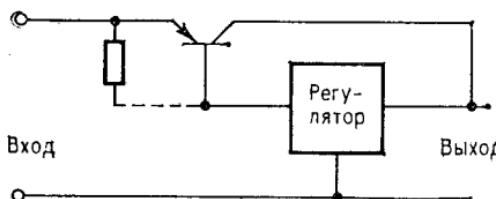


Рис. П3.9. Увеличение выходного тока без изменения напряжения с помощью дополнительного транзистора.

более ранней разработки. В качестве термочувствительного элемента используется специально предусмотренный на кристалле pn -переход, в случае необходимости переводящий стабилизатор в режим малых напряжений. После охлаждения это ограничение может быть снято, причем происходит восстановление исходного напряжения. Возрастание и снижение напряжения, наблюдавшееся в течение нескольких секунд, может означать значительную перегрузку стабилизатора.

Одна из не очень часто встречающихся разновидностей отказа, к которой склонны некоторые виды регуляторов, требует добавления к схеме внешнего диода, как показано на рис. П3.8. Причиной этого вида отказов является короткое замыкание на входе регулятора, т. е. кратковременное замыкание накапливающей емкости. При этом энергия, накопившаяся на выходном конденсаторе регулятора, может разрядиться через внутренние переходы регулятора. Для исключения этой ситуации можно использовать внешний диод, образующий цепь обхода регулятора. Короткие замыкания на выходе регулятора представляют

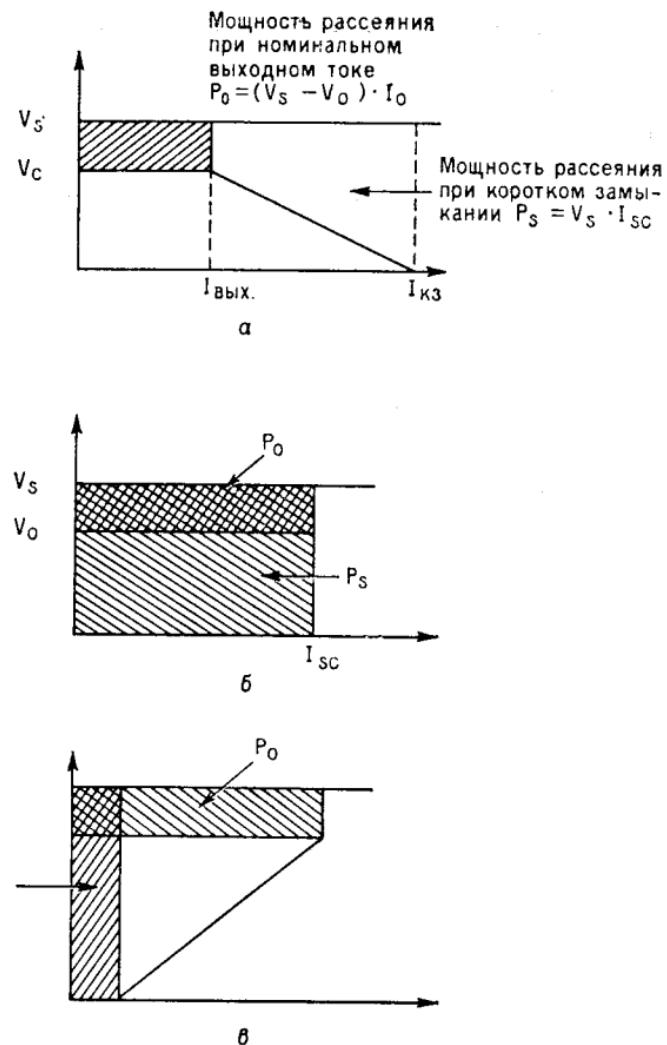


Рис. П3.10. Рассеяние мощности в регуляторах напряжения: а — площадь прямоугольников соответствует мощности, рассеиваемой в номинальном режиме и в режиме короткого замыкания; б — ограничение тока снижает избыточное рассеяние; в — уменьшение тока короткого замыкания и рассеяния мощности за счет ограничения свертыванием.

меньшую опасность благодаря наличию встроенных токовых и температурных ограничителей.

Еще одна доработка схемы, к которой проявляют интерес многие пользователи, направлена на достижение возможности увеличения выходного тока без потери стабилизированности напряжения. В добавляемом для этой цели транзисторе входной ток регулятора используется в качестве базового, а коллекторный ток транзистора — для питания нагрузки (рис. П3.9). Подобная схема позволяет значительно увеличить выходной ток регулятора, причем это увеличение может меняться в зависи-

мости от коэффициента усиления транзистора по току. Таким образом, в схеме утрачивается возможность точного ограничения выходного тока, для восстановления которой требуется использование большего числа транзисторов.

Рисунок П3.10 позволяет в первом приближении оценить параметры, характеризующие рассеяние мощности. Выходное напряжение простого регулятора, не снабженного ограничителем тока, остается постоянным, пока не достигается максимальное предельное значение выходного тока I_0 (рис. П3.10, а). Когда сопротивление нагрузки уменьшается до нуля, что соответствует режиму короткого замыкания, ток возрастает до существенно большей величины I_s . Рассеяние мощности в последовательном устройстве питания при максимальном потреблении по выходу зависит от разности первичного и выходного напряжений; оно показано на рисунке в виде заштрихованного прямоугольника. Прямоугольник гораздо большей площади представляет мощность, рассеиваемую в режиме короткого замыкания. Его размер несколько завышен в предположении, что первичное напряжение остается неизменным. Однако в любом случае рассеяние мощности в режиме короткого замыкания превышает рассеяние в нормальном режиме максимального потребления тока примерно на порядок. Ни применение очень больших транзисторов, ни надежда на то, что коротких замыканий не будет, не являются выходом из положения. Простое ограничение тока, иллюстрируемое рис. П3.10, б, снижает избыточное рассеяние во много раз, так как ток короткого замыкания при этом лишь на 10—20 % превышает расчетное значение выходного тока. Еще лучшие результаты дает прием, результаты применения которого представлены на рис. П3.10, в. Это так называемое свертывание, или возврат, при котором множественные цепи обратной связи понижают величину тока короткого замыкания. Заштрихованные на этом рисунке области, характеризующие рассеяние мощности при номинальном выходном токе и токе короткого замыкания, имеют примерно одинаковую площадь.

Принцип построения регуляторов, в том числе трехполюсных, иллюстрируется рис. П3.11. Напряжение, снимаемое с диода Зенера или другого устройства, формирующего стандартное напряжение, сопоставляется с частью выходного напряжения, которая образуется с помощью делителя. Причинами некорректности этого сопоставления могут быть

- изменения первичного напряжения, передающиеся на зенеровский диод;
- токи, ответвляющиеся от зенеровского диода и делителя выходного напряжения в схему сравнения;
- температурные изменения опорного напряжения и усилителей схемы сравнения.

Компоненты, показанные на рисунке, встроены в полупроводниковую схему трехполюсного регулятора, т. е. не могут быть изменены пользователем. Поддержание тока нагрузки обеспечивается применением организации схемы, показанной на

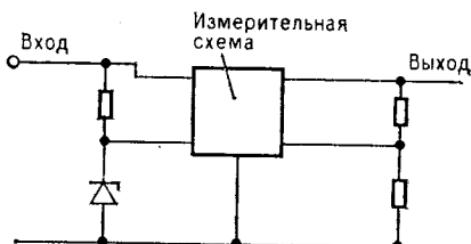


Рис. П3.11. Сравнение части выходного напряжения с опорным для управления выходным напряжением.

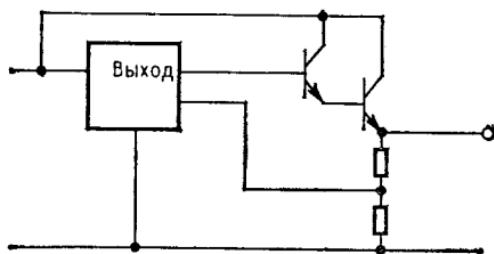
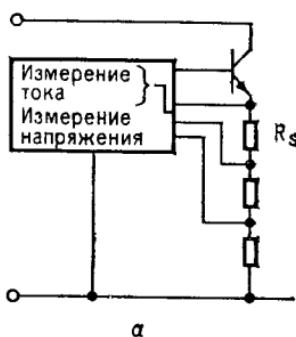
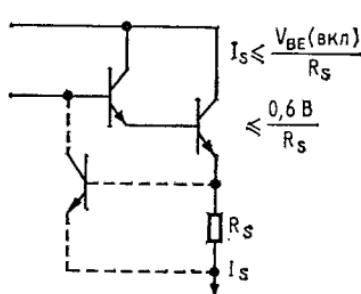


Рис. П3.12. Увеличение выходного тока через делитель, служащий для измерения напряжения нагрузки.

рис. П3.12. Между регулятором и нагрузкой устанавливаются эмиттерные повторители. Напряжение на делителе при этом несколько смещается при неизменной схеме формирования опорного напряжения. При достаточно больших токах нагрузки дополнительное падение напряжения на базово-эмиттерных пере-



а



б

Рис. П3.13. Измерение выходного тока с целью его ограничения: а — применение измерительного сопротивления с интегральным регулятором; б — добавление измерительных функций к существующему регулятору.

ходах увеличивает минимальную величину первичного напряжения при заданной величине выходного примерно на 1,4 В.

Ограничение выходного тока осуществляется за счет измерения тока нагрузки с помощью схемы, показанной на рис. П3.13, а. Падение напряжения на последовательно включенном сопротивлении R_s пропорционально току нагрузки. Это напряжение сравнивается с опорным так же, как это делается при измерении выходного напряжения. Особой точности при измерении тока нагрузки не требуется, поэтому роль сопротивления R_s

может с успехом выполнять базово-эмиттерный переход транзистора, как показано на рис. П3.13, б.

Ограничение тока может начать сказываться уже при 10—20 % -м превышении пикового номинального значения выходного напряжения регулятора. Низкое выходное напряжение может явиться следствием относительно небольшого возрастания тока нагрузки из-за подключения дополнительного периферийного устройства или установки в схему ИС, которая потребляет ток, несколько превышающий номинальный.

Трудно поддаются локализации неисправности системы питания, в которой используется несколько регулируемых напряжений с общими или получаемыми одно из другого опорными напряжениями. Избыточная нагрузка по одному из регулируемых напряжений может неожиданным образом сказаться на других.

Вышеприведенные рассуждения затрагивали поведение схем источников питания в статике в значительной степени потому, что оно более предсказуемо, чем работа в динамике. Описанные причины неисправной работы достаточно просто поддаются обнаружению и устранению.

В реальных условиях потребление тока как логическими схемами, так и периферийными устройствами имеет существенную переменную составляющую. Выходное напряжение стабилизированного источника, успешно поддерживаемое при более или менее постоянном токе нагрузки, может иметь существенные провалы при переходных токах значительно меньшей величины. Для выявления бросков на линиях питания и земли необходимо использовать осциллограф. Общепринятым средством подавления этих помех является локальная распределенная развязка питания логических ТТЛ-схем; для ТТЛ эта проблема является особенно острой, так как при изменении логического состояния ТТЛ-элемента оба последовательно соединенных транзистора его выходного каскада на короткое время оказываются одновременно открытыми и в этот момент имеет место большое импульсное потребление тока.

Коммутируемые регуляторы

При использовании линейных регуляторов управляющий сигнал непрерывно поступает на силовой узел источника, изменяя его выходные параметры в соответствии с изменениями внешних условий. Силовой узел вынужден затрачивать на это значительную часть полной рассеиваемой мощности, соответствующую разности первичного и выходного напряжений.

В соответствии с совершенно другим подходом между источником и нагрузкой устанавливается коммутатор, который варьи-

рует долю времени, в течение которого источник и нагрузка соединены между собой. Во многих ситуациях для этого не требуется никаких дополнительных компонентов; так, например,

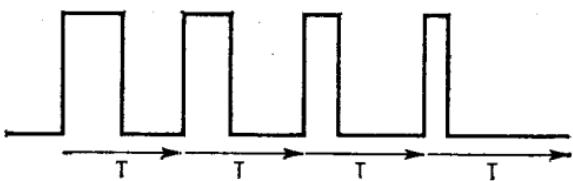


Рис. П3.14. Управление средним значением напряжения путем изменения коэффициента заполнения сигнала.

мощностью, потребляемой резистивной нагрузкой, теоретически можно управлять совсем без потерь, так как идеальный ключ не пропускает тока в разомкнутом состоянии, а в замкнутом со-

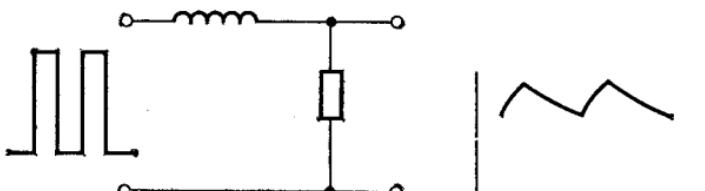


Рис. П3.15. Уменьшение переменной составляющей с помощью индуктивности без существенного изменения среднего значения напряжения.

стоянии на нем нет падения напряжения. Общий принцип широтно-импульсной модуляции поясняется рис. П3.14.

Для многих нагрузок необходимо, чтобы напряжение подвергалось как можно меньшим изменениям. Первым шагом

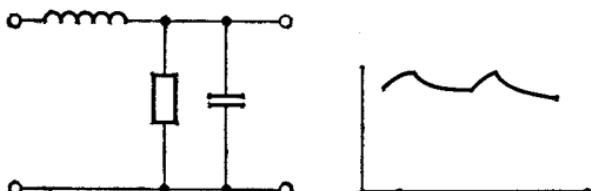


Рис. П3.16. Дальнейшее уменьшение пульсации с помощью емкостного фильтра.

к обеспечению этого является введение последовательной индуктивности (рис. П3.15). Если опять предположить, что компоненты имеют идеальные характеристики, можно утверждать, что индуктивность, высвобождающая запасенную энергию, когда приложенное к ней напряжение равно нулю, не рассеивает мощности. Пульсация напряжения, подающегося на нагрузку, уменьшается за счет увеличения либо частоты, либо индуктивности. Дополнительное снижение пульсации достигается с помощью фильтра, представляющего собой конденсатор, который

включен параллельно нагрузке (рис. П3.16), но для сведения пульсаций к милливольтовым величинам обычно используется еще и линейный регулятор. Его применение уже влечет за собой небольшие потери, поэтому выходное напряжение коммутируемого регулятора должно быть несколько выше нужного окончательного напряжения схемы.

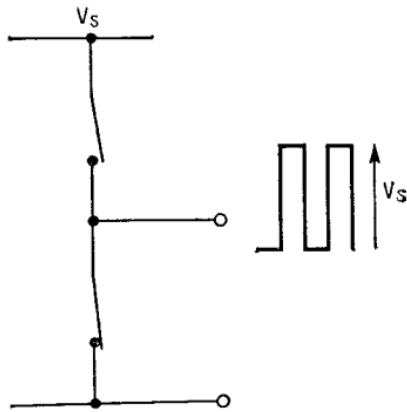


Рис. П3.17. Управление выходом схемы с помощью двух противофазных переключателей.

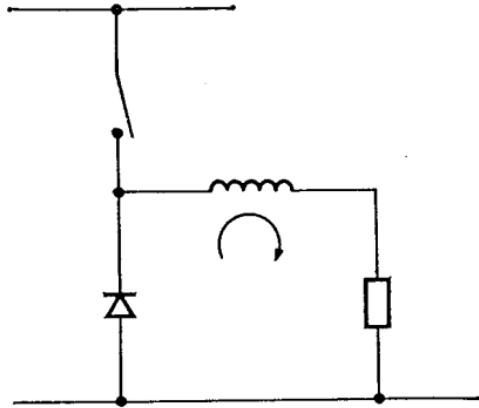


Рис. П3.18. Использование диода вместо одного из переключателей.

Коммутацию выхода источника постоянного напряжения можно осуществлять с помощью двухпозиционного переключателя, но такое устройство невозможно непосредственно реализовать средствами интегральной технологии. Просто оставлять выход источника питания «подвешенным» в состоянии обрыва недопустимо, так как при этом образуется избыточная обратная ЭДС. Одним из путей решения этой проблемы является применение пары двухпозиционных переключателей, работающих в противофазе (рис. П3.17). Очень важно при этом точное выдерживание временных соотношений, так как, если оба переключателя одновременно окажутся замкнутыми, произойдет короткое замыкание источника питания, а если оба одновременно разомкнутся, прервется протекание индуктивного тока. Для предотвращения возникновения подобных ситуаций один из переключателей обычно заменяют диодом (рис. П3.18). Этот так называемый «инерционный» диод обеспечивает поддержание тока при разомкнутом состоянии переключателя. Недостатком схемы с диодом является падение напряжения на диоде, которое увеличивает энергетические потери. Тем не менее коэффициент полезного действия такой схемы значительно выше 80 %, даже когда выходное напряжение является лишь незначительной частью входного напряжения регулятора, тогда как эффективность линейного регулятора не превышала бы в этой ситуации

ции 50 %. Протекание тока имеет одинаковый характер в обеих частях цикла. В качестве переключателя может быть применен

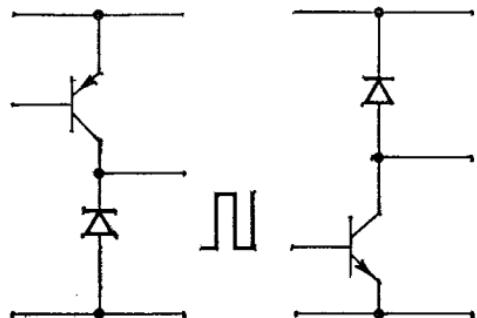


Рис. П3.19. Использование транзистора в качестве управляемого переключателя.

$p-n-p$ - или $n-p-n$ -транзистор (рис. П3.19). Когда транзистор открыт, падение напряжения может быть $\ll 1$ В, но снижение КПД на несколько процентов все-таки происходит вследствие наличия базового тока. В полевых транзисторах соответствующие токи затворов отсутствуют, но некоторые труднения вызывает быстрое приведение их в состояние проводимости.

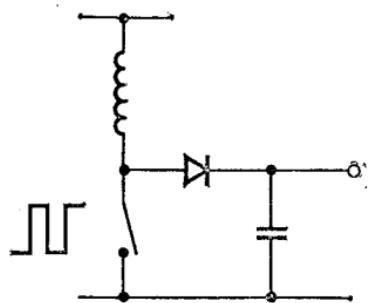


Рис. П3.20. Модификация схемы, позволяющая увеличить выходное напряжение.

Как видно из схемы на рис. П3.20, коммутируемый выход не обязательно понижает поступающее на него напряжение. В этой схеме индуктивности оказывается недостаточно для сглаживания выходного сигнала, и она должна быть дополнена сглаживающим конденсатором. Во всех приведенных примерах были рассмотрены лишь выходные каскады регуляторов. Предшествующие им управляющие каскады выполняют две функции. Как и во всех регуляторах, один из них проводит сравнение выходного напряжения с опорным и усиление получающейся разности. В соответствии с этим напряжением ошибка должна формироваться последовательность импульсов с переменным отношением длительности импульса и паузы. Один из способов решения этой задачи состоит в сравнении этого сигнала ошибки с пилообразным напряжением (см. рис. П3.21). Если схема сравнения имеет достаточно высокий коэффициент усиления, она вырабатывает сигнал, который может управлять электронным коммутатором. С помощью должным образом организованной глубокой обратной связи можно удерживать входной сигнал схемы сравнения на уровне пилообразного напряжения, обеспечивающем получение необходимого коэффициента заполнения выходного сигнала.

Такие устройства особенно широко применяются для коммутации в магистральных источниках питания с высоким переменным первичным напряжением. Их популярность выросла благодаря возможности относительно компактной реализации при заданном значении выходного напряжения, но коммутирующие устройства должны выдерживать высокое первичное напряжение при переключении с большой частотой. Должна также

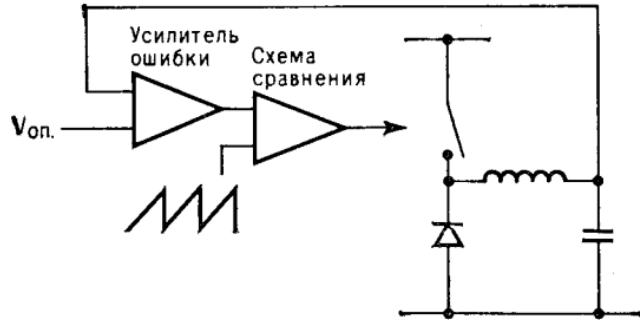


Рис. П3.21. Использование глубокой обратной связи для сравнения выходного напряжения с опорным и управления выходным коммутатором.

быть предусмотрена защита от переходных напряжений, генерируемых индуктивными элементами системы, как это сделано в схеме, приведенной на рис. П3.22. Схема содержит две выпрямительно-сглаживающие секции. Первая вырабатывает высокое постоянное напряжение из первичного переменного, а вторая служит для получения низкого постоянного напряжения из выходного сигнала первой, воспринимаемого с помощью трансформаторной связи. В схеме используется высокая частота коммутации, что позволяет применить для построения сглаживающих цепей конденсаторы небольшой емкости, а также малогабаритные трансформаторы и индуктивности.

Виды неисправностей коммутируемых регуляторов могут отличаться от тех, которые характерны для линейных схем, хотя, как показано выше, в регуляторе всегда имеются линейные каскады. Частота коммутации составляет десятки — сотни килогерц, и поэтому всегда существует проблема наводок, связанных с излучением и паразитными связями. Хотя многое может быть предусмотрено на этапе проектирования, неожиданную чувствительность к помехам могут проявить другие узлы системы питания, как, например, выходной выпрямительный каскад. Характеристики трансформаторов на ферритовых сердечниках и индикаторов существенно меняются при неправильной установке (слишком сильном или слабом креплении) и при вибрации, так как в этих ситуациях возможен перегрев. При неисправности управляющих каскадов коммутатор может оказаться в постоянно замкнутом состоянии, при этом естественно ожидать появления избыточных напряжений или полного выхода из строя источника питания.

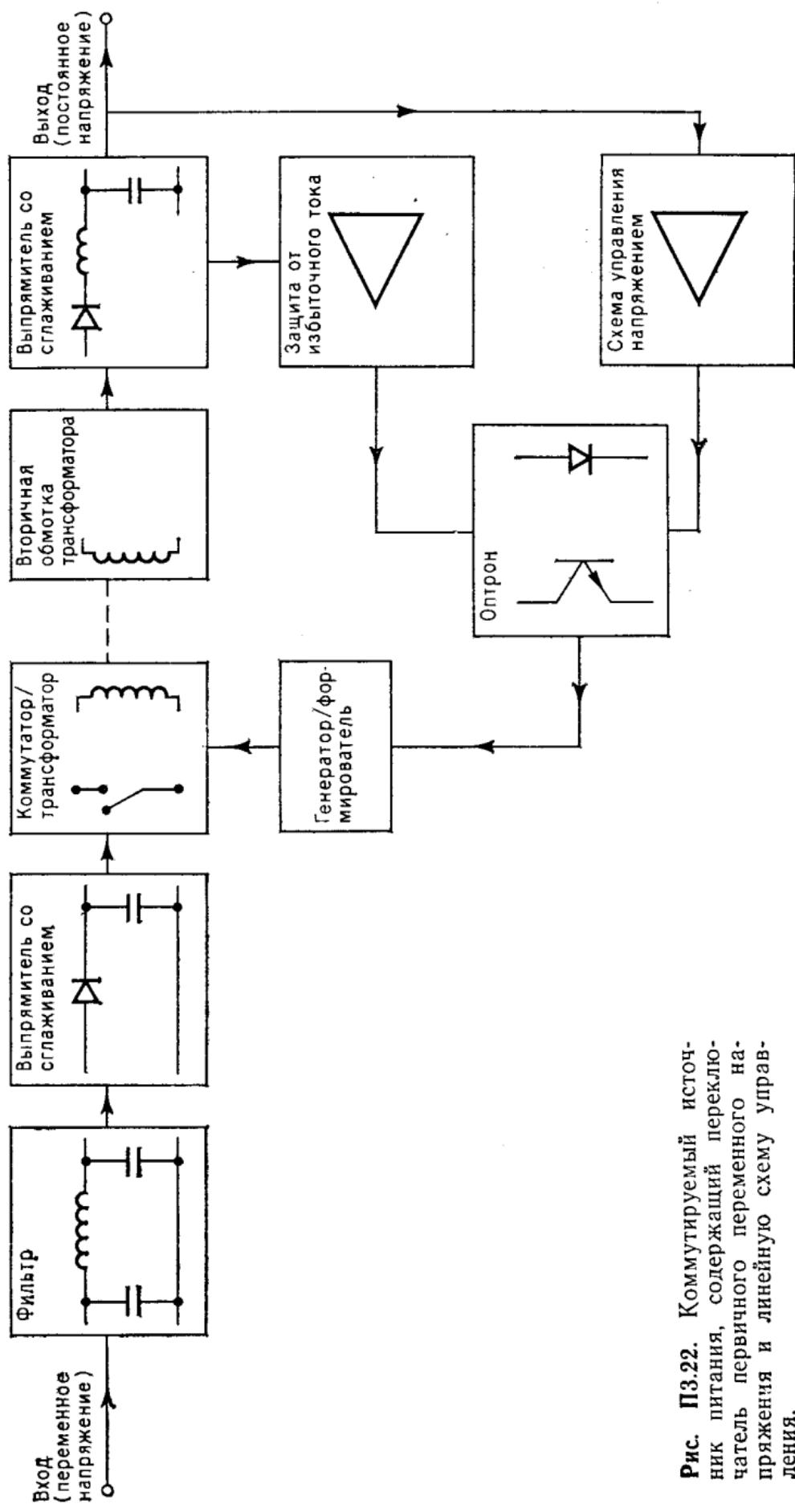
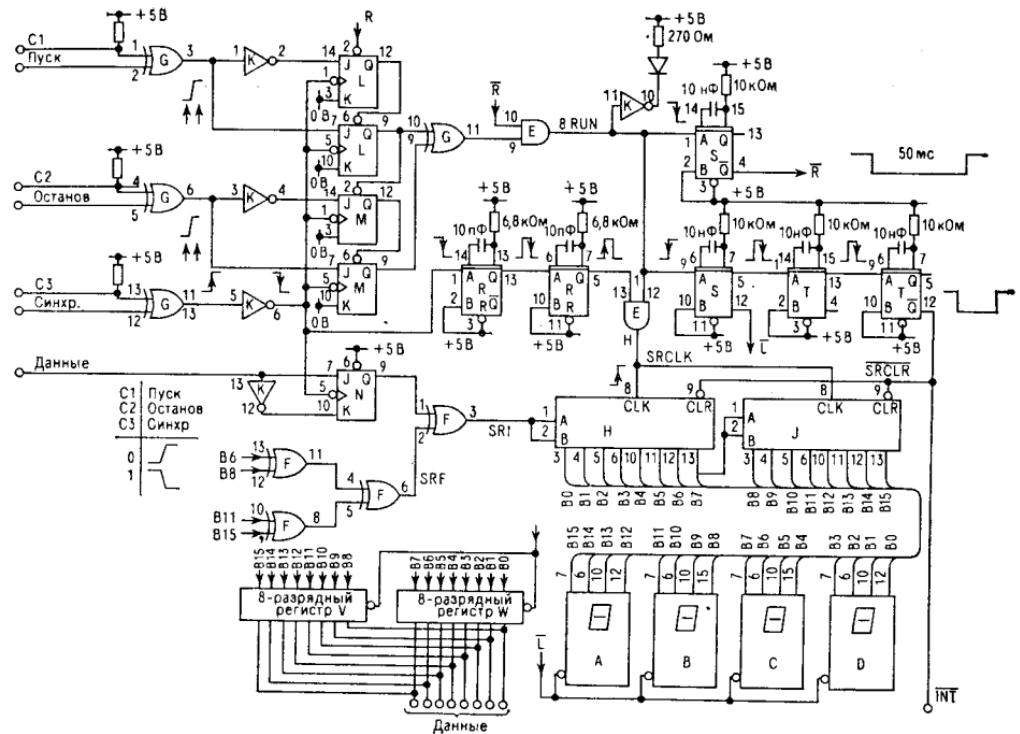


Рис. П3.22. Коммутируемый источник питания, содержащий переключатель первичного переменного напряжения и линейную схему управления.

Приложение 4

Варианты схем построения сигнатурных и логических анализаторов



Элемент	Тип	Функция	+ 5 В	0 В
A, B, C, D	TIL308	Светодиодный индикатор	16	8
E	7408	Счетвереный двухходовой вентиль И	14	7
F, G	7486	Счетвереный вентиль		
H, J	74164	ИСКЛЮЧАЮЩЕЕ ИЛИ	14	7
K	7414	8-разрядный сдвиговый регистр	14	7
L, M, N	7473	Шестнадцатеричный инвертор	14	7
R, S, T	74123	Сдвоенный JK-триггер	4	11
V, W	74373	Сдвоенный формирователь одиночного импульса	16	8
		8-разрядный регистр		

Рис. П4.1. Одноплатный сигнатурный анализатор.

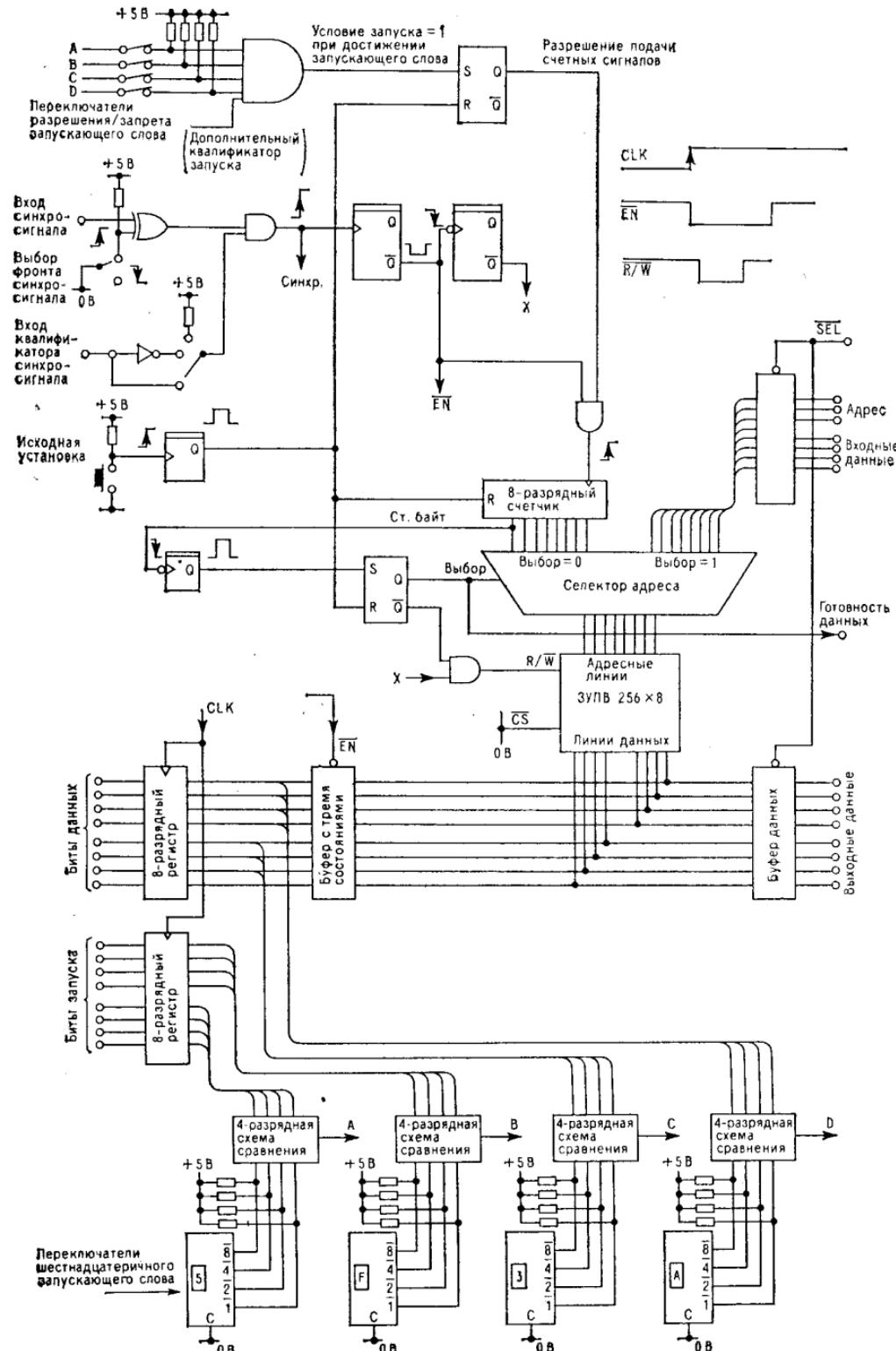


Рис. П4.2. Одноплатная приставка для логического анализа.

Предметный указатель

- Адаптер 204, 210
- Адрес 56, 271, 295
- Аккумулятор 65
- Алгоритм проверки 128
- Анализ
 - временных диаграмм 144
 - доменов данных или состояний 144
 - логический 117, 142, 145
 - сигнатурный 117, 178, 241
- Аналогово-цифровой преобразователь 37
- Арифметико-логическое устройство 65

- Бит четности 114, 155
- Блок данных 97
- Буфер двунаправленный 121
- Буферизированный сигнал 44
- Буферный каскад 28

- Ввод-вывод 74, 310
- Вентиль 14
- Воздействие
 - входное 119
 - стимулирующее 188
 - тестовое 204
- Восьмеричная запись 44
- Выход буферизированный 40
 - двухтактный 20
 - с тремя состояниями 28, 36

- Генератор символов 157, 256
 - тактовый 16, 40, 117

- Дарлингтоновская пара 21
- Двухпроводная схема 111
- Декодирование 52
- Декремент 69
- Дешифратор 51, 55
- Динамическая МОП-структура 19
- Диод защитный 33
 - Шоттки 23
- Дисассемблер 144
- Дисковая память 126
- Дифференциальная система передачи 113
- Достоверный сигнал 27

- Запоминающее устройство
 - полупроводниковое 51
 - постоянное 52, 56, 135
 - программируемое 52
 - с произвольной выборкой 35, 53, 131
- Запускающее слово 146, 156, 256
 - событие 171

- Импеданс 15, 34, 42
- Инвертор
 - сложный 21
 - управляемый 245
 - шестнадцатеричный 19

- Индикатор 11, 78, 213, 300
- Инкремент 61, 69
- Интерфейс 18, 39, 78, 154, 213, 300, 312
- Информация 9
- Источник питания 121, 313

- Карта распределения памяти 54, 208
- Квалификатор 145, 149, 198
- Квитирование 80, 141
- Клавиатура 150, 208, 213, 302
- Клипсы логические 17
- КМОП-логика 33, 254
- Колодка свободного прогона 119
- Команда 38, 61, 69, 273, 279, 285, 296
- Коммутатор 204, 323
- Компаратор 18
- Конвертер 262
- Конденсаторы развязывающие 27
- Контроллер 42, 216, 307
- Контроль передачи 79
- Короткое замыкание 10, 318

- Линейный режим 28
- Логика инжекционная 31

- Маскирование прерываний 73
- Масочное программирование 52
- Матрица переключателей 151
- Методы тестирования 9
- Микроконтроллер 39
- Микропроцессор 39, 61, 267
- Многофункциональная схема 61
- Модель программная 295
- Монтажное ИЛИ 17, 21
- МОП-структуры 32
- Мультиплексирование 42, 50, 121

- Набор проектировщика 302
- Нагрузка логического элемента 21
- Напряжение питания 269
 - уровень 9
- Настроочный код 218
- Насыщение 22
- Начальная установка 44, 118, 245
- Неустойчивость сигнатуры 181

- Область памяти 41
- Обратный отсчет 146
- Обрыв 10, 29
- Однокристальный МП 39
- Одноплатная ЭВМ 196, 208, 224, 297
- Одноплатный анализатор 240
- Окно сигнатурное 180, 197, 221
- Ошибки пользователей 127

- Память 39, 57
- Пересылка данных 46
- Переход 72

Плотность компоновки 33
 Помеха 125
 Помехоустойчивость 27
 Порт ввода-вывода 71, 79, 141, 208
 Преобразование интерфейсное 249
 — формы данных 97
 Прерывание 46, 64, 73, 268
 Принтер 126, 208
 Пробник 12, 167, 206
 Прогон свободный 119, 171, 185
 Программа 39, 304
 Программатор 166
 Прямой доступ к памяти 42
 Пульсатор 15

Развязка питания 27, 323
 Регенерация памяти 57, 59, 118
 Регистр
 — команд 61
 — флагов 67
 Регулятор напряжения 316, 323

Сдвиг циклический 71
 Серия импульсов 13
 Сигналы микропроцессора 267, 270, 274, 281
 Сигнатура 178, 184, 186, 229, 235
 Символ печатаемый 114
 Система микропроцессорная 39
 — проектирования микропроцессорных систем (СПМС) 163
 Скорость передачи 125, 154
 Соотношения временные 47
 Сопряжение уровней 111
 Состояние 46, 268
 Способность нагрузочная 22
 Сравнение 70
 Стабилизация напряжения питания 313
 Стимулирующая программа 178, 188
 Стек 65
 Схемотехника 18
 Счетчик
 — байтов 69
 — команд 63
 — реверсивный 62

Таймер 77, 141
 Температура 124, 320
 Тест 201, 228
 Тестер 189, 222
 Тестирование

— внутрисхемное 302
 — импульсное 17
 Тестовая программа 131, 226, 236
 Токовая петля 113, 251
 Точка контрольная 8
 Транзистор многоэмиттерный 22
 Трассировка 142, 152, 171
 Триггер 244

Указатель стека 64
 Укладка многослойная 55
 Универсальный асинхронный приемопередатчик (УАПП) 97, 161, 254
 Уровней преобразование 250
 Установка для обслуживания 206
 Устройства периферийные 39
 Утечка тока 29

Формат команды 67

Холла эффект 13

Целевая система 163, 175
 Цикл 64
 Циклический избыточный код 135, 209, 235

Частота тактовая 8
Чувствительность 13

«Шахматная доска» 131, 196, 229
 Шестнадцатеричная запись 43
 Шина
 — адресная 41
 — данных 40
 — мультиплексируемая 50
 — системная 35, 40
 — управляющих сигналов 44
 Шинная организация 18

Эмиттерные связи 30
 Эмуляционная приставка 190
 Эмуляция внутрисхемная 17, 163, 175, 222, 224

Ячейка 41, 51

Оглавление

Предисловие переводчика	5
Предисловие	6
Часть I. Микропроцессорные системы	8
Глава 1. Цифровые устройства и системы	8
Токовый пробник	12
Логический пульсатор	15
Логические клипсы	17
Схемотехника логических ИС	18
ТТЛ	19
Импульсные характеристики: переходные процессы	27
Медленные переходные процессы	28
ТТЛ-элемент с тремя состояниями	28
Отыскание неисправностей	29
Логика с эмиттерными связями (ЭСЛ)	30
Интегральная инжекционная логика (И ² Л)	31
nМОП/rМОП/КМОП-структуры	32
Быстродействующие КМОП-ИС	35
Заключение	38
Глава 2. Микропроцессорные системы	39
Компоненты системы	39
Системные шины	40
Основные временные соотношения	47
Запоминающие устройства и распределение памяти	50
Микропроцессор	61
Схемы ввода-вывода	74
Заключение	77
Глава 3. Интерфейс	78
Параллельный интерфейс	78
Интерфейс Centronics	81
Интерфейсная шина IEEE 488	84
Представление печатаемых символов	114
Заключение	115
Часть II. Методы обслуживания	116
Глава 4. Методы поиска неисправностей	116
Простейшие виды контроля системы	117
Тестирование с входными воздействиями	118
Свободный прогон микропроцессора	119

Источники питания	121
Влияние температуры	124
Помехи	125
Неисправности интерфейса	125
Периферийные устройства	126
Виды ошибок	127
Учет неисправностей	128
Перечни проверок и алгоритмы их выполнения	128
Заключение	129
Глава 5. Принципы самоконтроля	131
Тестирование ЗУПВ	131
Тестирование ПЗУ	135
Устройства ввода-вывода	140
Заключение	141
Глава 6. Логический анализ	142
Основные функции логических анализаторов	145
Тестирование матричной клавиатуры	150
Заключение	152
Глава 7. Поиск неисправностей в последовательных интерфейсах	154
Общие методы	154
Специальные методы контроля последовательного интерфейса	156
Заключение	161
Глава 8. Внутрисхемная эмуляция	162
ВСЭ как инструмент проектирования	163
Эмуляция с использованием СПМС	164
Трассировка	171
Внутрисхемная эмуляция как средство поиска неисправностей	174
Заключение	176
Глава 9. Сигнатурный анализ	177
Сигнатурный анализатор	179
Сигнатурное окно	181
Сигнатуры «земли» и напряжения питания	184
Свободный прогон как средство задания стимулирующих воздействий	185
Пример: плата контроллера на базе микропроцессора Z80	186
Программы стимулирующих воздействий	188
Тестеры и эмуляторы	189
Micropolis	191
Программа генерации тестовых воздействий для одноплатной ЭВМ	196
Заключение	198
Глава 10. Контроль с применением ЭВМ	200
Автоматическая контрольная аппаратура	201
Автоматизированный контроль	205
Сигнатурный анализ под управлением ЭВМ	216
Заключение	221

Оглавление	335
Часть III. Обслуживание с небольшими затратами	222
Глава 11. Тестеры на базе микроЭВМ	222
Аппаратные средства внутрисхемного эмулятора	224
Тестовые программные средства	226
Сигнатуры ПЗУ	229
Пример реализации тестовой программы	236
Заключение	238
Глава 12. Одноплатные анализаторы	240
Сигнатурный анализатор	240
Логический анализатор	246
Заключение	248
Глава 13. Средства обмена информацией	249
Интерфейсный преобразователь	249
Преобразование сигналов интерфейсов RS232 и RS423	249
Преобразование сигналов токовой петли	251
Приставка к логическому анализатору для обработки последовательных данных	253
Генератор последовательных символов	256
Заключение	266
Приложения	267
Приложение 1. Сравнительные характеристики некоторых микропроцессоров	267
Сведения о микропроцессорах	267
Микропроцессор 6502	270
Сигналы 6502	270
Команды микропроцессора 6502	273
Микропроцессор 8085	274
Сигналы 8085	274
Команды микропроцессоров 8080/8085	279
Микропроцессор Z80	281
Сигналы микропроцессора Z80	281
Команды микропроцессоров Z80 и NSC800	285
Микропроцессоры 8088 и 8086	289
Сигналы микропроцессоров 8088 и 8086	290
Программная модель микропроцессоров 8088/86	295
Приложение 2. Технические характеристики трех одноплатных микроЭВМ	297
Описание микроЭВМ AIM 65	297
Одноплатный набор проектировщика SDK-85	302
Одноплатный контроллер MEDC Z80	307
Приложение 3. Источники и стабилизаторы напряжения питания	313
Источники питания	313
Коммутируемые регуляторы	323
Приложение 4. Варианты схем построения сигнатурных и логических анализаторов	330
Предметный указатель	331