

ИНСТРУМЕНТАЛЬНЫЕ И УЧЕБНЫЕ СРЕДСТВА

6.1. Средства для разработки систем на микроконтроллерах и ПЛИС

Комплекс инструментальных средств, необходимый для разработки систем на микроконтроллерах и ПЛИС, включает:

- средства разработки прикладного программного обеспечения микроконтроллеров;
- средства ввода принципиальной схемы контроллера и проектирования топологии печатной платы;
- средства разработки и верификации проекта на ПЛИС;
- средства автономной отладки аппаратуры;
- средства комплексной отладки аппаратуры и программного обеспечения в реальном масштабе времени;
- средства программирования БИС памяти программ и программируемой логики.

Общепринятым подходом в настоящее время является использование в качестве ведущего процессора (host-processor) инструментальных средств персонального компьютера (рис. 6.1). Это позволяет разработчику инструментария переложить на компьютер общесистемные задачи и сосредоточиться в каждом случае на реализации специфических функций проектирования и отладки. Одновременно персональный компьютер служит в качестве объединяющего ядра, в том числе конструктивного, и средством коммуникации в вычислительных сетях. Пользователь при этом получает возможность, помимо средств проектирования, пользоваться широким набором прикладного программного обеспечения для MS DOS и Windows.

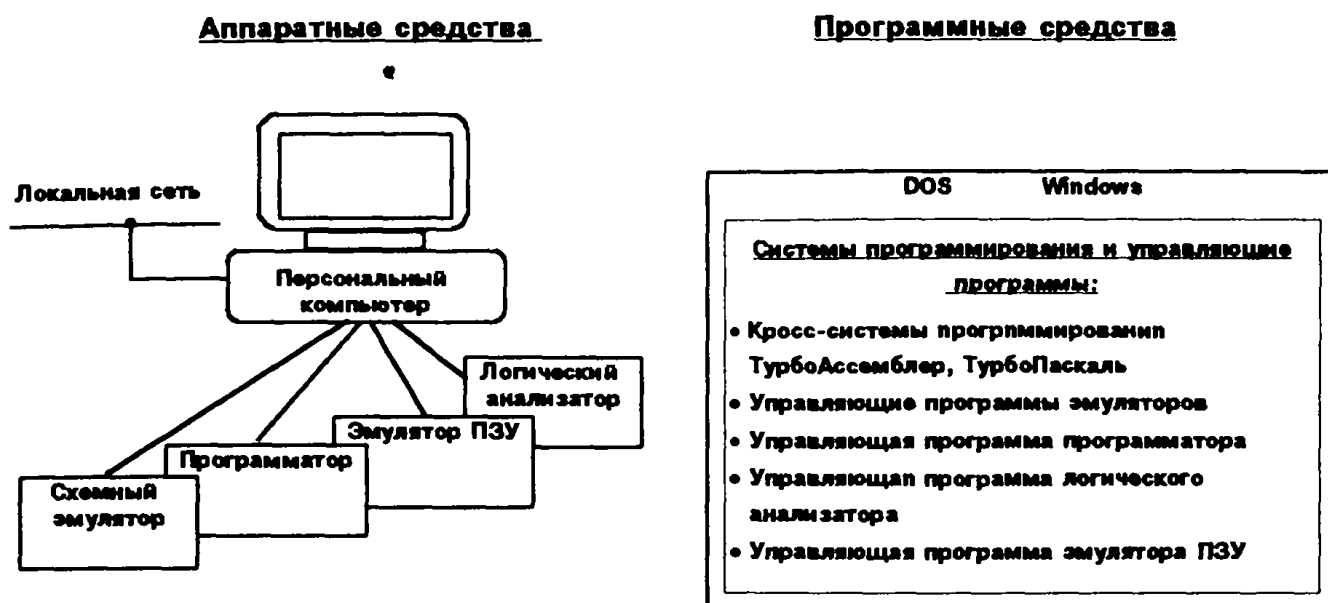


Рис. 6.1. Структура комплекса инструментальных средств

Перечисленные выше группы средств проектирования и отладки включают приборы и системы различной категории сложности и стоимости. Далее мы рассмотрим профессиональные средства, обеспечивающие максимальную поддержку разработчику и позволяющие создать изделие с микропроцессорной системой управления в запланированные сроки и в пределах утвержденной сметы затрат.

В качестве средств ввода принципиальных схем и разводки топологии печатных плат на персональных компьютерах используются системы проектирования типа PCAD (ACCEL EDA) и OrCad. В последнее время вышло несколько хороших книг, описывающих эти средства и работу с ними [5, 6], поэтому мы этого вопроса касаться не будем.

В предыдущей главе рассмотрена система MAX+plusII. Она является хорошим примером средств разработки и верификации проектов на ПЛИС.

В качестве средств автономной отладки аппаратуры систем на микроконтроллерах и ПЛИС используются осциллографы, логические анализаторы, логические пробники и т.д. В разделе 6.3.2 этой главы будет описан логический анализатор/эмулятор ПЗУ, реализованный на ПЛИС типа FPGA.

В основном в этой главе рассмотрены средства разработки программного обеспечения, средства комплексной отладки программ и аппаратуры, средства программирования энергонезависимой памяти и ПЛИС, учебные классы микропроцессорной техники и ПЛИС.

6.2. Средства разработки программ

В настоящее время проектировщику доступны средства разработки программного обеспечения, поддерживающие все этапы процесса. Прежде всего следует отметить, что средства программирования микропроцессорных сис-

тем делятся на *резидентные* и *кросс-средства*. Резидентные средства формируют исполняемый код, который может быть выполнен инструментальном компьютере. Это возможно в тех случаях, когда контроллер выполнен на микропроцессоре, причем той же серии, что и микропроцессор инструментального компьютера. К этому типу относятся все средства программирования систем промышленной автоматики на основе процессоров i386/i486/Pentium, например производимых фирмами Advantech и Octagon Systems.

Кросс-средства могут генерировать и отлаживать код программы для любого типа микроконтроллера и микропроцессора. Это подход более общий, но для эффективной работы кросс-системы необходимо, чтобы производительность инструментального компьютера существенно превосходила производительность целевого процессора (на плате контроллера).

Средства разработки проекта на языках Си и Ассемблер, справочную информацию по архитектуре ряда микропроцессорных семейств фирмы Intel обеспечивает пакет *ApBUILDER*. Он функционирует в среде Microsoft Windows, результатом его работы является исходный текст программы, который далее должен транслироваться с использованием традиционных средств. Собственно компиляторы в состав этого пакета не входят. Пакет построен по модульному принципу и может быть настроен на архитектуру семейств Intel 186 и 386, MCS-51, MCS-96. На рис. 6.2 приведено изображение головного экрана программы *ApBUILDER*, настроенной на архитектуру MCS-96.

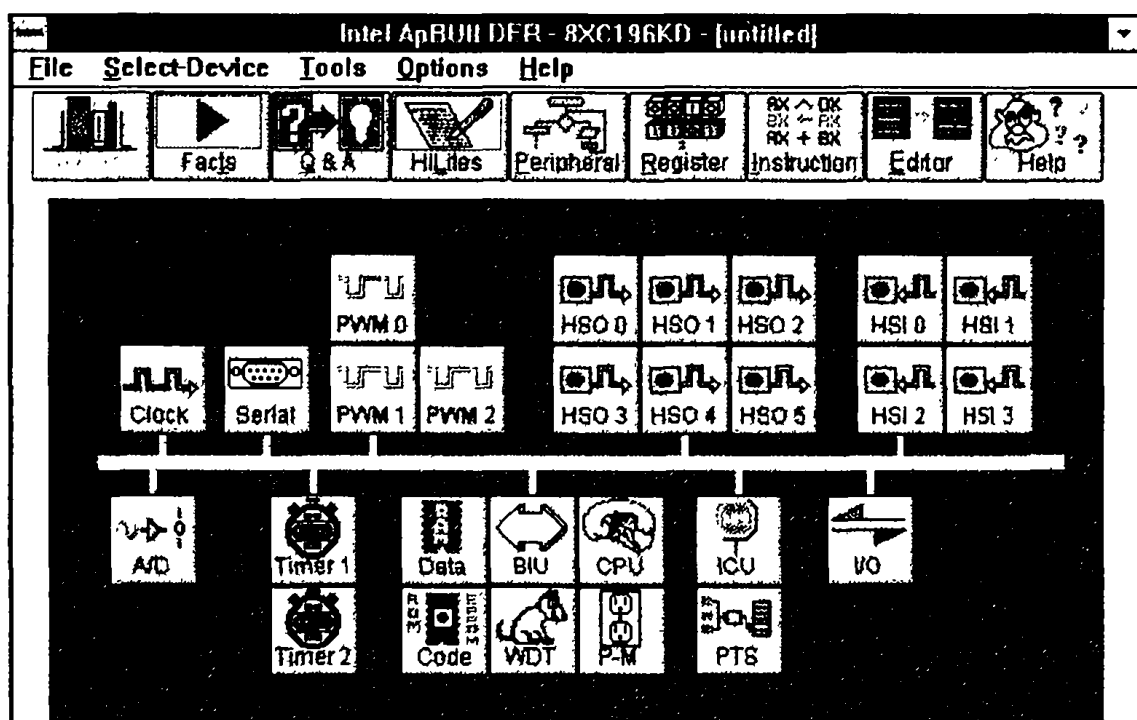


Рис. 6.2. Вид головного экрана программы *ApBUILDER*

Этот экран содержит три меню: команд сервиса и конфигурации оболочки (верхняя строка), команд вызова информационно-справочных проце-

дур (пиктографическое меню), команд выбора разделов (в виде модульной структуры микропроцессора или микроконтроллера). Команду любого меню можно активизировать с помощью курсора, управляемого мышью.

Команды верхней строки позволяют выбрать рабочий файл (File) и микропроцессорное семейство (Select-Device), запустить информационно-справочную процедуру (Tools), сконфигурировать оболочку (Options), получить справочную информацию по системе (Help).

Команды пиктографического меню дублируют опции сервисной команды Tools и позволяют получить информацию определенного типа о выбранном блоке микроконтроллера и создать фрагменты рабочей программы. Наиболее интересны команды HiLites, Peripheral, Register и Instructions. Первая из них выдает краткую характеристику блока, вторая вызывает редактор (процедуру) конфигурации блоков ввода/вывода микроконтроллера. В интерактивном редакторе конфигурации можно определить режим работы блока, после чего будет автоматически сформирован фрагмент исходного текста на выбранном языке для инициализации этого блока. Команда Register вызывает редактор (процедуру) конфигурации регистров управления выбранного блока. В диалоговом экране отображаются все регистры с описанием их функций, имен и адресов, отдельных битов. Этот редактор работает вместе с предыдущим и позволяет более детально изучить режимы работы блока. По команде Instructions открывается окно со справочной информацией о командах, связанных с выбранным блоком. Указывается формат каждой команды, время выполнения при определенной тактовой частоте. Это дает возможность уточнить синтаксис и длительность командного цикла.

Программы типа *ArBUILDER* не следует переоценивать, средства этого уровня играют роль демонстрационную и учебную, помогают быстро ознакомиться с возможностями определенного семейства микропроцессоров или микроконтроллеров.

Пакеты инструментальных программ для микропроцессоров и микроконтроллеров представляют собой комплексы кросс-программ, обеспечивающие трансляцию исходных текстов рабочих программ с языков Паскаль, Си, Ассемблер в перемещаемый объектный код, последующую компоновку отдельных модулей в единую программу и размещение ее по абсолютным адресам памяти программ.

Последовательность этапов разработки программного обеспечения контроллера, содержание этапов и используемые при этом инструментальные средства следующие:

- создание исходного текста программы. Исходные модули программы могут быть написаны как на языке высокого уровня, так и на ассемблере. Используется редактор текста, который в состав традиционного инструментального пакета обычно не входит;

- трансляция. Транслятор и Ассемблер преобразуют исходный текст в перемещаемый объектный код. Кроме файлов объектного кода каждый из них формирует файл листинга трансляции. Используются кросс-транслятор и кросс-ассемблер из пакета инструментальных средств;
- создание библиотечных модулей. Библиотекарь позволяет помещать перемещаемый модуль в библиотеку, извлекать модуль из библиотеки, создавать новую библиотеку и т.д. Используется программа-библиотекарь из пакета инструментальных средств;
- компоновка перемещаемых объектных модулей (в том числе библиотечных) в единую программу и размещение ее по абсолютным адресам памяти. Используется программа-компоновщик пакета инструментальных средств;
- преобразование абсолютного кода в файл типа .hex. Используется утилита-конвертор из пакета инструментальных средств.

Следует отметить, что языки программирования микропроцессорных контроллеров служат для создания программ, функционирующих, как правило, без операционных систем. По этой причине отсутствуют возможности работы, например, с файлами. Ограниченные вычислительные возможности микроконтроллеров приводят к исключению рекурсии, ограничивают комбинации сложных типов данных (типа массив записей). С другой стороны, каждый язык, например Паскаль-51, ориентирован на архитектуру определенного семейства, включает машинно-зависимую компоненту. В нем можно использовать имена регистров и битов микроконтроллера без предварительного описания. Для работы в реальном масштабе времени введены процедуры прерываний. В настоящее время версии языков программирования, конкретные компиляторы и связанные с ними программы из состава пакетов программирования разрабатываются независимыми фирмами-поставщиками инструментальных средств. При создании исходного текста прикладной программы следует пользоваться руководствами типа «Описание языка» из состава документации пакета.

Инструментальные программы традиционных пакетов запускаются на выполнение командными строками операционной системы. Очевидным недостатком такого подхода является недостаточный уровень интерфейса пользователя. Отладка программы по листингам трансляции занимает много времени. Однако традиционные пакеты инструментальных программ имеют ценность и в настоящее время, поскольку в них решены задачи, связанные с генерацией эффективного кода, что для микропроцессорных контроллеров имеет первостепенное значение. Новый этап жизни пакетов программирования микропроцессоров и микроконтроллеров начался в связи со все большим распространением операционной системы Linux.

Традиционные пакеты в настоящее время являются основой систем программирования, функционирующих в среде MS Windows и имеющих

улучшенный интерфейс пользователя за счет оболочек, позволяющих работать в интерактивном режиме.

Системы программирования с интерфейсом в виде оболочки. Наиболее простым способом облегчить работу с традиционными пакетами программирования является создание оболочки, имеющей улучшенный интерфейс пользователя. Оболочки обычно имеют меню, позволяющее выбрать один из возможных режимов работы, оконный интерфейс, механизм оперативного режима Help. В качестве собственно инструментальных программ используются традиционные пакеты. Внутренние вызовы программных модулей, как инструментальных, так и пользовательских, осуществляются с использованием обычных командных файлов.

Основные ограничения программных систем-оболочек заключаются в неполном согласовании протоколов обмена между отдельными ранее созданными инструментальными программами, например, редактором текста и кросс-ассемблером. Это приводит к ограничениям в процессе отладки. Например, после трансляции и сообщении об ошибках приходится просмотреть листинг трансляции, а затем в редакторе править исходный текст. Выход заключается в создании собственного полностью согласованного варианта инструментальной программы. После создания собственного редактора текста программа-оболочка получает возможность передать ему результаты трансляции, а он помечает строку с ошибочным синтаксисом. Ограничения систем-оболочек снимаются при наличии полных данных о протоколах обмена в инструментальных пакетах и формате отладочной информации.

Интегрированные системы программирования. Наиболее эффективным средством разработки ПО для микроконтроллеров являются интегрированные системы программирования. Прототипом таких систем является Турбо-Паскаль фирмы Borland, но в отличие от этой системы резидентного типа для микроконтроллеров в основном используются кросс-системы.

Отличием интегрированных систем от оболочек является полное согласование между компонентами по протоколам обмена и форматам отладочной информации, что позволяет организовать сквозной цикл проектирования и отладки, начиная с программирования на языке высокого уровня, через отладку на программной модели и до управления ресурсами схемного эмулятора при отладке рабочей программы совместно с аппаратурой. Платой за эти преимущества является очень большая сложность и трудоемкость работ по созданию и сопровождению такой системы.

Рассмотрим возможности интегрированных систем на примере кросс-системы Паскаль-51 для программирования микроконтроллеров семейства Intel 8051/52.

Интегрированная система Паскаль-51 включают редактор текста, кросс-транслятор с кросс-ассемблером, редактор связей, библиотеку периода выполнения, символический отладчик.

Основой этой системы является язык программирования Паскаль-51. Он отличается от авторской версии Н. Вирта следующими особенностями:

- наличием машинно-зависимой компоненты, т.е. возможности обращаться к программно-доступным ресурсам микроконтроллера без их предварительного описания;
- наличием типа `bit`. Битовые переменные являются глобальными и накладываются на пространство битовой памяти, имеются встроенные процедуры работы с битовыми переменными, введены логические операции над битами;
- язык Паскаль-51 является языком программирования реального времени. Это означает, что библиотека исполняющей системы и код, формируемый компилятором являются полностью прерываемыми. Возможно создание специальной процедуры обработки прерываний `interrupt` и вызов ее по имени;
- язык Паскаль-51 допускает ассемблерные вставки.

Язык, компилятор и редактор связей поддерживают модульную технологию программирования. Имеется понятие «проект», под которым понимается список файлов, транслируемых отдельно, но затем собираемых в один исполняемый модуль редактором связей. В проект могут входить файлы с исходным текстом на языке Паскаль-51, на языке ассемблера, объектные файлы. Первый файл проекта рассматривается как главный (Main file), редактор связей создает исполняемый модуль с расширением `.hex` от имени главного файла.

После ввода в систему модулей, составляющих рабочую программу, их связывание, подключение библиотек и распределение ресурсов выполняется в автоматическом режиме. Это предохраняет данные от непреднамеренного затирания.

Символический отладчик, входящий в интегрированную систему Паскаль-51, позволяет выполнять отладку на уровне операторов языка высокого уровня, отслеживая изменения логических переменных рабочей программы в окне Watch. Одновременно в окне Resource можно видеть состояние аппаратных ресурсов микроконтроллера (ячеек памяти и регистров).

Один из возможных видов рабочего экрана интегрированной системы «Паскаль-51» представлен на рис. 6.3.

Видны основное меню и открытые окна: редактора с исходным текстом рабочей программы; дизассемблера с последовательностью команд в символической форме; ресурсов микроконтроллера с набором регистров и ячейками внутренней памяти.

При выполнении в процессе отладки последовательности команд рабочей программы отладчик (с использованием программно-логической модели) ведет подсчет машинных тактов команд и отображает длительность

выполнения фрагмента программы при выбранной частоте синхронизации микроконтроллера.

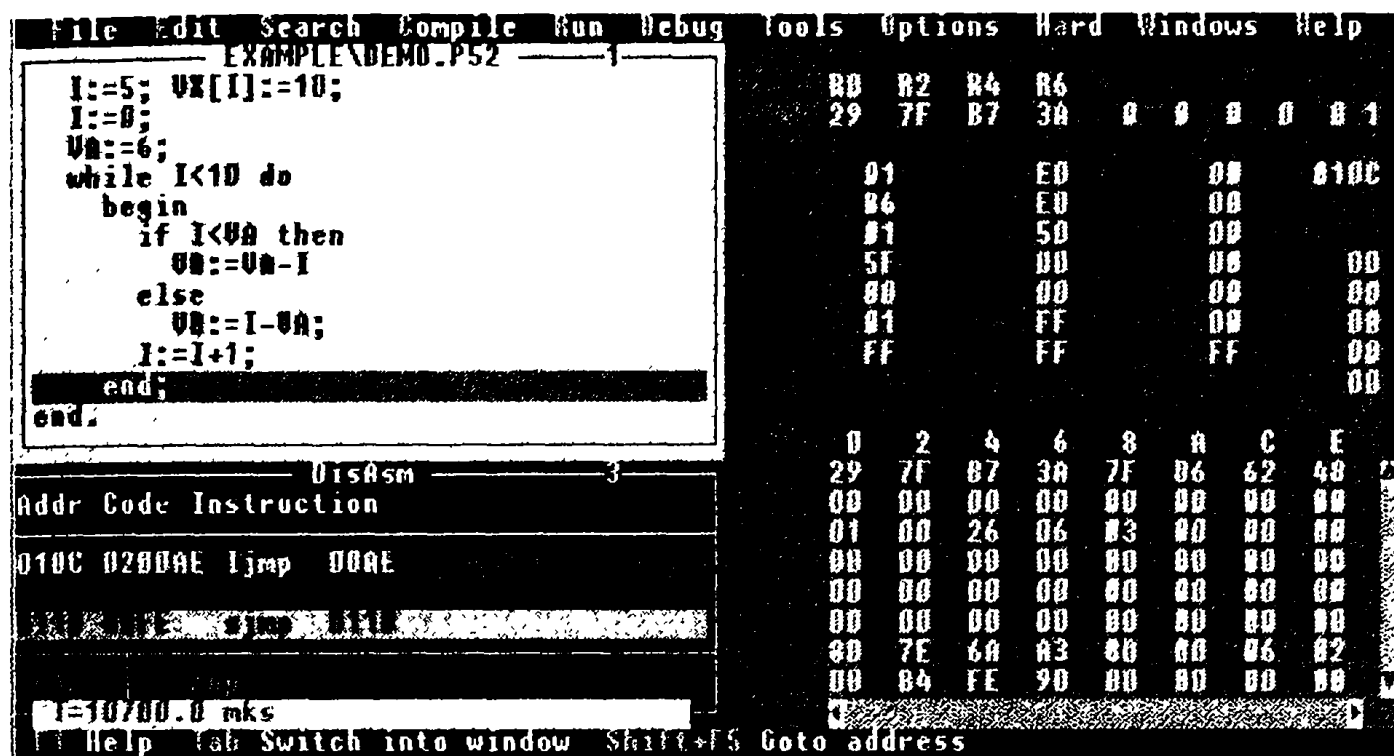


Рис. 6.3. Вид экрана интегрированной системы программирования «Паскаль-51»

Команды основного меню в основном соответствуют набору, который использовала фирма Borland в своих системах программирования. Главное отличие связано с командой **Hard**, которая позволяет настроить систему на конкретный тип микроконтроллера типа MCS-51 (Intel, Philips, Dallas и др.), а также аппаратный эмулирующий модуль (одноплатный контроллер, схемный эмулятор). Система позволяет создавать рабочие программы для нескольких десятков типов микроконтроллеров семейства MCS-51.

6.3. Средства отладки в реальном масштабе времени

Основой методов отладки микропроцессорных систем в реальном времени является контролируемое управление магистралью и наблюдение на магистрали ответной реакции отдельных модулей системы. Понятие «контролируемое управление» подразумевает здесь задание необходимой (в соответствии с алгоритмом тестирования) последовательности функциональных состояний на магистрали. Задатчиком является микропроцессорная БИС или замещающий ее имитирующий процессор. Набор функциональных состояний определяется всеми возможными комбинациями логических сигналов на выводах микропроцессорной БИС при выполнении набора команд. Основным функциональным тестом (последовательностью состояний)

при отладке является рабочая программа. При наблюдении за реакцией системы кроме фиксации отклика модулей по магистрали нужно фиксировать и состояние самой микропроцессорной БИС (или имитирующего процессора).

В настоящее время интеграцию аппаратуры и программного обеспечения при отладке микропроцессорной системы в реальном масштабе времени позволяют выполнить три типа средств: отладочные платы, схемные эмуляторы и эмуляторы ПЗУ (памяти программ). Отладочная плата представляет собой типовую комбинацию аппаратных средств. Такая платформа позволяет сосредоточиться только на разработке программного обеспечения. После отладки плата может быть использована для управления как автономный контроллер. Схемные эмуляторы и эмуляторы ПЗУ являются внешними средствами отладки и отличаются местом подключения их к магистрали целевой системы и способами управления/наблюдения.

Эмуляция как метод подразумевает замещение некоторого модуля системы (процессора или памяти программ) функциональным аналогом (эмулятором), который позволяет сделать процесс управления контролируемым и наблюдаемым. Особенностью эмуляции микропроцессорных систем является то, что функции управления и наблюдения также выполняются эмулятором, а не отдельными блоками отладочного стенда. Это связано с полной интеграцией аппаратного и программного обеспечения и сложностью логико-временных диаграмм функционирования этих систем. Отдельные приборы наблюдения могут подключаться как дополнительные. Таким образом можно заключить, что основными функциями эмуляции как метода отладки микропроцессорных систем является замещение микропроцессорной БИС или модуля памяти программ, управление формированием последовательности функциональных состояний на магистрали и наблюдение за откликами модулей системы, а также состоянием центрального процессора. Требования режима реального времени приводят к необходимости производить замещение не только функциональное, но и с точки зрения важнейших электрофизических параметров.

Схемный эмулятор микропроцессора или микроконтроллера строится на основе имитирующего процессора, замещающего целевую микропроцессорную БИС. Кроме ресурсов ЦПУ этой БИС традиционно эмулируется память программ и данных. Эмулятор подключается к магистрали отлаживаемой системы эмуляционной вилкой через розетку вместо целевой микропроцессорной БИС (сечение 1 на рис. 6.4). Непосредственное управление магистралью осуществляет имитирующий процессор, рабочая программа и функциональные тесты вначале размещаются в памяти программ эмулятора. Поскольку изменять набор операций и диаграммы обмена эмулятора трудно, этот метод применяется только для отладки систем на основе однокристалльных микропроцессоров и микроконтроллеров с фиксированной

системой команд. Каждому микропроцессорному семейству соответствует свой схемный эмулятор.

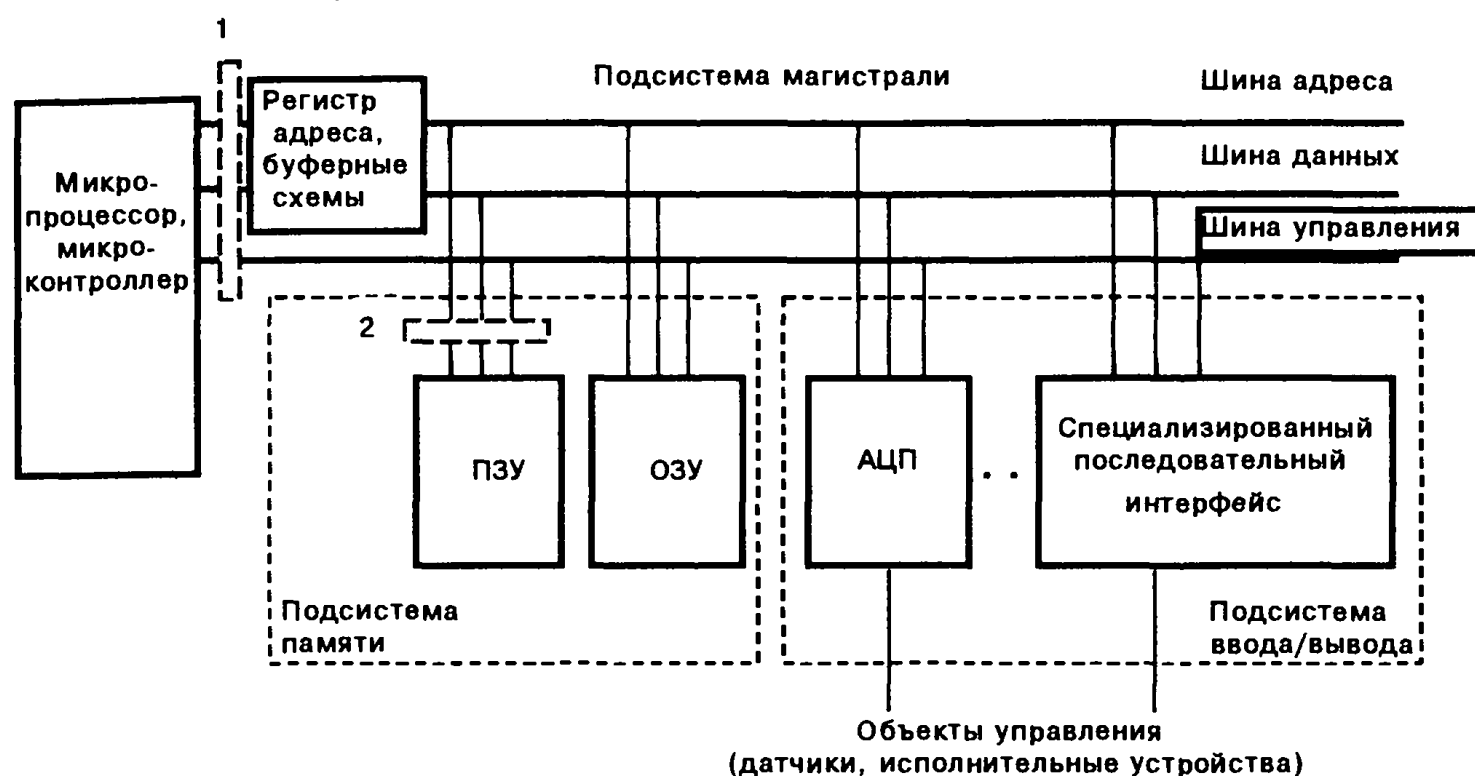


Рис. 6.4. Возможные места подключения для управления магистралью при отладке

Эмулятор ПЗУ строится с использованием матрицы памяти, замещающей память программ. Он подключается к магистрали отлаживаемой системы эмуляционной вилкой через розетку вместо микросхемы памяти (сечение 2 на рис. 6.4). Непосредственное управление магистралью осуществляет микропроцессорная БИС, рабочая программа и функциональные тесты размещаются в памяти программ эмулятора ПЗУ.

6.3.1. Одноплатные контроллеры (EC196, EV32, EC812)

Это аппаратные платформы типовой конфигурации, ориентированные на решение собственных задач, но имеющие ресурсы для целей отладки. Наиболее простыми и дешевыми являются платы типа evaluation board и target board. В комплекте с ними обычно поставляются демонстрационные версии систем программирования с ограничением, например, размера адресуемой памяти пределом 2 – 4 Кбайт. Эти средства служат в основном для целей обучения и ознакомления пользователей с возможностями определенного семейства микропроцессорных БИС. На другом конце спектра лежат платы промышленных контроллеров на основе 32-разрядных микропроцессоров, предназначенные для работы под управлением операционных систем реального времени. Такие платы управления входят в полный комплект аппаратных средств промышленной автоматики. Из комплекта при необхо-

димости могут быть выбраны каркасы, объединительные платы, клавиатуры и мыши в защищенном исполнении, платы процессоров DSP и ввода/вывода. Эти средства очень дороги, их применение целесообразно на промышленных объектах, когда стоимость всего технологического оборудования значительно превышает стоимость электронной системы управления. К промежуточному уровню можно отнести одноплатные контроллеры на основе 8- и 16-разрядных микроконтроллеров. Такие контроллеры имеют типовую конфигурацию процессорного ядра и интерфейсов ввода/вывода с минимальными дополнениями в виде схем, поддерживающих отладку. Предварительно отлаженная аппаратная компонента сокращает трудозатраты пользователя, от которого требуется только разработать и отладить управляющую программу. Отладка производится с помощью инструментального компьютера, к которому одноплатный контроллер обычно подсоединяется через последовательный интерфейс. После завершения отладки одноплатный контроллер может использоваться в автономном режиме в качестве встроенного модуля управления прибором или установкой. Преимущества типовых одноплатных контроллеров перед промышленными проявляются при необходимости иметь небольшую партию изделий, стоимость и энергопотребление которых существенно ограничены. При массовом производстве на смену типовым приходят заказные контроллеры.

В качестве примера средств этого класса рассмотрим систему проектирования для семейства MCS-96, которая включает одноплатный контроллер ЕС196Кх и интегрированный пакет программирования/отладки на языках Ассемблер или Паскаль-96. В режиме отладки система работает с инструментальным персональным компьютером.

Контроллер представляет собой отлаженную аппаратную платформу, разработчику остается только создать программное обеспечение (ПО) и загрузить исполняемый код в память программ. После завершения процесса отладки контроллер способен самостоятельно управлять приборами и оборудованием. Его важной особенностью является тестопригодность – служебный разъем, через который контроллер в режиме отладки работает с компьютером, позволяет проверить выполнение рабочей или специальной программы. Контроллер может выступать в роли тестирующего блока для всего изделия. Для этого нужно вернуться к режиму отладки и загрузить в контроллер программу функционального тестирования узлов системы управления и оборудования.

Плата контроллера изображена на рис. 6.5. Она выполнена в виде набора функционально-топологических модулей, набор модулей следующий:

- процессорный модуль, включающий БИС микроконтроллера со схемой рестарта и схемой обслуживания АЦП;
- рабочая память;
- модуль поддержки отладочного режима;
- модуль общего управления;

- последовательный порт пользователя со схемой согласования уровней;
- модуль индикации.

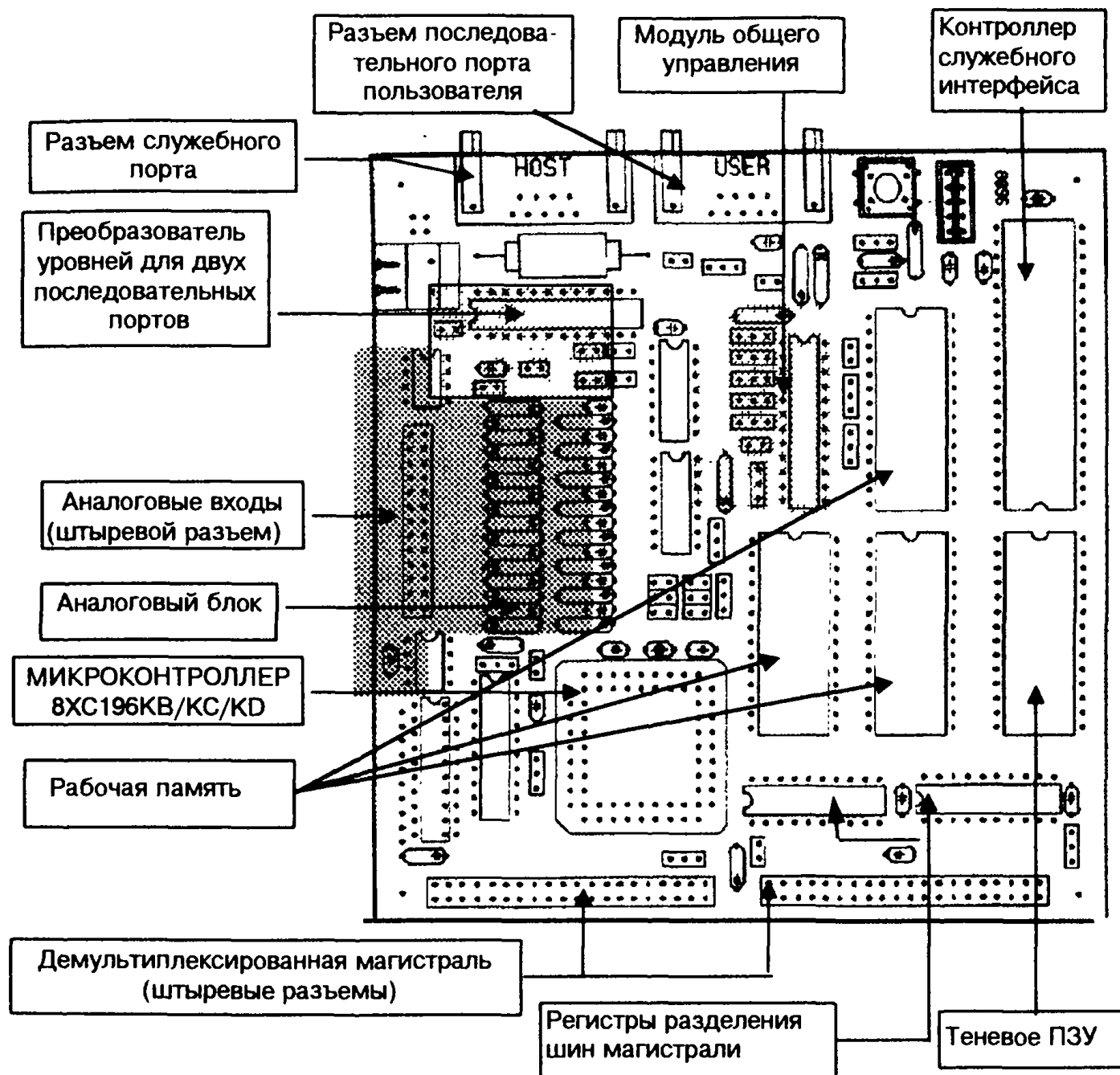


Рис. 6.5. Одноплатный контроллер ЕС196Кх

Процессорный модуль содержит микроконтроллер типа 8XC196KB/KC или KD (установленный в розетке), схему рестарта и схему обслуживания встроенного в микроконтроллер аналого-цифрового преобразователя. Последняя включает элементы защиты и фильтрации для всех аналоговых входов и источник опорного напряжения.

Рабочая память контроллера объемом до 64 Кбайт может настраиваться перемычками в 16-разрядную конфигурацию (до 32 Кслов) или 8-разрядную конфигурацию (32 Кбайта плюс 16-разрядные слова до 6 Кслов). В режиме отладки рабочая память реализуется в виде микросхем

ОЗУ на розетках. После завершения отладки память программ заменяется на запрограммированные БИС ПЗУ или ППЗУ.

Модуль поддержки отладочного режима включает теневое ПЗУ и служебный последовательный порт со схемой согласования уровней (ТТЛ – RS-232). Обслуживание последовательного порта осуществляется с использованием микросхемы 8250. С этим модулем связан программный модуль, входящий в управляющую программу инструментального компьютера ПЗУ.

Модуль общего управления формирует сигналы управления, выполнен в виде микросхемы ПЛИС типа PAL22V10.

Последовательный порт пользователя включает разъем и схему согласования уровней. Этот модуль дает возможность разработчику использовать внутренний последовательный порт микроконтроллера. Модуль индикации позволяет, при необходимости, визуально следить за выполнением рабочей или тестирующей программы.

В целях наращивания системы управления с платы контроллера через разъем выведены демультимплексированные шины адреса и данных (разделение производится на плате с использованием регистров-защелок) и все линии специализированного ввода/вывода микроконтроллера (широкоимпульсного модулятора, линии скоростного ввода/вывода).

Разработчик может свободно использовать все ресурсы микроконтроллера за исключением прерываний NMI и TRAP, которые задействованы для целей отладки.

Поскольку контроллер ориентирован на практическое использование, его топология тщательно проработана с точки зрения уменьшения габаритов и удобства интеграции в приборы и системы. Четырехслойная печатная плата размером 130x130 мм обеспечивает устойчивую работу на частотах до 20 МГц. Общий потребляемый ток составляет 250–300 мА при частоте $F_{osc} = 12$ МГц и зависит от того, работает блок индикации или нет. Поскольку рабочая компонента контроллера выполнена с использованием kMOS микросхем, потребляемая мощность в режиме целевого контроллера может быть дополнительно снижена за счет отсутствия в уже отлаженных изделиях служебной компоненты.

В процессе отладки вся память реализуется в виде микросхем ОЗУ на розетках. После завершения отладки память программ заменяется на запрограммированные отлаженным кодом БИС ПЗУ или ППЗУ.

В настоящее время для работы с одноплатным контроллером доступны интегрированные системы программирования на языках Ассемблер и Паскаль-96. Каждая система включает редактор текста, компилятор с языка высокого уровня или полноразмерный кросс-ассемблер, библиотеки периода выполнения, редактор связей, символический отладчик. Имеется контекстный Help. Вид экрана и главного меню похожи на интерфейс пользователя рассмотренной выше системы Паскаль-51. Отличие заключается в наборе аппаратных платформ, которыми эта система программирования способна

управлять. Кроме одноплатного контроллера ЕС196Кх, под управлением этих систем может работать плата evaluation board EV80C196 фирмы Intel. Пользователь имеет возможность не выходя из системы загружать свои программы в память контроллера, выполнять их по шагам, в прогоне, с остановом по контрольным точкам. Поддерживается модульная технология программирования с раздельной трансляцией модулей. Отладка программ производится в исходном тексте, причем на каждом шаге отладки можно наблюдать за изменениями внутренних ресурсов как на логическом, так и на физическом уровне, а также модифицировать их. Подключаемые библиотеки арифметики с плавающей запятой и библиотеки тригонометрических функций плюс типизация данных на уровне окна Watch позволяют успешно работать с данными типа Integer и Real. Моделирование выполняется с использованием аппаратных средств одноплатного контроллера.

Основным ограничением процесса отладки на средствах целевого контроллера является, как правило, изъятие прерывания с наиболее высоким приоритетом (NMI) из распоряжения пользователя. Устранить это ограничение можно при использовании более развитых средств управления, которые применяются в схемных эмуляторах.

Одноплатный контроллер EV32 с присоединяемым супервизором. В качестве кардинального решения задачи создания дешевого одноплатного контроллера, способного работать как часть системы проектирования-отладки, сотрудниками лаборатории «Микропроцессорные системы» МИФИ предложено выделить из его структуры и реализовать в виде отдельной платы супервизор, обеспечивающий выполнение отладочных функций. При этом в структуру контроллера должен быть введен специальный отладочный разъем, через который контроллер подключается к плате супервизора и всему комплексу инструментальных средств при разработке программного обеспечения.

Реализовать это решение позволил опыт использования БИС программируемой логики. С помощью этой элементной базы удалось разделить целевые и отладочные функции и предоставить пользователю готовый целевой контроллер, к которому внешние средства отладки подключаются через специальный разъем. Такой подход делает возможным в сжатые сроки разработать недорогое изделие, поскольку создается и отлаживается только программное обеспечение системы управления, а серийная аппаратная платформа не имеет избыточности.

Комплекс для разработки систем управления состоит из платы контроллера EV32_system, платы супервизора EV32_supervisor и интегрированной системы программирования «ТурбоАссемблер» или «ТурбоПаскаль», работающей в среде MS DOS инструментального компьютера. Комплекс предназначен для разработки и отладки систем управления на базе микроконтроллеров 8031/32/51FX. Плата контроллера EV32_system может быть использована при серийном производстве целевых систем.

Одноплатный контроллер (рис. 6.6) включает БИС микроконтроллера, память программ и данных (на розетках), блок синхронизации и контроля питания, дешифратор адреса внешних устройств, два разъема параллельного интерфейса для подключения внешних устройств, блок последовательного порта (микроконтроллера) с оптронной развязкой, отладочный разъем.

Контроллер допускает установку до 48 Кбайт памяти программ и до 32 Кбайт памяти данных. В случае использования в качестве памяти программ ОЗУ (при отладке или для возможности перезагрузки) ее максимальный объем составит 32 Кбайт. Имеется возможность сохранения перезагружаемой программы и данных в памяти при выключении питания.

Для контроля питания используется микросхема МАХ707.

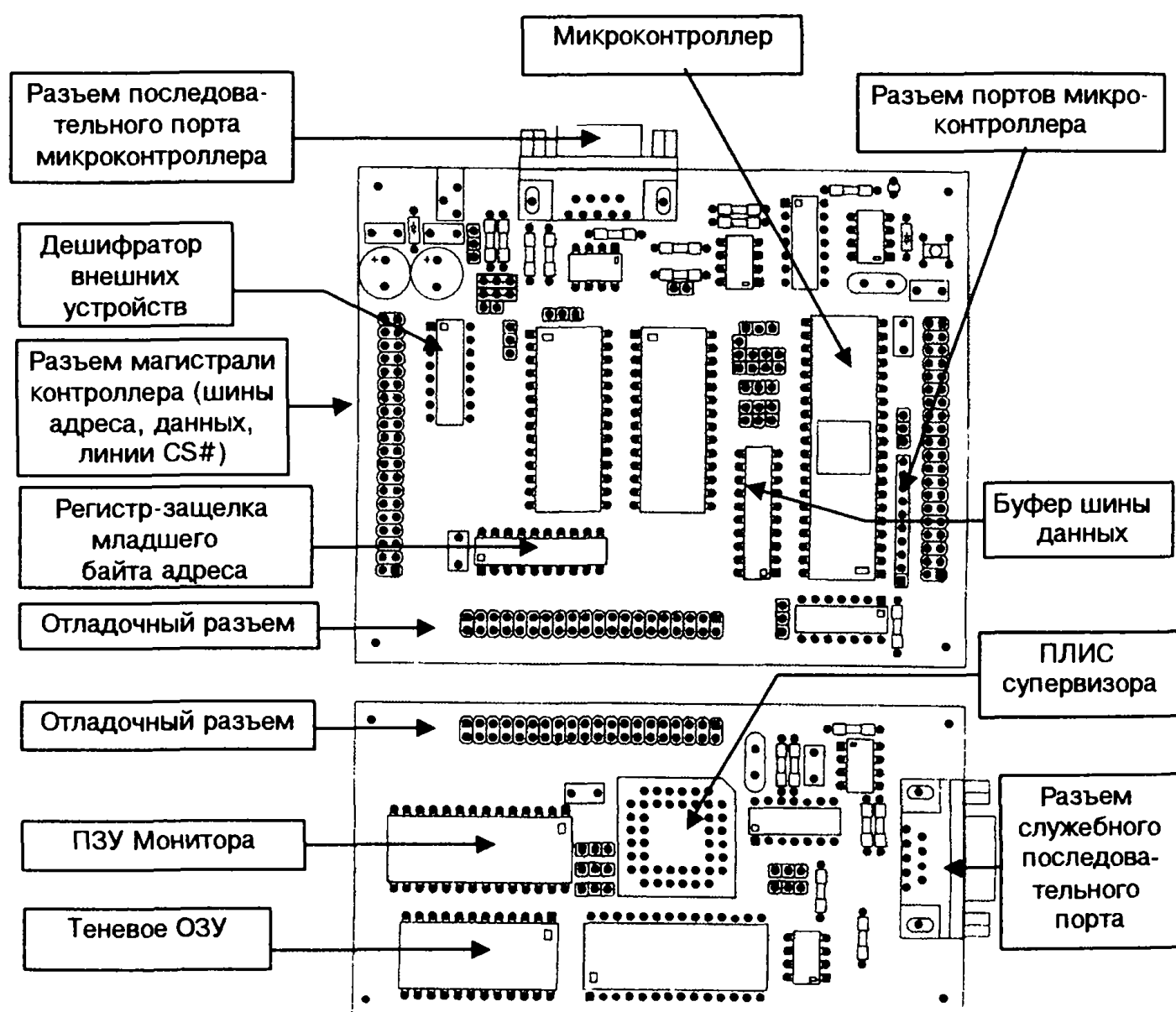


Рис. 6.6. Одноплатный контроллер и плата супервизора

Дешифратор адреса внешних устройств формирует 8 сигналов CS# для устройств ввода-вывода, которые выведены на разъем магистрали контроллера и могут быть использованы по усмотрению пользователя. Кроме того,

на этот разъем выведены демультимплексированные шины адреса и данных. На разъем портов микроконтроллера подключены небуферизованные сигналы непосредственно с выводов БИС.

Линии последовательного порта микроконтроллера через оптронную развязку выведены на разъем и всегда предоставлены в распоряжение пользователя.

Плата контроллера EV32_system может работать в двух режимах: в составе отладочного комплекса под управлением инструментального компьютера и самостоятельно, как процессор системы управления. При работе в составе комплекса плата контроллера подключается гибким кабелем к плате супервизора, которая формирует сигналы управления на отладочном разъеме по командам от компьютера. При этом в качестве памяти программ используется ОЗУ. После завершения отладки рабочей программы в розетку памяти программ устанавливается ПЗУ с «защитым» кодом, и контроллер может использоваться самостоятельно.

На этапе автономного использования в системе управления одноплатный контроллер отключается от платы супервизора, при этом обеспечиваются следующие технические характеристики:

Основные характеристики одноплатного контроллера EV-32_system

Микроконтроллер	8031/32/51FA
Напряжение питания, В	+5
Ток потребления, мА	70
Частота кварцевого резонатора, МГц	11,059
Температура эксплуатации, °С	0 +70 (-40 +80)
Габариты, мм	116 x 93 x 20

Плата супервизора EV32_sup подключается к контроллеру через специальный разъем и обеспечивает выполнение вместе с контроллером функций отладки. Плата супервизора (рис. 6.6) содержит собственный последовательный интерфейс с оптронной развязкой для связи с инструментальным компьютером, ПЗУ с управляющей программой Монитор, ПЛИС типа CPLD с супервизором и теневое ОЗУ. Она позволяет произвести отладку рабочей программы на аппаратуре контроллера с использованием следующих режимов:

- покомандное выполнение программы под управлением оператора,
- автоматическое покомандное выполнение программы под управлением компьютера,
- выполнение фрагмента программы в реальном времени до контрольной точки,
- прогон рабочей программы до ее конца.

Программа Монитор обеспечивает управление процессом отладки, устанавливает связь через один из последовательных портов инструментального компьютера с системой программирования-отладки. Монитор исполь-

зует прерывание INT1 микроконтроллера для обеспечения пошагового режима и выхода из режима прогона в реальном времени. На плате имеется переключатель, позволяющий отдать прерывание INT1 в распоряжение разработчика, но при этом теряется возможность трассировки и остановки программы (остается только возможность прогона в реальном времени без остановки), что допустимо на окончательной стадии отладки. При использовании пошагового режима на начальных этапах отладки адреса 13h–15h контроллера должны быть свободны от программного кода. В остальных требованиях к рабочей программе не отличаются от традиционных, описанных в документации к поставляемой с платой системе программирования «ТурбоАссемблер-51». При самостоятельной работе контроллера ограничений на использование адресного пространства не вводится (плата супервизора отключена).

Одноплатная система сбора/предобработки информации ЕС812 с присоединяемым супервизором. На основе микроконвертера AD μ C812 в лаборатории «Микропроцессорные системы» МИФИ создана одноплатная система сбора/предобработки информации ЕС812, которая имеет

- 8 аналоговых входов, подключаемых к 12-разрядному АЦП,
- два 12-разрядных ЦАП с повторителями на операционных усилителях,
- 8-разрядное процессорное ядро MCS-51 с 4-6 цифровыми линиями ввода/вывода (возможна плата-расширитель в виде мезонина, обеспечивающая до 32 цифровых линий),
- Flash-память программ (8 Кбайт), загружаемая через последовательный интерфейс,
- ОЗУ данных объемом 32–1024 Кбайт с возможностью хранения информации при батарейном питании,
- последовательные интерфейсы UART, I²C, SPI,
- стабилизатор напряжения питания, схему контроля питания и сторожевой таймер,
- отладочный разъем для подключения платы супервизора

Двухслойная плата размером 135*75 мм разработана с учетом требований точности измерений сигналов на аналоговых входах. В частности, входные аналоговые цепи экранированы, линии питания и земли аналоговой и цифровой частей развязаны фильтрами. Входные цепи аналоговых входов представляют собой RC-цепочки. На плате предусмотрена установка мезонина, на котором, при необходимости, могут располагаться цепи для масштабирования входных сигналов и организации дифференциальных входов. Плата выполнена с использованием элементов технологии поверхностного монтажа и демонстрирует современный подход к проектированию микропроцессорных контроллеров на основе функционально-топологических модулей и программных модулей.

Важнейшим условием применения новых микропроцессорных средств является наличие средств отладки. Использование инструментальных средств MCS-51 для отладки систем на основе семейства AD μ C не тривиально из-за закрытого характера архитектуры современных микроконтроллеров с внутренней flash-памятью программ и отсутствия эмуляционных кристаллов. Однако, если совместно с таким микроконтроллером применяется внешняя память данных, то наличие шин адреса и данных позволяет присоединить внешний супервизор, аппаратно поддерживающий операции отладки.

Система отладки для EC812, включает плату супервизора нового типа и интегрированную систему программирования ТурбоАссемблер-812. Для подключения системы отладки на плате EC812 имеется отладочный разъем.

После отладки рабочая программа загружается во внутреннюю flash-память микроконвертора AD μ C812 через разъем порта RS232, который после этого может работать в режиме обычного последовательного интерфейса. При автономной работе информационно-управляющей системы данные, принятые по входным аналоговым и цифровым линиям, а также сформированная внутри микроконвертора информация (например, временные отсчеты), заносятся во внешнюю память данных. Размер этой памяти на рассматриваемой плате может быть от 32 Кбайт до 1 Мбайта, в зависимости от типа установленных микросхем. В процессе измерений или после его завершения эта информация может быть передана в инструментальный компьютер через интерфейсы RS232, I²C, SPI. На основе этих интерфейсов из нескольких одноплатных систем может быть создана распределенная информационно-управляющая микроконтроллерная сеть.

6.3.2. Эмулятор ПЗУ/логический анализатор на основе FPGA

До настоящего времени в качестве эмуляторов ПЗУ предлагались простые устройства, обеспечивающие только загрузку памяти и переключение системы в рабочий режим. Такие устройства не реализуют всех функций эмуляции, являются по существу имитаторами с примитивным управлением и не представляют особого интереса с точки зрения отладки микропроцессорных систем.

Новый бум в области микропроцессоров и микроконтроллеров, связанный с качественно новыми возможностями микроэлектроники, привел к появлению в последнее время большого количества разнообразных микропроцессорных БИС. Они принадлежат к различным архитектурам и в каждом семействе используется значительная номенклатура корпусов. Быстро создать схемные эмуляторы для всех микропроцессорных БИС невозможно, поэтому возродился интерес к эмуляторам ПЗУ.

Параллельно с микропроцессорами бурно развивается направление БИС программируемой логики. Использование БИС типа FPGA позволяет реализовать эмулятор ПЗУ с настройкой функции замещения под широкую гамму типов микросхем памяти, с реализацией функции наблюдения в виде памяти трасс и функции управления работой целевого микропроцессора при наличии у него сигналов типа READY.

В нашей лаборатории реализован эмулятор ПЗУ «RET» (ROM-Emulator-Tracer) на основе FPGA типа EPF8282 фирмы Altera.

Эмулятор представляет собой небольшой аппаратный модуль (размером 70*90*20) в корпусе фирмы «Каскад», подключаемый к инструментальному компьютеру через интерфейс Centronics (рис. 6.7).

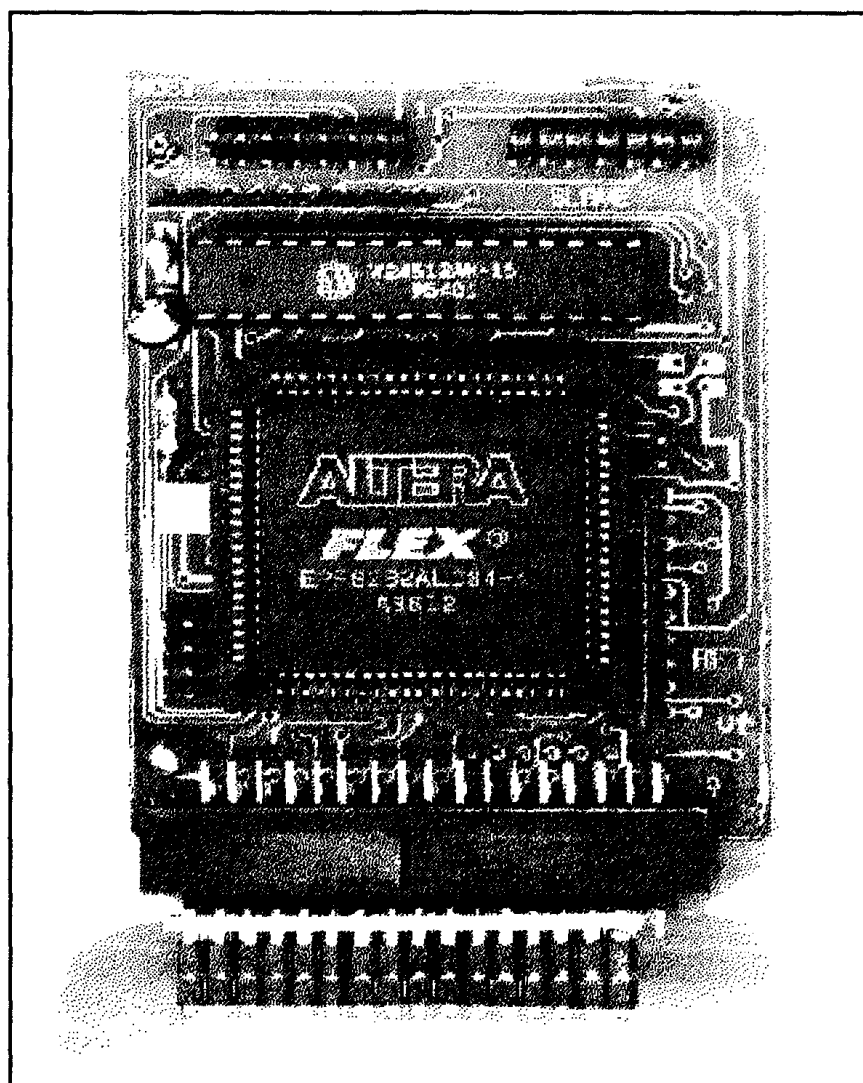


Рис. 6.7. Плата эмулятора ПЗУ «RET»

Управляющая программа обеспечивает работу с файловой системой инструментального компьютера, загрузку файла с кодом в эмулятор и контроль загрузки, чтение и просмотр в двоичном коде снятой трассы выполнения отлаживаемой программы. В качестве эмулирующей памяти использовано статическое ОЗУ объемом 64 Кбайта (возможно 128 Кбайт) с вре-

менем доступа 15 нс. Предусмотрено разрядное наращивание системы отладки посредством параллельного соединения двух эмуляторов, что обеспечивает эмуляцию 16-разрядной памяти.

Основные характеристики эмулятора RET

- объем эмулируемой памяти – 64 (128) Кбайт,
- время выборки адреса – 35 нс,
- потребляемый ток – не более 25 мА (1 мА в режиме ожидания),
- эмулируемые ПЗУ – K573РФ2 – 8, K556PT5 – 7/16–18/161, 2716/32 – 512/010, 28C64/256,
RAM – 2Kx8-128Kx8,
- объем трассировочной памяти – 32Kx8 в режиме эмуляции ПЗУ, 64Kx8 в режиме логического анализатора

Имеется три возможных варианта использования прибора

1. Эмуляция памяти программ с трассировкой выполнения программы. Преимуществом использования программируемой логики типа FPGA является возможность замещения широкой номенклатуры БИС ПЗУ с точки зрения комбинаций (топологии) выводов адреса, данных, управления. Ограничением является местоположение вывода «Земля» на разъеме эмулятора ПЗУ и объем эмулирующего ОЗУ. Эмулятор RET, кроме стандартного набора БИС с рекомендованной JEDEC топологией выводов (K573РФxx и т.п.), поддерживает также микросхемы ПЗУ с плавкими перемычками серий K556, K1623, K541, K1608. Имеется возможность эмуляции и памяти данных (статические ОЗУ с 8-разрядной организацией).

В этом режиме, если есть запас по объему эмулирующей памяти, она может быть поделена на две матрицы по 32 Кбайта (64 Кбайта – опция). Одна из матриц служит для замещения памяти целевой системы, а другая используется как память трасс. Поскольку адреса памяти (по которым производилось обращение) являются 16-разрядной информацией, память трасс имеет организацию 16К*16.

Трасса просматривается в двоичном коде. Имеется два режима трассировки:

- от начала программы до заполнения трассировочной памяти
- постоянная трассировка с перезаписью информации

Условием начала трассировки является переход сигнала ОЕ в пассивное состояние. Игнорирование сигнала CS# (задается программно) позволяет трассировать обращения к другим (не эмулируемым) матрицам памяти в системе.

Для реализации функции управления отлаживаемой системой предусмотрен специальный разъем, позволяющий перехватить два сигнала управления магистрали системы. Условиями полной эмуляции являются

- подключение указанных линий к сигналам типа ALE и READY отлаживаемой системы,
- память программ системы должна полностью замещаться эмулятором ПЗУ

2. Режим логического анализатора. Прибор способен работать в режиме 8-разрядного анализатора состояний с глубиной рабочей памяти 64 Кбайт. Частота синхронизации (до $F_{\max} = 30$ МГц) подается при этом от исследуемого устройства. Разрабатывается вариант 30-разрядного анализатора с глубиной памяти 16 Кслов ($F_{\max} = 8$ МГц). При отладке большинства типов микропроцессорных систем этого вполне достаточно, поскольку частота обращений к памяти и устройствам ввода/вывода определяется временем магистрального цикла и составляет не более 4–8 МГц. Запуск и останов записи в буферную память производится по внешнему событию. Таким событием может являться

- появление заданного кода на шине данных,
- появление определенного кода на двух линиях внешних квалификаторов, (возможно программирование условия события на перепады «0» – «1» или «1» – «0»),
- переполнение буферной памяти

3. Режим Evaluation Kit для ПЛИС серий EPM 7000 и 8000. Разработчики, использующие программируемую логику типа EPM 7000 и EPF 8000 фирмы Altera могут быстро провести натурные испытания устройства управления, загрузив файл, созданный при помощи системы MAX+PlusII в БИС программируемой логики прибора RET. Разрабатываемое устройство (или фрагмент) должно при этом иметь не более 30 линий ввода/вывода.

К целевой системе эмулятор RET подключается с помощью сменных эмуляционных вилок на 28 (32, 24) контактов. В комплект входит 28-выводная вилка на гибком кабеле и 32-выводная вилка для непосредственного соединения эмулятора и целевой платы. Питание эмулятора может осуществляться от исследуемой платы, от внешнего источника напряжением +5 В (50 мА), или инструментального компьютера через разъем GAME.

6.3.3. Схемные эмуляторы

Это в настоящее время наиболее совершенное средство комплексной отладки аппаратного и программного обеспечения микропроцессорных систем в реальном времени. Схемный эмулятор подключается к целевой плате гибким кабелем с эмуляционной вилкой, которая вставляется в розетку вместо микропроцессора или микроконтроллера. Другим кабелем он подключается к инструментальной ЭВМ, на которой работает управляющая программа. Встречаются эмуляторы, выполненные в виде плат расширения

инструментальной персональной ЭВМ. Схемный эмулятор управляет выполнением рабочей программы и наблюдает за состоянием отлаживаемой системы через магистраль. Поэтому перед его использованием должен быть выполнен этап автономной отладки аппаратуры и магистраль системы должна функционировать правильно.

Специализированные схемные эмуляторы. В лаборатории «Микропроцессорные системы» МИФИ в конце 80-х, начале 90-х годов были разработаны схемные эмуляторы для микроконтроллеров семейств Intel MCS-51 и MCS-48, микропроцессора 8085. Каждый из них представлял собой самостоятельный прибор, предназначенный для отладки систем на основе определенной микропроцессорной БИС. Структура этих эмуляторов представляет интерес и в настоящее время, поскольку воплощает все функции приборов этого типа в развернутом виде.

Рассмотрим структуру специализированного схемного эмулятора МЭ-31 (рис. 6.9). Основой схемного эмулятора является *имитирующий процессор*, замещающий целевой микроконтроллер. Имитирующий процессор может быть реализован различными способами, в том числе на основе микропрограммируемой структуры и посредством реализации функций целевого контроллера на основе БИС программируемой логики. Для микропроцессоров и микроконтроллеров без внутренней памяти программ в качестве имитирующего процессора можно использовать микропроцессорную БИС, аналогичную целевой, но установленную на плате эмулятора.

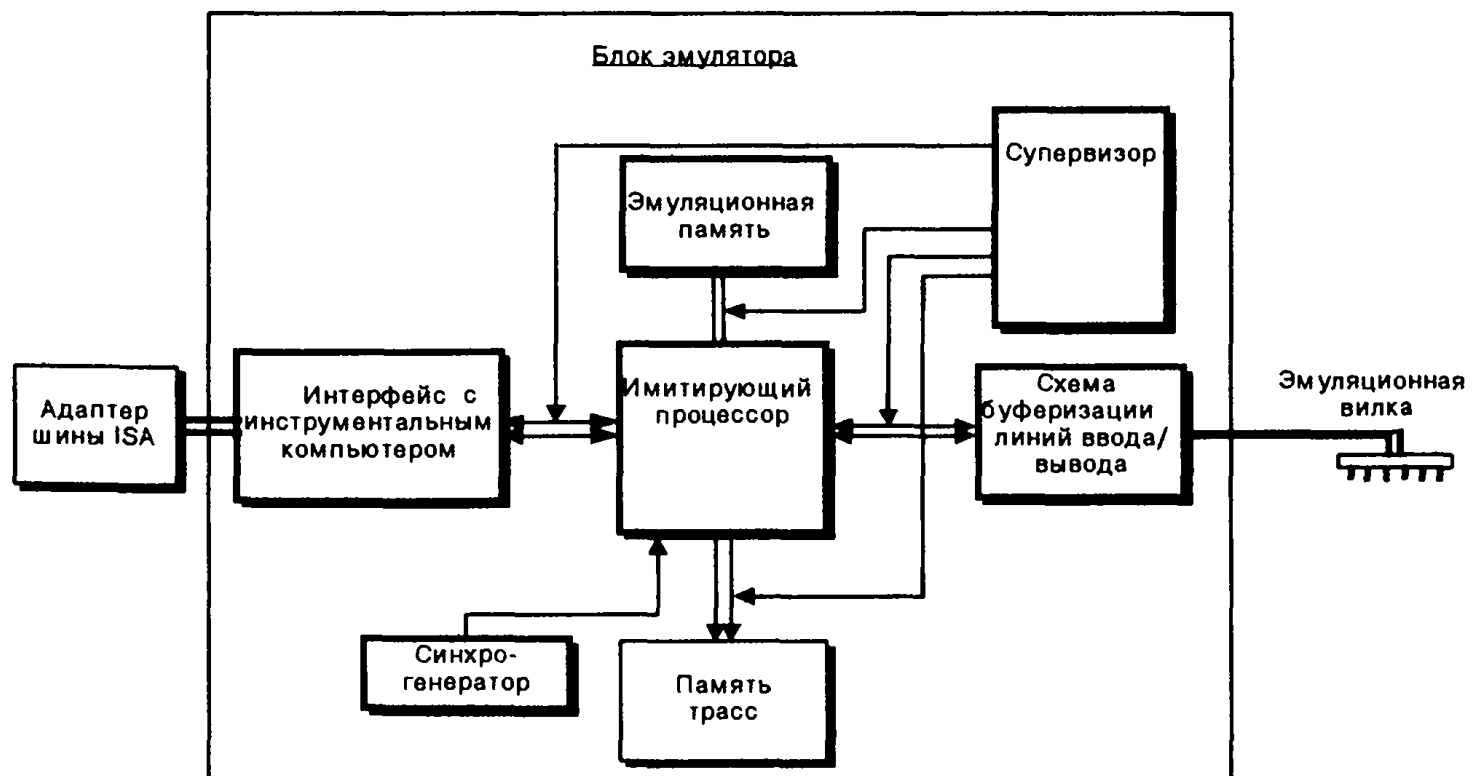


Рис. 6.9 Структура специализированных эмуляторов МЭ-31, МЭ-51, МЭ-85

Имитирующий процессор для микроконтроллеров с внутренней памятью программ должен иметь дополнительные порты ввода/вывода, позво-

ляющие выполнять функции управления и наблюдения в реальном времени без искажения процесса выполнения рабочей программы. Поэтому для эмуляции микроконтроллеров с внутренней памятью типа 8051/8751 в эмуляторе МЭ-51 в качестве имитирующего процессора использованы 6-портовые БИС с архитектурой MCS-51.

Схемный эмулятор, кроме процессора, замещает память программ и данных. Для этого служит *эмуляционная память* эмулятора. В целях отладки она разбита на сегменты, *карта памяти* позволяет процессору обращаться либо к ячейкам эмуляционной памяти, либо к ячейкам памяти, размещенной на целевой плате. В рассматриваемых эмуляторах эмуляционная память программ девятиразрядная. Старший разряд обслуживает механизм *контрольных точек*, в которых происходит останов процессора при выполнении программы в прогоне.

Буфером, в который в реальном времени записывается информация о последовательности выполненных команд рабочей программы, является *память трасс*. Она организована по принципу кольца, т. е. при превышении объема информации младшие ячейки переписываются. Ее объем в описываемых приборах составляет 2047 ячеек.

Функции управления и наблюдения в реальном времени выполняет в рассматриваемых эмуляторах блок *супервизора*. Он отслеживает поток данных по магистрали, анализирует формат выбираемых команд и осуществляет контекстное переключение с рабочей программы на служебную. Слежение за выборкой кода операции и формирование управляющих сигналов осуществляется с использованием микропрограммного автомата. При таком подходе микропрограммируемый супервизор существует и работает параллельно процессорно-зависимой части схемного эмулятора.

Для передачи информации по кабелю между блоком эмулятора и эмуляционной вилкой необходимо увеличить нагрузочную способность портов микроконтроллера. Для этого используется *схема буферизации линий ввода/вывода микроконтроллера*.

При останове операции чтения/записи инструментальным компьютером ресурсов эмулятора регистров микроконтроллера, ячеек эмуляционной памяти, памяти трасс (только чтение) — выполняются через интерфейс связи с инструментальным компьютером. В слот расширения компьютера вставляется адаптер шины ISA, определяющий диапазон адресов эмулятора в адресном пространстве компьютера и обеспечивающий параллельный обмен между блоком эмулятора и компьютером по кабелю значительной длины. Использование параллельного обмена позволяет сократить время обновления информации на экране и высвобождает штатные порты компьютера.

Все специализированные схемные эмуляторы в лаборатории «Микропроцессорные системы» создавались на основе функционально-модульного подхода как на аппаратном, так и на программном уровнях. Это позволило далее перейти к интеграции процессорно-независимой части эмуляторов на

универсальной платформе, а изменяющуюся часть реализовать в виде набора плат специализации

Схемные эмуляторы на основе универсальной платформы. Основным преимуществом такого решения является значительное снижение финансовых затрат пользователя при переходе с одного типа микропроцессора или микроконтроллера на другой. Если при покупке каждого специализированного эмулятора приходится оплачивать весь аппаратно-программный комплекс, то имея эмулятор на универсальной платформе можно ограничиться приобретением только плат специализации и ПО

Основой структуры эмуляторов этого типа является модуль супервизора, реализованный с использованием БИС программируемой логики типа FPGA (рис 6 10) Высокое быстродействие этой элементной базы позволило поместить супервизор в центр структуры универсального эмулятора и коммутировать через него потоки информации между остальными модулями, а также формировать системные управляющие сигналы

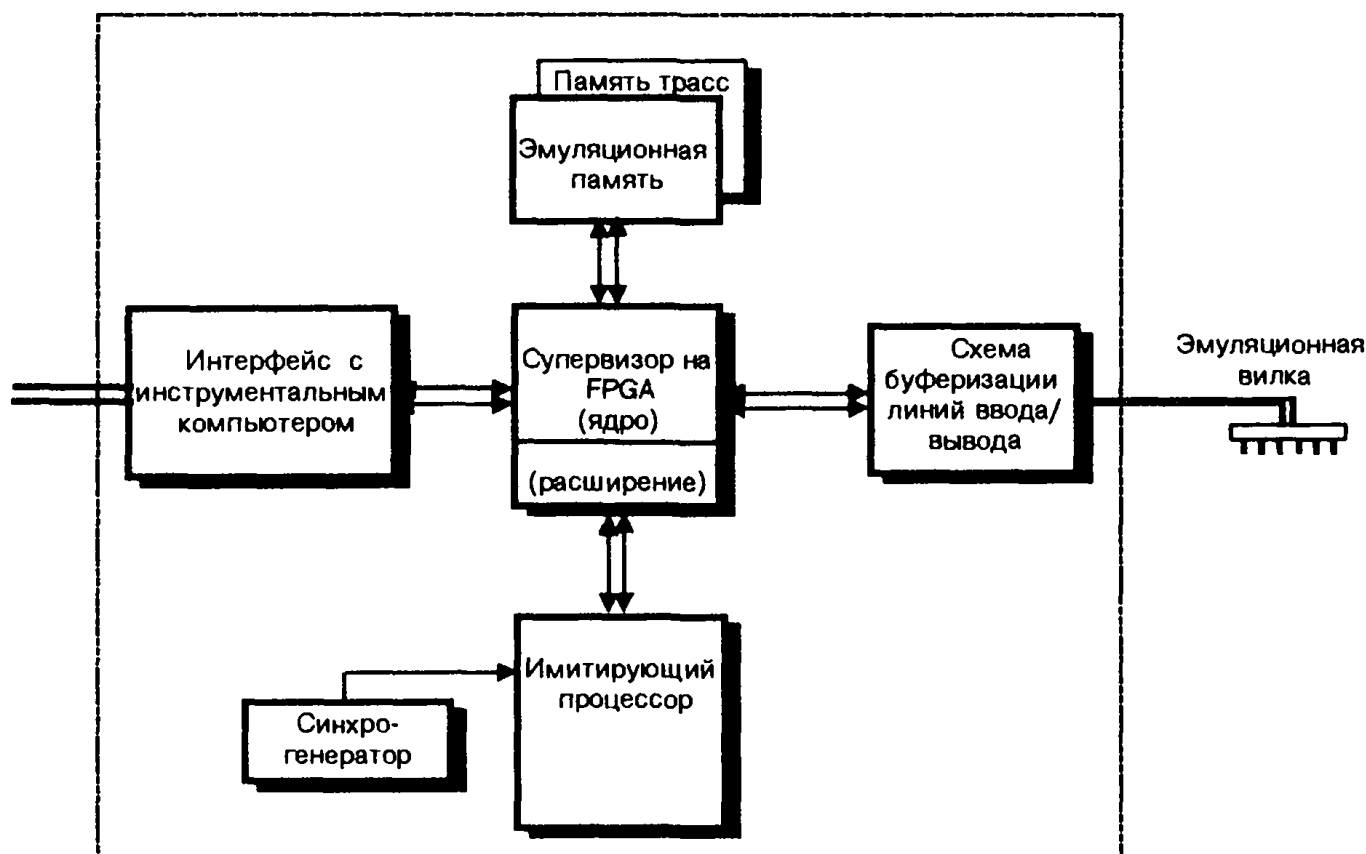


Рис 6 10 Обобщенная структура универсального эмулятора на основе FPGA

Возможность изменять функции супервизора на основе FPGA загрузкой файла конфигурации из инструментального компьютера привело к идее объединить на универсальной платформе процессорно-независимую часть эмуляторов с перестраиваемым супервизором, а изменяющуюся часть реализовать в виде набора плат специализации. При этом сам супервизор в зависимости от архитектуры эмулируемой микропроцессорной БИС может

состоять только из функционального ядра, связанного с управлением матрицами служебной памяти, либо иметь еще компоненту расширения, выполняющую дополнительные функции при эмуляции БИС со сложными протоколами обмена (в частности, анализ кода операции в эмуляторах семейства MCS-51)

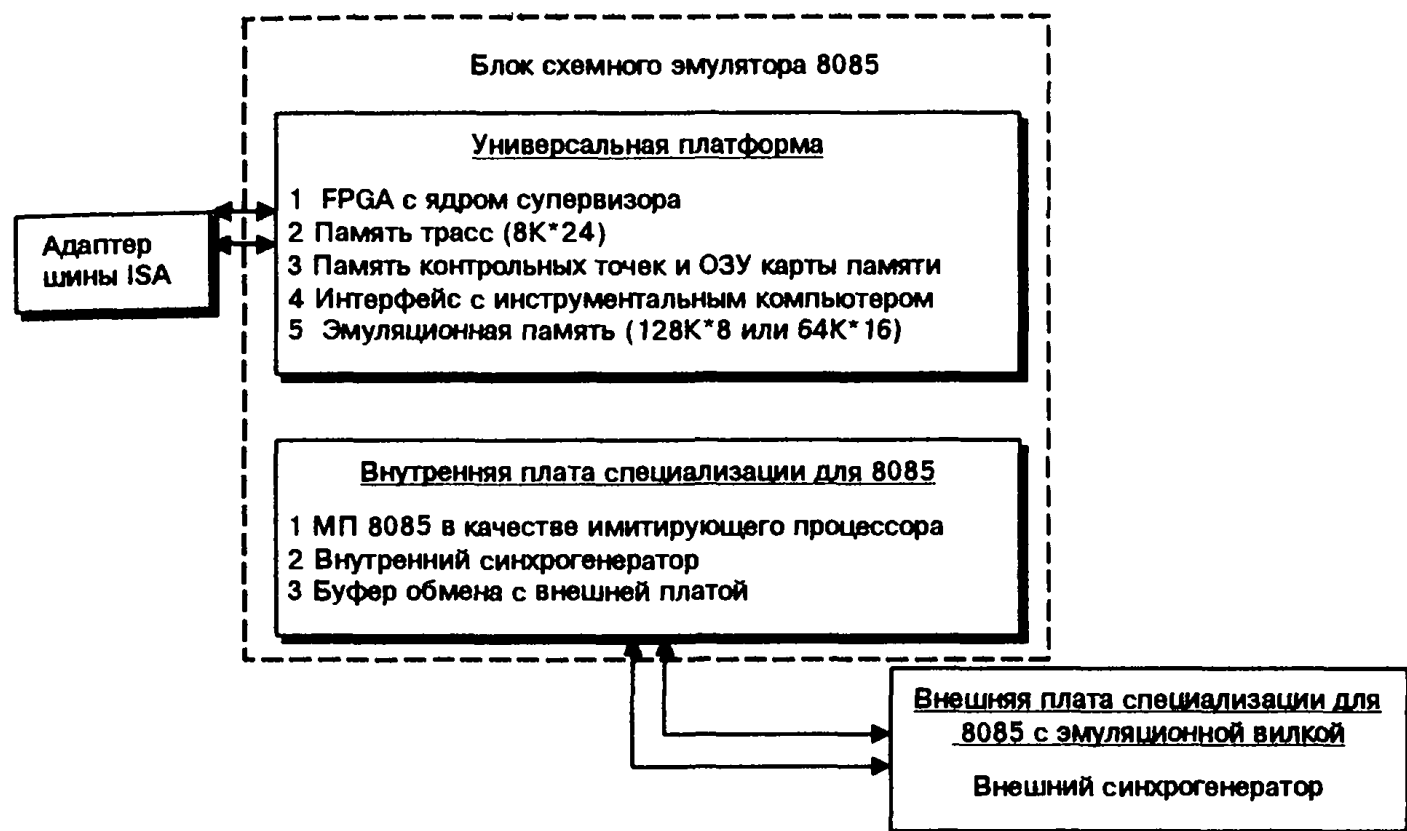
Структурные схемы эмуляторов на основе универсальной платформы с супервизором на FPGA приведены на рис 6 11 В настоящее время существует два варианта реализации эмуляторов для микропроцессоров (на рис 6 11а приведена структура эмулятора для 8085) и для микроконтроллеров семейства MCS-51 (на рис 6 11б приведена структура эмулятора для 8051)

Новые эмуляторы подключаются к инструментальному компьютеру через адаптер шины ISA или порт LPT Универсальная платформа содержит БИС программируемой логики типа FPGA с ядром супервизора (осуществляет управление блоками памяти), эмуляционную память (может конфигурироваться в матрицы 128К*8 или 64К*16), память контрольных точек и трасс (матрица 8К*24), а также схему интерфейса с инструментальным компьютером В блоке эмулятора к платформе через два разъема всегда присоединяется внутренняя плата специализации В эмуляторе микропроцессора 8085 на ней установлен имитирующий микропроцессор и синхрогенератор, а в эмуляторах микроконтроллеров семейства MCS-51 – БИС FPGA расширения супервизора с дополнительной матрицей служебной памяти и буферы внутренних шин данных/адреса эмулятора

У всех эмуляторов имеется внешняя плата специализации с эмуляционной вилкой, соединенная с основным блоком гибким кабелем В эмуляторе 8085 на ней размещается только синхрогенератор, работающий с кварцем целевой системы В эмуляторах MCS-51 на внешней плате находится БИС имитирующего процессора, синхрогенератор и буферы системных шин В настоящее время имеется три типа внешних плат для эмуляции микроконтроллеров с архитектурой MCS-51 POD31/32 для микроконтроллеров с внешней памятью программ, POD51/52 для микроконтроллеров Intel с внутренней памятью программ, POD552 для микроконтроллеров фирмы Philips

На блоке эмулятора установлен разъем типа DB-15, через который в память трасс может синхронно с состояниями имитирующего процессора записываться информация о состоянии восьми внешних линий (уровни КМОП или ТТЛ) На этом же разъеме имеется вход, активный низкий уровень на котором может остановить прогон рабочей программы

В качестве программ управления эмуляторами может использоваться три типа систем программирования оболочка, объединяющая внешние редактор текста, кросс-ассемблер, редактор связей и модель, интегрированная система ТурбоАссемблер (с библиотеками), интегрированная система ТурбоПаскаль (с библиотеками)



а)



б)

Рис 6 11 Структурные варианты схемных эмуляторов на основе универсальной платформы а) эмулятор микропроцессора 8085, б) эмулятор микроконтроллера 8051

Стоимость перехода на новый тип микропроцессора или микроконтроллера. Оценку стоимости инструментальных средств при переходе на новый тип микропроцессора или микроконтроллера проведем при следующих допущениях

- примем значения стоимости специализированных схемных эмуляторов, эмулятора микропроцессора 8085 на универсальной платформе, а также плат специализации для построения на универсальной платформе эмуляторов Intel 8051 и Philips 8X552, указанные в таблице

Аппаратура	Относительная стоимость
1 Специализированный эмулятор Intel 8051	1,0
2 Специализированный эмулятор Philips 8X552	1,0
3 Специализированный эмулятор Intel 8085	0,8
4 Эмулятор на универсальной платформе Intel 8085	0,9
5 Платы специализации для эмулятора Intel 8051	0,5
6 Внешняя плата специализации для Philips 8X552	0,2

- допустим, что на первом этапе был приобретен эмулятор на универсальной платформе для микропроцессора Intel 8085 (K1821BM85) Тогда для перехода на микроконтроллер Intel 8051 необходимо приобрести внутреннюю плату специализации для MCS-51 и внешнюю плату для Intel 8051 (POD51), а при последующем переходе на микроконтроллер Philips 8X552 – только внешнюю плату (POD552)

Стоимость перехода на новый тип микроконтроллера при указанных допущениях иллюстрирует рис 6 12

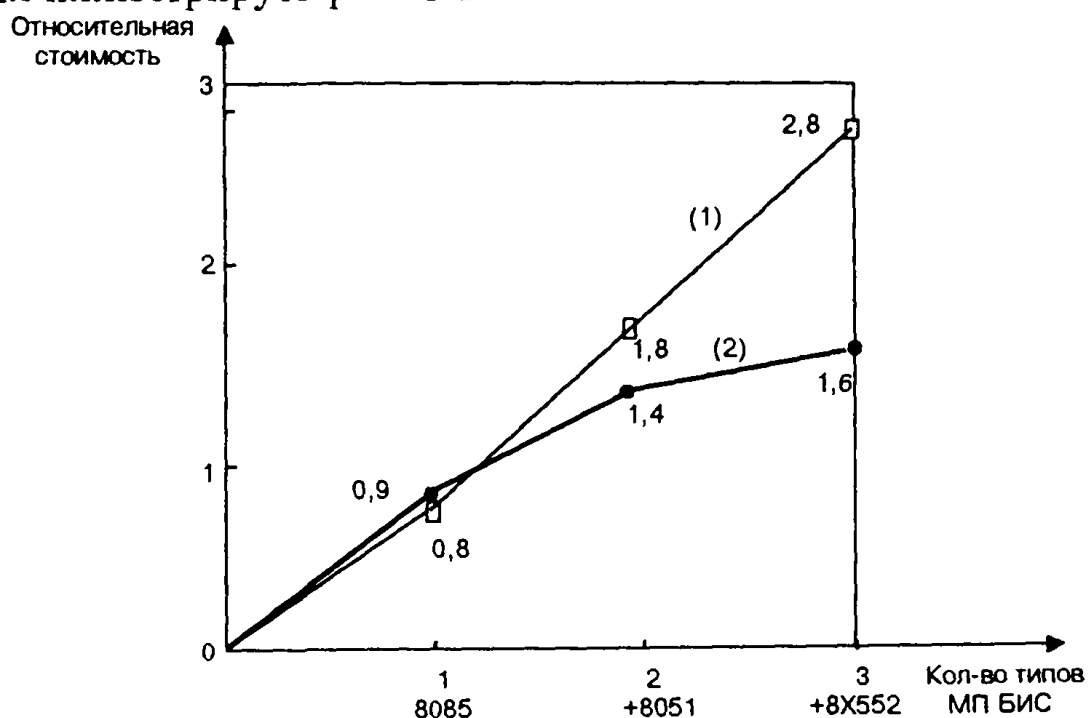


Рис 6 12 Стоимость перехода на новый тип микроконтроллера при использовании специализированных эмуляторов серии МЭ (1), при использовании эмуляторов серии К на универсальной платформе (2)

Из рис 6 12 видно, что при использовании в практике проектирования микропроцессорных систем трех распространенных в нашей стране типов 8-разрядных микропроцессоров и микроконтроллеров относительная стоимость комплекта эмуляторов на универсальной платформе ниже комплекта специализированных эмуляторов в 1,75 раза

6.4. Средства программирования микросхем энергонезависимой памяти, микроконтроллеров и ПЛИС

В настоящее время существуют два метода программирования микросхем энергонезависимой памяти, микроконтроллеров и ПЛИС.

- внутрисистемное программирование с использованием встроенных последовательных интерфейсов SPI, I²C, JTAG,
- параллельная загрузка с помощью программатора

Внутрисистемное программирование. Этот метод программирования в последнее время становится все более популярным. Причина заключается во все более широком распространении flash-памяти и микросхем с планарными корпусами типа SOIC и QFP, которые программируются после пайки выводов. Механизм внутрисистемного программирования в совокупности с наличием матрицы репрограммируемой памяти существенно повышает эффективность процесса отладки и является единственной возможностью наращивания характеристик изделия (upgrade) без перепайки микроконтроллера или ПЛИС. Наибольшую пользу преимуществ внутрисистемного программирования реализованы в микроконтроллерах Atmega163 фирмы Atmel, где имеется возможность создания загрузочного драйвера пользователем. Это открывает широкие возможности для методов отладки при удаленном доступе по сети, а также для использования криптографических методов защиты информации во внутренней памяти программ.

Для внутрисистемного программирования через интерфейс RS232C используется, как правило, программа-монитор (firmware), встроенная в микроконтроллер. Примерами являются БИС AduC812 фирмы Analog Devices, T89C51RD2 фирмы Temic, P89C51RD+ фирмы Philips. При этом вход в режим программирования производится при наличии специальной комбинации сигналов на входах (PSEN=0) в момент перехода сигнала сброса (Reset) в пассивное состояние.

Внутрисистемное программирование через интерфейс SPI характерно для всех микроконтроллеров AVR и микроконтроллеров серии AT90Sxx фирмы Atmel. Загрузка памяти производится при активном сигнале Reset.

Отличительной особенностью перечисленных кристаллов является то, что стирание и программирование памяти производится при номинальном

напряжении питания (+5 В), так как преобразователь напряжения $+5 \text{ В} \Rightarrow V_{pp}$ (необходимый для записи информации в матрицу памяти) размещен на кристалле микроконтроллера. Сравним для программирования микроконтроллера Philips P89C558, разработанного более 5 лет назад, требуется внешнее напряжение программирования $V_{pp} = +12 \text{ В}$. Упомянутые микроконтроллеры кроме режима внутрисистемного программирования поддерживают также параллельное программирование, ориентированное на применение программатора. Типовая схема организации внутрисистемной загрузки flash-памяти программ микроконвертера AD μ C812 приведена на рис 6 13.

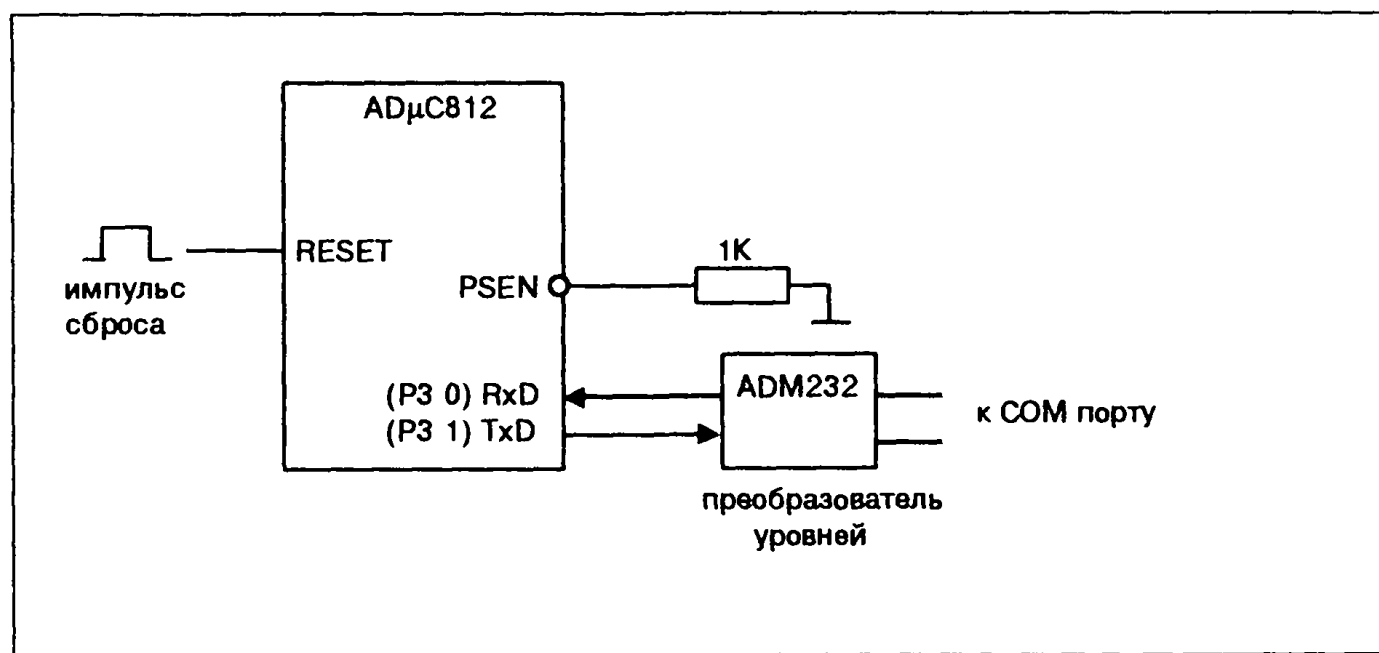


Рис 6 13 Внутрисистемная загрузка микроконвертера AD μ 812

Интерфейс типа I²C используется для программирования микроконтроллеров PIC фирмы Microchip. При этом для большинства типов микросхем требуется внешнее напряжение $V_{pp} = +13 \text{ В}$, а параллельный режим отсутствует.

Общая тенденция внедрения функций тестирования и отладки в серийные БИС, в том числе микроконтроллеры и ПЛИС, стимулировала разработку стандартного интерфейса JTAG (Joint Test Access Group). Данный стандарт обеспечивает выполнение функций загрузки памяти и отладки программы в реальном времени в некоторых микроконтроллерах, например Texas Instruments MSP430. Наиболее широко интерфейс JTAG используется для загрузки ПЛИС с энергонезависимым хранением конфигурации (Altera, Xilinx, Lattice и др.). В результате появляется очевидный выигрыш для пользователя: загрузка конфигурации в различные ПЛИС одной фирмы производится через один загрузочный кабель, причем, у каждой фирмы кабель уникальный. «Размывать границы» начала фирма Atmel, выпустив ПЛИС серии ATF1500, совместимые по выводам с ПЛИС

MAX7000 фирмы Altera. Вместе с тем, программа загрузки для ATF1500 может работать как с собственным кабелем Atmel, так и с кабелем типа ByteBlaster, специфицированным фирмой Altera. Типовая схема организации внутрисистемной загрузки конфигурации ПЛИС серий MAX7000S, ATF1500 приведена на рис 6 14. Единственным ограничением применения внутрисистемной загрузки ПЛИС является необходимость резервирования линий интерфейса JTAG. Это значит, что порты ПЛИС с предопределенными функциями JTAG (TMS, TDI, TDO, SCK) не могут быть использованы в проекте пользователя, в то время, как использование параллельной загрузки позволяет программировать их как универсальные порты ввода/вывода.

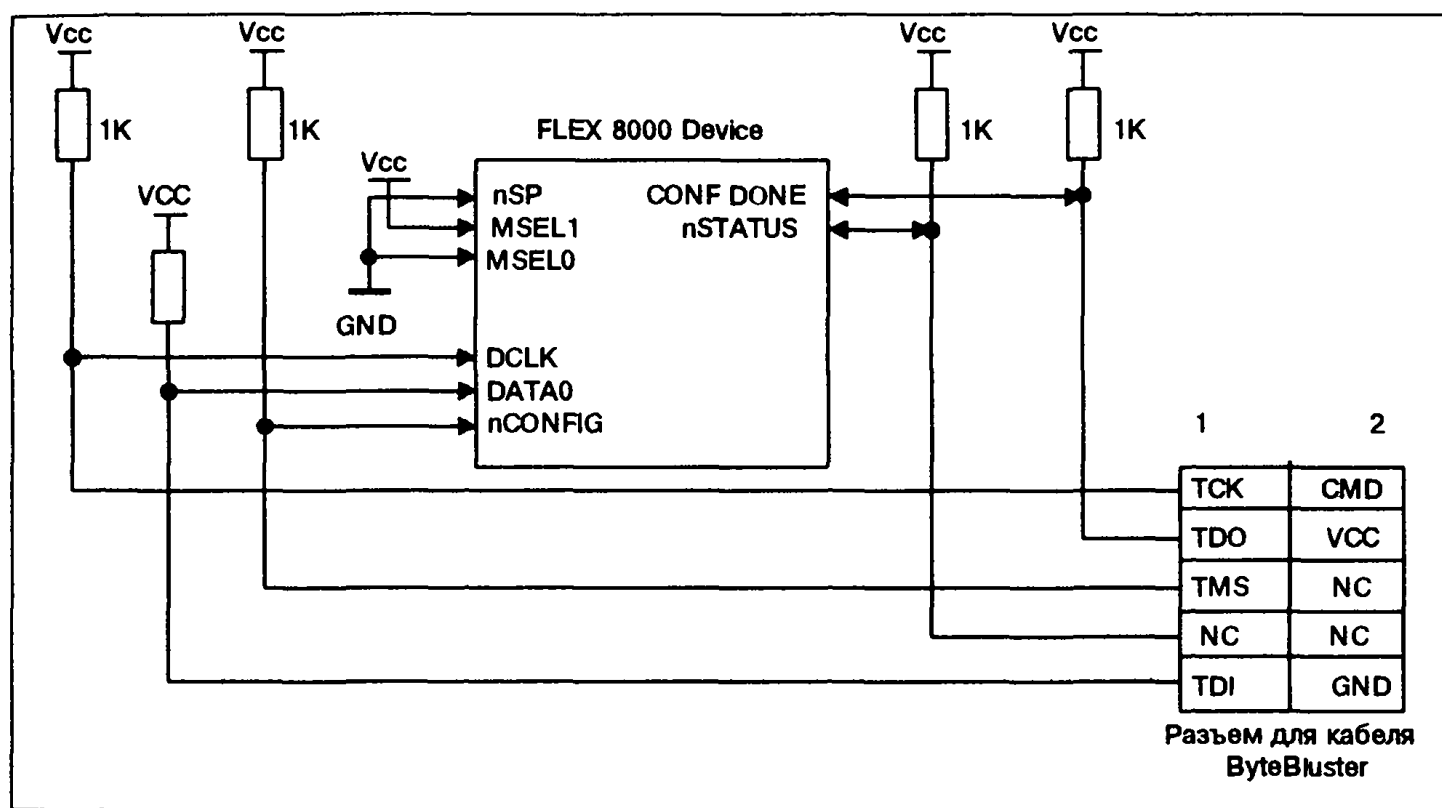


Рис 6 14 Внутрисистемная загрузка ПЛИС

Чтобы поддержать унификацию загрузки различных типов микросхем в лаборатории «Микропроцессорные системы» принято использовать кабель ByteBlaster для внутрисистемного программирования микроконтроллеров с интерфейсом, аналогичным SPI. Например, программные пакеты «Турбо-Ассемблер AVR», «ТурбоАссемблер 51/52» поддерживают загрузку микроконтроллеров фирмы Atmel (AVR, AT89Sxx) через ByteBlaster. Принципиальная схема кабеля приведена на рис 6 15.

Микросхемы EEPROM с последовательным доступом также можно программировать внутрисистемно. Для этого принципиальная схема целевого устройства должна обеспечивать пассивное состояние интерфейса в момент загрузки, например, удерживать ведущий микроконтроллер в состоянии сброса. К таким микросхемам можно отнести серии 24Cxx, 85Cxx.

(I2C), 25Cxx (SPI), 93Cxx (Microwire, эмулируется с помощью SPI), а также AT17Cxx – конфигурационные ПЗУ для ПЛИС с энергозависимым хранением информации

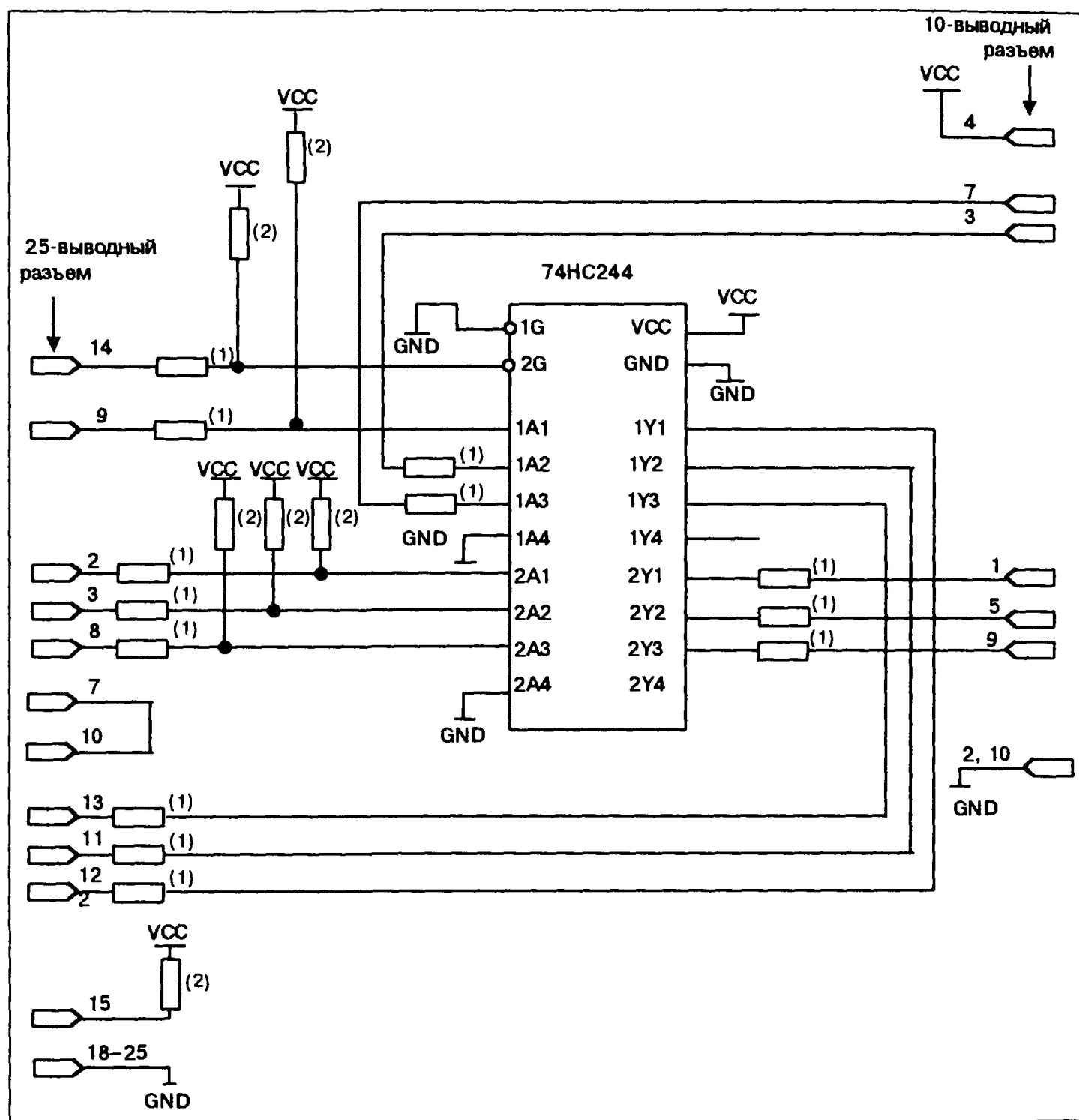


Рис 6 15 Схема устройства типа ByteBlasterMV

Все перечисленные серии микросхем программируются при номинальном напряжении питания/программирования (+5 В)

Программаторы. В настоящее время в продаже имеется несколько типов программаторов, имеющих относительно невысокую цену. Каковы критерии выбора программатора, который может использоваться при разработке, изготовлении и обслуживании промышленной аппаратуры?

Современные профессиональные программаторы должны, по нашему мнению, отвечать следующим требованиям

- программировать в соответствии с требованиями ТУ возможно более широкую номенклатуру БИС, в том числе в корпусах различных типов,
- обеспечивать наращивание номенклатуры программируемых БИС без существенного увеличения аппаратных средств и стоимости прибора,
- работать с входными файлами стандартных типов (hex, bin, jed и т.д.),
- верифицировать запрограммированную микросхему на соответствие кода входному файлу, в том числе при изменении напряжения питания микросхемы в рабочем диапазоне,
- иметь удобный интерфейс управляющей программы,
- иметь высококачественные розетки под корпуса различных типов,

Важнейшими являются требования универсальности и соблюдения режимов программирования, указанных в документации на микросхемы. Современные разработчики-профессионалы используют всю гамму программируемых БИС для достижения максимально высоких характеристик изделия. Коллектив действует в условиях сильной конкуренции, жестких требований по срокам разработки и стоимости изделия. Складывается набор функционально-топологических и программных модулей, а конкретный контроллер быстро «собирается» с использованием программируемых БИС. Следует заметить, что универсальность в данном случае подразумевает программирование полного набора важнейших типов программируемых микросхем, а не длинный перечень однотипных изделий, выпускаемых разными производителями (и взятый из списка аналогов). В набор важнейших типов в настоящее время входят

- БИС микроконтроллеров с репрограммируемой и flash-памятью,
- микросхемы памяти с ультрафиолетовым стиранием,
- микросхемы памяти с электрическим репрограммированием,
- микросхемы flash-памяти,
- микросхемы памяти, программируемые пережиганием перемычек,
- БИС программируемой логики – ПЛИС

Что касается соблюдения режимов программирования, рекомендованных изготовителями микросхем, то анализ схемотехники ряда программаторов показывает, что в них отсутствует источник напряжения +6,25 В, необходимый для программирования популярных типов БИС памяти (например, серии 27Сxxx). Кроме того, отсутствие цифроаналогового преобразователя в структуре программаторов не позволяет устанавливать напряжение программирования с точностью 0,25 В, а это требование специально оговаривается в документации на микросхемы. Таким образом можно заключить, что многие программаторы, имеющиеся в продаже, не обеспечивают соблю-

дение режимов программирования, указанных в документации на микросхемы

Возможность наращивания функций уже купленного прибора всегда полезна, но для программаторов это жизненно необходимо, поскольку имеющий место бум микроэлектроники приводит к непрерывной замене версий уже существующих типов микросхем и появлению новых семейств

Первые два из перечисленных выше условий приводят к необходимости формирования различных диаграмм программирования и коммутации сигналов на различные выводы корпусов. Эта задача решается в общем виде при создании программатора на основе микросхем цифро-аналоговых преобразователей, таймеров и программируемых сильноточных коммутаторов. На первый взгляд это увеличивает его стоимость, но на самом деле эффективная стоимость (в расчете на каждое семейство программируемых микросхем) универсальных программаторов ниже, чем специализированных. Поэтому специализированные программаторы применяются в основном при параллельном тиражировании в условиях серийного производства.

Сложной является задача создания коммутирующего узла для работы с корпусами различных типов. Обычно она решается созданием переходных устройств типа DIP – PLCC, DIP – QFP и т.д. Фирма «Data I/O» в старших моделях программаторов применила универсальный коммутирующий узел на основе управляемой токопроводящей панели в комбинации с универсальной рамкой, осуществляющей позиционирование корпуса.

Требуется уточнения критерий «удобство интерфейса пользователя» по отношению к управляющей программе программатора. Многолетний опыт показывает, что разработчика, работающего с прибором ежедневно, раздражает обилие несистематизированной информации, помещенной в тесных (из-за полос прокрутки и панелей управления) окнах. Основным требованием к интерфейсу является тщательная методическая проработка отображаемой информации, уменьшение уровней вложенности меню. С другой стороны, важную роль играет наличие в документации информации об особенностях программирования микросхем различных серий.

Структура современных программаторов вполне определилась. Большинство из них реализуется либо в виде периферийного устройства персонального компьютера, подключаемого через один из внешних интерфейсных разъемов (RS-232, Centronix), либо в виде платы расширения, размещаемой внутри компьютера, с выносом коммутирующего узла посредством гибкого кабеля. Интерфейс с пользователем, загрузку рабочего файла, конфигурацию аппаратных средств программатора выполняет управляющая программа, функционирующая в среде операционной системы инструментального компьютера.

В качестве примера реализации функций универсального программатора рассмотрим прибор LabTools-48 фирмы Advantech. Он использует в качестве инструментального компьютера типа PC, к которому подсоединяется

через порт Centronix. Прибор позволяет программировать более 1200 разновидностей микросхем всех основных типов. Конфигурацию аппаратных средств выполняет управляющая программа, ее текущая версия доступна через BBS и может быть самостоятельно установлена пользователем. Аппаратура программатора создана на основе четырех микросхем и универсальных программируемых MOSFET коммутаторов. Это позволяет подать на любой вывод программируемой микросхемы одно из четырех указанных напряжений с 8-разрядным разрешением, либо логические уровни ТТЛ «0»/«1»/«highZ», либо синхросигнал. В структуру программатора включены микропроцессор и большая матрица буферной памяти. Они дают возможность быстро программировать микросхемы памяти большого размера (8 Мбит EPROM за 100 секунд) и сделать скорость программирования слабо зависящей от производительности инструментального компьютера. Программатор способен выполнять верификацию запрограммированной микросхемы при напряжениях питания 5 В, $5 \text{ В} \pm 5\%$, $5 \text{ В} \pm 10\%$. Все микросхемы в DIP корпусах вставляются в единственную универсальную 48-контактную розетку. Коммутация корпусов типа PLCC, SOP, TSOP, QFP и SDIP осуществляется с использованием переходных узлов (стоимостью \$100-200 за каждый). Управляющая программа способна функционировать в среде MS DOS, а также под Windows в режиме эмуляции DOS. Читаются входные файлы следующих типов JEDEC, POF, Binary, Intel HEX, Intel EXT HEX, Motorola S, HP64000ABS. Платой за универсальность является довольно высокая для отечественного рынка цена (\$1250 в базовом варианте без переходных узлов).

Программатор KROM (рис. 6.13), разработанный в лаборатории «Микропроцессорные системы» МИФИ, сочетает качества универсального профессионального прибора с приемлемой ценой.

Компромисс достигнут за счет структурных решений, все перечисленные выше важнейшие требования выполняются. Временные диаграммы программирования формируются с использованием трех микросхем ЦАП. Кроме возможности программировать широкую номенклатуру микросхем памяти и ПЛИС, это дало возможность реализовать специальный режим программирования (Special), в котором напряжение программирования задается с клавиатуры управляющего компьютера с шагом 0,25 В.

Вместо дорогостоящих программируемых коммутаторов использована коммутация с использованием пассивных катриджей в виде небольших сменных печатных плат. На основе предварительного анализа были выделены группы программируемых микросхем, имеющих аналогичные функции выводов. Для каждой группы разработан свой катридж, осуществляющий коммутацию исходного набора сигналов программирования на выводы универсальной розетки.

Конструктивно программатор состоит из двух частей: платы ЦАП, вставляемой в слот инструментального компьютера, и двух плат силовых

ключей, заключенных в типовой корпус фирмы «Vorla». На корпусе установлена универсальная розетка, в которую устанавливаются все программируемые микросхемы в DIP корпусах. Плата расширения и выносной блок связаны кабелем, позволяющим осуществлять параллельный обмен, что весьма важно при больших объемах программируемых данных и контрольной информации. Надежность программатора обеспечивается специальными схемами защиты, внутренним преобразователем напряжения, высококачественной импортной розеткой (нулевое усилие, рычаг зажима, позолоченные контакты).



Рис. 6.16 Универсальный профессиональный программатор KROM

Программатор KROM в настоящее время обеспечивает программирование следующих типов микросхем:

- микроконтроллеры Intel 8748/49, 87C51-87C58; 87C51Fx; Dallas DS87C520; Atmel AT89C51-55, AT89C1051-4051; AT89S8252/S53; AT90Sxx; PIC16Cxx, PIC16Fxx; Motorola MC68HC705P9/P6, 87C196Kx; Z86xx; K1816BE48,49;

- микросхемы с ультрафиолетовым стиранием 2716-27040, K573PФ2-6,8,10, KC1626PФ1,
- микросхемы с электрическим репрограммированием 28C04-28C040, 24Cxx, 25Cxx, 85Cxx, 93Cxx, K558PP2,3, K1801PP1, AT17Cxx,
- flash-память 28F256-28F040, 29F512-29F040, 29C256-040, 29EE512-020, 49F010-040, 49F001-002,
- память, программируемая пережиганием перемычек 556PT4-18, PT161, 1623PT1,2, K541PT2, K155PE3, N82Sxx, N74Sxx,
- ПЛИС 556PT1/2, PAL16xx, ATF16V8/20V8/22V10, AT22V10, ATV750, PALCE16V8/20V8,

Этот список постоянно расширяется. Кроме того, имеется возможность тестировать некоторые типы 8-разрядных микросхем ОЗУ. Удобство работы с программатором обеспечивают следующие сервисные функции:

- редактирование буфера данных, в том числе заполнение буфера константой, маскирование информации в буфере (функции AND, OR, XOR),
- перестановка столбцов в массиве,
- подсчет контрольной суммы содержимого буфера или ПЗУ,
- верификация/просмотр содержимого ПЗУ,
- представление функций выходов ПЗУ в псевдографической форме,
- сравнение содержимого буфера и ПЗУ,
- анализ возможности допрограммирования пережигаемого ПЗУ,
- чтение/запись файлов в форматах HEX, BIN, DMP, JED, S19,
- программный интерфейс с пакетом проектирования ABEL,
- возможность контроля ПЗУ при напряжении питания 4,5 5,5В

Программатор работает под управлением программы, функционирующей в среде MS DOS. Управление построено на основе меню команд, каждой позиции которого соответствует свой экран. Меню включает следующие команды:

Type	- выбор типа ПЗУ,
Disk	- чтение и запись файлов пользователя,
Buffer	- просмотр и редактирование буфера данных,
Verify	- верификация/просмотр содержимого ПЗУ,
Prog	- программирование ПЗУ/тестирование ОЗУ,
Compare	- сравнение содержимого буфера и ПЗУ,
Analysis	- анализ возможности допрограммирования ПЗУ,
Special	- специальные режимы и установки

Командой **Type** пользователь обычно начинает свою работу, выбирая тип программируемой микросхемы. Каждому типу микросхемы соответствует файл конфигурации аппаратуры программатора, который иницируется

при выборе и определяет функции выводов корпуса, временные диаграммы программирования, наличие битов защиты и т.д.

Команда **Disk** дает возможность просмотреть всю файловую систему инструментального компьютера и загрузить в буфер файл с кодом для программирования микросхемы. Файл может быть одного из следующих типов: hex, bin, dmp, jed, s19. Код для программирования может быть считан и из эталонной микросхемы.

По команде **Buffer** на экране постранично отображается содержимое памяти в шестнадцатеричном формате, курсор указывает на ячейку с адресом, который просматривался при последнем входе в данный режим. Основная часть работы пользователя сводится к редактированию информации в буфере.

В зависимости от выбранного типа ПЗУ обрабатываются 4-, 8- или 16-разрядные данные, для отображения которых используется 1, 2 и 4 знака, соответственно. Возможно непосредственное редактирование всей информации в буфере. При редактировании буфера данных с помощью клавиш F1-F10 могут быть активизированы дополнительные сервисные функции. Имена функций и назначение клавиш в процессе работы индицируется в окне меню.

F1 – Split Обработка массива при работе с расслоенной (interleaved) памятью. Эта функция используется при подготовке данных для 16-разрядного микропроцессора с памятью, организованной в виде двух банков по младшим и старшим байтам слова (например, 80x86).

F2 – Addr Переход к произвольному адресу для редактирования. Значение адреса вводится в окне, появляющемся при активизации функции.

F3 – Fill Заполнение кодом (константой) ячеек памяти в заданном диапазоне адресов.

F4 – Mask Маскирование фрагмента буфера. Над всеми байтами фрагмента и маской побитно выполняется одна из логических функций AND, OR, XOR, INV (инверсия адресов и данных).

F5 – Bits Перестановка битов во фрагменте буфера. Во всех байтах массива производится перестановка битов по заданному шаблону. Допускается многократное копирование бита в пределах байта. Предполагается, что перед выполнением операции биты расположены в порядке 7 6 5 4 3 2 1 0 (шаблон – 76543210).

F6 – Seek Поиск заданной последовательности байтов в буфере. Для выполнения операции необходимо в окне «Модель для поиска» ввести кодовую последовательность.

F7 – Copy Копирование заданного фрагмента в буфере на область адресов, начинающуюся с заданного адреса (смещение). В окне параметров необходимо задать начальный и конечный адреса фрагмента и адрес смещения.

F8 – Read Чтение заданного фрагмента из ПЗУ в буфер, начиная с заданного адреса (смещение) В окне параметров необходимо задать начальный и конечный адреса фрагмента в ПЗУ и адрес смещения в буфере

F9 – Print Вывод содержимого фрагмента буфера на печать

F10 – Sum Подсчет контрольной суммы В окне параметров задаются начальный и конечный адреса в буфере Вычисления производятся одновременно тремя способами суммирование байтов без учета переполнения, исключающее ИЛИ всех байтов в диапазоне адресов, по методу РК86, что означает вычисление 16-разрядной контрольной суммы, где младший байт вычисляется по методу 1, а старший байт равен сумме байтов с учетом переполнения, возникающего при вычислении текущего значения младшего байта (последний байт не учитывается при вычислении значения старшего байта контрольной суммы)

Весьма полезен и нагляден режим развернутого просмотра содержимого буфера в виде логических диаграмм (клавиши ALT-F1, функция Spread) При этом данные каждой ячейки представляются в кодах ASCII, десятичном, шестнадцатеричном и двоичном формате, а также в виде двоичного уровня на каждом из выходов

Команда **Verify** позволяет быстро определить, запрограммирована БИС ПЗУ или нет Если во всех ячейках микросхемы записан исходный код (0 или FFH в зависимости от типа ПЗУ), то выдается сообщение «ПЗУ не запрограммировано» Если хотя бы в одной ячейке записан код, то программа переходит к режиму просмотра ПЗУ с записью кодов в буфер и отображением содержимого буфера Отображение на экране аналогично тому, что имеет место при работе в режиме Buffer Возможен вывод фрагмента содержимого ПЗУ на печать, подсчет контрольной суммы ПЗУ, а также просмотр буфера в развернутом виде (ALT-F1)

Для ПЗУ, имеющих встроенный идентификатор типа (Intelligent Identifier, Signature Bytes), при выполнении команды Verify производится считывание этого идентификатора Прочитанная информация отображается в правой части строки состояния

Если при включении питания ток, потребляемый микросхемой, превышает 400 мА (900 мА для микросхем серии K556), то управляющая программа снимает напряжение питания и выдает сообщение «Большой ток нагрузки»

Команда **Prog** запускает процесс электрической записи информации в микросхему ПЗУ или тестирование ОЗУ В окне параметров устанавливаются следующие величины начальный и конечный адреса фрагмента в буфере, адрес смещения в ПЗУ, напряжение программирования Установка напряжения программирования производится с точностью 0,25 В (не для всех типов ПЗУ) Максимальная величина устанавливаемого напряжения равна 25,5 В Этот параметр можно изменять лишь в особых случаях Для ряда типов ПЗУ можно выбрать алгоритм записи – стандартный или быст-

рый Стандартный алгоритм соответствует рекомендованному для данной микросхемы по ТУ. При использовании быстрого алгоритма время записи информации в каждую ячейку устанавливается индивидуально, в зависимости от результата проверки. Например, быстрый алгоритм для микросхем серии K573 – это алгоритм Intelligent, рекомендованный фирмой Intel. Для микросхем серии 27xx фирмы Intel предусмотрены алгоритмы Intelligent и Quick Pulse.

Для ряда микросхем можно установить опцию контроля содержимого ПЗУ до и после программирования («Проверка»). Если проверка выключена, то запись производится в каждый байт микросхемы, отличающийся от незапрограммированного состояния, независимо от содержащейся в ПЗУ информации.

Тестирование ОЗУ осуществляется в два этапа. Сначала буфер заполняется тестовой информацией и содержимое заданного диапазона записывается в ОЗУ. Затем производится считывание данных из ОЗУ и сравнение с буфером. Информация о несовпадении данных отображается так же, как в режиме Compare. При тестировании ОЗУ записываемые данные извлекаются из буфера. Если при выполнении операции буфер чист (заполнен константой), то в нем автоматически формируется тест по правилу: содержимое ячейки равно инвертированной сумме старшего и младшего байтов адреса ячейки.

Команда **Compare** сравнивает содержимое буфера и ПЗУ. В этом режиме необходимо в окне параметров указать диапазон адресов в буфере, адрес смещения в ПЗУ (по умолчанию те же, что и в режиме программирования), а также величину напряжения питания. При этом предлагаются следующие опции: + 5В – операция выполняется при номинальном напряжении, + 4,5 В – операция выполняется при напряжении $V_{CC} - 10\%$, + 5,5 В – операция выполняется при напряжении $V_{CC} + 10\%$, Auto – операция выполняется последовательно при всех трех напряжениях.

Если информация идентична, то появляется сообщение «Массивы совпадают», если нет – программа переходит в режим просмотра несовпадающих ячеек массивов и адреса несовпадающих ячеек фиксируются во временном буфере. Если количество таких ячеек больше 1024, то анализ прекращается. При просмотре несовпадений на экране отображаются только данные, находящиеся в ячейках по адресам, зафиксированным во временном буфере.

Команда **Analysis** производит анализ возможности допрограммирования микросхемы ПЗУ. В этом режиме производится побайтное сравнение массивов данных в буфере и в ПЗУ. Программа определяет, возможна ли запись массива из буфера в ПЗУ без искажения информации. В зависимости от типа ПЗУ можно дописывать в каждый бит «0» или «1». Результат анализа выдается в виде сообщения «Допрограммирование возможно», в противном случае программа переходит в режим просмотра ячеек, не удов-

летворяющих условиям допрограммирования, аналогично предыдущему пункту

Команда **Special** позволяет использовать специальные установки и режимы. С ее помощью можно установить величины напряжений, используемых при чтении/записи информации в ПЗУ. Она также активизирует специальные режимы записи, если они предусмотрены для данной микросхемы. Управляющая программа поддерживает следующие специальные режимы:

- стирание информации в электрически репрограммируемых микросхемах КР558РР2/РР3/РР4, К1801РР1, 28Fххх, АТ89С51, АТ89С2051,
- программирование кода ПЗУ – для К573РФ3, К1801РР1,
- программирование битов защиты и таблицы кодирования – для 87х51хх, 89Схх, 85С22х, РAL16хх

Программируются все источники питания, используемые при работе с программируемыми схемами, которые выдают напряжения, отличные от значения +5 В. Таких источников два типа: напряжения питания при программировании (устанавливается для микросхем серии 27хх) и напряжение программирования (дублирует установку, описанную в команде Prog, которая является паспортной величиной микросхемы). Установка напряжений программирования требует осторожности, чтобы не повредить буферные каскады микросхем.

Программатор снабжен подробной документацией. Значительный объем в ней занимает информация об особенностях программирования микросхем различных серий.

6.5. Учебные практикумы микропроцессорной техники и ПЛИС

Несмотря на публикации в периодической печати, издание нескольких монографий [1, 2], в области микропроцессорной техники ощущается значительный неудовлетворенный спрос на средства и методы обучения. При создании рассматриваемых далее учебных практикумов мы исходили из следующих соображений:

- Учебные методики и средства должны разрабатывать специалисты, имеющие большой опыт разработки систем проектирования и целевых систем управления.
- В качестве учебных средств должны использоваться версии профессиональных систем проектирования, поскольку использование чисто учебных средств искажает методику проектирования и приводит к необходимости последующего переучивания.

- Ввиду большого объема информации по элементной базе и сложности систем проектирования основным принципом лабораторного практикума должен быть «делай как я», т.е. в каждой работе должно быть обязательное задание, процесс выполнения которого подробно описан. Это позволит не только передать знания и практические навыки, но и привить правильный стиль проектирования,

На основе этих принципов в лаборатории «Микропроцессорные системы» МИФИ созданы учебные средства, позволяющие в короткий срок передать практические навыки разработки микропроцессорных систем управления на основе микроконтроллеров и ПЛИС. Класс «Проектирование систем на микроконтроллерах и ПЛИС» в настоящее время позволяет изучать

- 8-разрядные микроконтроллеры MCS-51 с традиционной (CISC) архитектурой,
- RISC-микроконтроллеры AVR фирмы Atmel,
- проектирование цифровых устройств с использованием ПЛИС фирмы Altera и САПР Max+PlusII

В состав класса включены аппаратные средства типа одноплатных контроллеров, программные средства в виде интегрированных систем программирования микроконтроллеров и систем автоматизированного проектирования цифровых устройств на ПЛИС, методическое обеспечение в виде лабораторных практикумов, универсальный программатор

Главным звеном учебного класса являются методические материалы лабораторных практикумов. Каждый из них состоит из введения, в котором описана архитектура рассматриваемого семейства БИС и методика проектирования, набора лабораторных работ и приложений, в которых приведены технические характеристики плат и элементной базы. Перечень лабораторных работ практикума по микроконтроллерам AVR фирмы Atmel имеет следующий вид

“ПРОЕКТИРОВАНИЕ СИСТЕМ НА МИКРОКОНТРОЛЛЕРАХ AVR mega103”

ОГЛАВЛЕНИЕ

- Введение
- Работа 1 Методы адресации, команды передачи данных и управления
- Работа 2 Команды обработки данных
- Работа 3 Контроль внешних устройств через параллельные порты
 работа с клавиатурой
- Работа 4 Реализация и обслуживание подсистемы прерываний
- Работа 5 Реализация таймерных функций
- Работа 6 Организация последовательного обмена данными между
 контроллерами
- Работа 7 Обслуживание АЦП
- Приложение 1 Интегрированная система программирования «ТурбоАссемблер AVR»
- Приложение 2 Учебный контроллер AVR mega103

Описание каждой лабораторной работы содержит информационно-теоретическую часть, обязательное задание (включая фрагмент текста программы с комментариями и разбором реализации алгоритма) и варианты заданий для самостоятельного выполнения. Например, в работе 4 практикума по микроконтроллерам AVR описаны механизмы реализации прерываний, регистры управления микроконтроллеров. Затем приведен ассемблерный текст примера обязательного задания, формирующий таблицу переходов и инициализирующий указатель стека, разрешающий определенные прерывания и определяющий их приоритеты. Далее следуют процедуры обслуживания прерываний и программные заглушки вместо процедур замаскированных прерываний. Завершается работа набором самостоятельных заданий, которые выполняются посредством изменения текста разобранного задания.

Обязательные задания практикума нарастают по сложности и в последующих работах используются программные модули из предыдущих работ.

Структура практикума позволяет без привлечения дополнительной литературы подготовиться как преподавателю, так и студенту или слушателю курсов повышения квалификации. Имеется и опыт самостоятельного изучения предмета с его помощью.

Лабораторный макет практикума по микроконтроллерам AVR фирмы Atmel имеет типичную для наших учебных контроллеров структуру, которая включает ядро (в данном случае на основе микроконтроллера ATmega103), модули последовательных интерфейсов SPI и RS232, стабилизатор питания, схему рестарта и синхронизации, 12-клавишную клавиатуру, блок семисегментных индикаторов и пьезодинамик. Плата выполнена с использованием элементов технологии поверхностного монтажа и демонстрирует современный подход к проектированию микропроцессорных контроллеров на основе функционально-топологических модулей.

Программирование микропроцессорных систем изучается с использованием интегрированных систем, для рассматриваемого практикума это «ТурбоАссемблер-AVR». Система относится к классу профессиональных, объединяет редактор текста, транслятор, библиотеки периода выполнения, символический отладчик. Наличие программно-логической модели в составе системы позволяет использовать ее в первых работах без платы контроллера, наблюдая состояние регистров на мониторе компьютера. Транслятор системы анализирует текст программы и выдает сообщения об обнаруженных ошибках, а наличие команды Help позволяет получить оперативную информацию о командах микроконтроллера, директивах ассемблера, библиотеках и т.д. Именно развитость инструментальной системы программирования и большой объем сервисных функций позволяет быстро освоить систему команд и разработку прикладных программ. Кроме того, в современных условиях важно, что после обучения специалист сразу может раз-

рабатывать изделия с использованием освоенных инструментальных средств, без закупок новой техники и переучивания

Лабораторный практикум по проектированию систем на микроконтроллерах MCS-51 в целом аналогичен практикуму по микроконтроллерам AVR

Лабораторный практикум «Проектирование цифровых устройств на ПЛИС» описан в параграфе 5.7