Национальный технический университет Украины

«Киевский Политехнический Институт»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

Расчетно-графическая работа

по дисциплине

«Архитектура компьютера»

Выполнила:

Ст. Смирнова Анастасия Михайловна

Группа ІО-81

Факультет ИВТ  
Зачетная книжка № 8117

Номер техническогозадания 1111110110101

Допущенакзащите\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Киев 2010 г.

Описание альбома

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *N п.п.* | *Формат* | *Обозначение* | | | *Наименование* | *К. листов* | | *№ экз.* | | *Примечание* | | |
|  |  |  | | | Документацияобщая |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | | Разработана заново |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  | А4 | ИАЛЦ 463617.001 ОА | | | Микропроцессорная система | 1 | |  | |  | | |
|  |  |  | | | *Описание альбома* |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  | А4 | ИАЛЦ 463617002 ТЗ | | | Микропроцессорная система | 1 | |  | |  | | |
|  |  |  | | | *Техническое задание* |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  | А3 | ИАЛЦ 463617.004 Е2 | | | Микропроцессорная система | 1 | |  | |  | | |
|  |  |  | | | *Схема структурная* |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  | А4 | ИАЛЦ 463617.003ПЗ | | | Пояснительная записка | 23 | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  |  |  | *ИАЛЦ 463617.001 ОА* | | | | | | | |
|  |  |  |  |  |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| *Выполнил* | | *Смирнова А.М.* |  |  | Микропроцессорная система  *Описание альбома* | | *Лит.* | | | | *Лист* | *Листов* |
| *Проверил* | | *Ткаченко В.В.* |  |  |  | |  |  | *1* | *1* |
|  | |  |  |  | *НТУУ «КПИ» ФИВТ*  *Гр. ИВ-81* | | | | | |
| *Н. контр.* | |  |  |  |
| *Зав. каф.* | |  |  |  |

Техническоезадание

Техническое задание на расчетную работу

Проектирование микропроцессорных систем

студенткиСмирновой А. М.

группыИВ-81

№ зачетной книжки

|  |  |
| --- | --- |
| Выбор элементной базы | 1816ВЕ51 |
| Организация шины | С объединенными шинами адреса и данных |
| Выбор системы команд | Комплексная |
| КПП, КПДП | Централизованный |
| Способ умножения, деления, разрядность операндов | Умножение – байтовое  Деление – байтовое  Разрядность – 16(деление – 8) |
| КоличествоВУ | 116 |
| Функция |  |
| Вспомогательные порти, периферийные адаптеры | ВВ55, Р4,Р7 |
| Адреса для интерфейса внешнего устройства | Самостоятельно |
| Адреса для ППА | 98h, 99h, 9Ah, 9Bh |
| Внешняя память данных | по 64К(10 страниц) |
| Внешняя память программ | по 64К(2 страницы) |
| Структурна схема | МПС |

Задание выдал

Задание принял

Ткаченко В.В.

Пояснительная записка

СОДЕРЖАНИЕ

[ВСТУПЛЕНИЕ 2](#_Toc280296568)

[1. Обзор микроконтроллера 3](#_Toc280296569)

[2. Обзор микропроцессорной системы 3](#_Toc280296570)

[2.1 Описание архитектуры микропроцессорной системы 3](#_Toc280296573)

[2.2 Память программ 4](#_Toc280296574)

[2.3 Память данных 4](#_Toc280296575)

[2.4 Внешние устройства и ППА 6](#_Toc280296576)

[2.5 Режим прерываний 8](#_Toc280296577)

[2.6 Прямой доступ к памяти 10](#_Toc280296578)

[2.7 Подключение дополнительных портов 12](#_Toc280296579)

[2.8 Система команд 13](#_Toc280296580)

[2.8.1 Основные команды 13](#_Toc280296581)

[2.8.2 Арифметические и логические команды 13](#_Toc280296582)

[2.8.3 Команды ветвления и передачи управления 14](#_Toc280296583)

[3. Структурная схема МПС 15](#_Toc280296584)

[4. Программная часть 16](#_Toc280296585)

[4.1 Подпрограмма умножения 16](#_Toc280296586)

[4.1.1 Блок-схема алгоритма 16](#_Toc280296587)

[4.2 Подпрограмма деления 17](#_Toc280296588)

[4.2.1 Блок-схема алгоритма 17](#_Toc280296589)

[4.3 Блок-схема алгоритма вычисления основной функции 18](#_Toc280296590)

[4.4 Листинг программы 19](#_Toc280296591)

[ВЫВОДЫ 23](#_Toc280296592)

[ЛИТЕРАТУРА 23](#_Toc280296593)

# ВСТУПЛЕНИЕ

Микроконтроллер — микросхема, предназначенная для управления электронными устройствами. Типичный микроконтроллер сочетает в себе функции процессора и периферийных устройств, может содержать ОЗУ и ПЗУ. Большая часть выпускаемых в современном мире процессоров — микроконтроллеры.

Основным классификационным признаком микроконтроллеров является разрядность данных, обрабатываемых арифметико-логическим устройством (АЛУ). По этому признаку они делятся на 4-, 8-, 16-, 32- и 64-разрядные.

При проектировании микроконтроллеров приходится соблюдать баланс между размерами и стоимостью с одной стороны и гибкостью и производительностью с другой. Для разных приложений оптимальное соотношение этих и других параметров может различаться очень сильно. Поэтому существует огромное количество типов микроконтроллеров, отличающихся архитектурой процессорного модуля, размером и типом встроенной памяти, набором периферийных устройств, типом корпуса и т. д.

Неполный список периферии, которая может присутствовать в микроконтроллерах, включает в себя:

* универсальные цифровые порты
* различные интерфейсы ввода-вывода
* аналого-цифровые и цифро-аналоговые преобразователи
* широтно-импульсные модуляторы
* таймеры, встроенный тактовый генератор и сторожевой таймер
* контроллеры без коллекторных двигателей
* контроллеры дисплеев и клавиатур
* радиочастотные приемники и передатчики
* массивы встроенной флэш-памяти

Программирование микроконтроллеров обычно осуществляется на языке ассемблера или Си. Для отладки программ используются программные симуляторы (специальные программы для персональных компьютеров, имитирующие работу микроконтроллера), внутрисхемные эмуляторы и интерфейс JTAG.

# Обзор микроконтроллера

Intel 8051 — это однокристальный микроконтроллер (не путать с процессором) гарвардской архитектуры, который был впервые произведен Intel в 1980 году, для использования во встраиваемых системах. В течение 1980-ых и начале 1990-ых годов был чрезвычайно популярен. Однако в настоящее время устарел и вытеснен более современными устройствами. Существует также советский клон данной микросхемы, КР1816ВЕ51. Официальное название 8051-семейства микроконтроллеров Intel — MCS 51.

* Состоит из процессорного ядра (CPU), ОЗУ, ПЗУ, последовательного порта, параллельного порта, логики управления прерываниями, таймер и т. д.
* Шина данных — 8-ми битная шина данных. Возможность обработки 8 бит данных за одну операцию. Обуславливает название 8-битный микропроцессор.
* Шина адреса — 16 битная адресная шина. Возможность доступа к 216 адре­сам памяти, то есть 64 Кбайт адресное пространство в ОЗУ и ПЗУ.
* Встроенное ОЗУ — 128 байт (Памяти данных).
* Встроенное ПЗУ — 4 КБ (Памяти программ).
* Четыре порта ввода/вывода: один двунаправленный и три квазидвунаправ­ленных.
* Последовательный интерфейс UART
* Два 16-битных таймера.
* Два уровня приоритета прерываний.
* Энергосберегающий режим.

Чрезвычайно полезной особенностью 8051 ядра является обработка булевых данных, что позволило ввести бинарную логику, оперирующую напрямую с битами внутренней ОЗУ (области из 128 прямо-адресуемых битов) и регистров.

# Обзор микропроцессорной системы



## Описаниеархитектуры микропроцессорной системы

Разрабатываемаямикропроцессорная система состоит из нескольких микросхем, в состав которых входит микроконтроллер КР1816ВЕ51. В состав МПС должны входить:

* Процессор
* Основнаяпамять (ОЗУи ПЗУ)
* Внешние устройства
* Контролер прерываний
* Контролер прямого доступа к памяти

Объем внешней памяти программ – 2 страницы по 64К. Объем внешней памяти данных – 10 страниц по 64К. Шина адресаиданных объединена. КПП и КПДП централизованы. Каждое ВУимеет фиксированный адрес в адресном пространствепериферийныхустройств. Количество внешних устройств – 116. Система команд – комплексная.

## Память программ

Память программ предназначена для хранения программ и имеет отдельное от памяти данных адресное пространство объемом 64 Кбайт, причем для хранения программ на кристалле микроконтроллера расположено ПЗУ. Это ПЗУ отображается в область младших адресов памяти программ. Микроконтроллеры семейства MCS-51 имеют внешний вывод EA, с помощью которого можно запретить работу внутренней памяти, для чего необходимо подать на вывод EA логический "0". При этом внутренняя память программ отключается и, начиная с нулевого адреса, все обращения происходят к внешней памяти программ.

 Доступ к внешней памяти программ осуществляется в двух случаях:

1. при действии сигнала EA=0 независимо от адреса обращения,
2. в любом случае, если программный счетчик (РС) содержит число большее, чем максимальная ячейка внутренней памяти программ.

## Память данных

Внешняя память данных предназначена для временного хранения информации, используемой в процессе выполнения программы. Максимальный объем этой памяти определяется регистром DPTR и составляет 64 Кбайта. Точно также как и в случае внешней памяти программ, объем внешней памяти данных может быть увеличен за счет использования портов P1 и P3 до 1 Гбайта. Внешняя память данных для своей работы требует использование портов P0, P2 и P3. Это приводит к увеличению габаритов устройства, увеличению уровня помех и, в конечном итоге, увеличения стоимости устройства в целом.

Для обращения к внешней памяти данных используются команды

MOVX A, @DPTR (команда чтения) и

MOVX @DPTR, A (команда записи)

Отметим, что в качестве внешней памяти данных могут быть использованы как микросхемы ОЗУ, так и микросхемы ПЗУ.

Распределение памяти данных микроконтроллеров серии MCS-51 приведено на рисунке 2.1.

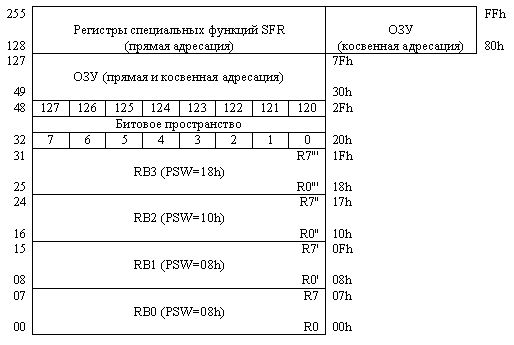


Рисунок 2.1. Адресное пространство внутренней памяти данных

Внутреннее ОЗУ данныхпредназначено для временного хранения информации, используемой в процессе выполнения программы, и занимает 128 младших байт, с адресами от 000h до 07Fh.

Система команд микроконтроллера позволяет обращаться к ячейкам внутренней памяти данных при помощи прямой и косвенно-регистровой адресации. При обращении к ячейкам памяти с адресами 0-127 использование любого из этих видов адресации будет производить выборку одной и той же ячейки памяти. При обращении к ячейкам ОЗУ с адресами 128-256 следует воспользоваться косвенно-регистровой адресацией. Учитывая, что работа со стеком ведётся при помощи косвенной адресации, то имеет смысл размещать в этой области памяти стек. Если же требуется обратиться к регистрам специальных функций, то нужно использовать прямую адресацию.

Регистры общего назначения позволяют писать самые эффективные программы. У микроконтроллеров семейства MCS-51 доступны восемь регистров. Более того, в этом семействе микроконтроллеров есть четыре банка регистров с именами RB0 - RB3. Банк регистров состоит из восьми восьмиразрядных регистров с именами R0,…, R7.  Переключение банков регистров производится при помощи двух особых бит регистра слова состояния программы PSW (RS0 и RS1).

Следующие после банков регистров внутреннего ОЗУ данных 16 ячеек памяти (адреса 20Н-2FH) образуют область памяти, к которой возможна как байтовая, так и битовая адресация. В этих ячейках располагаются 128 программных флагов (битовых ячеек памяти). Обращение к отдельным битам этих ячеек возможно по их битовым адресам. Оставшаяся область памяти используется как обычное ОЗУ без особенностей.

## Внешние устройства и ППА

В данном режиме инициатором обмена является процессор. Для синхронизации используется бит готовности в порте ВУ. Этот бит устанавливается контроллером ВУ, когда оно готово к обмену, и сбрасывается при обращении к порту данных.

При наличии нескольких ВУ для их обслуживания используется программный поллинг, т.е. опрос ВУ в соответствии с их приоритетом. Режим опроса готовности имеет преимущества и недостатки по сравнению с другими режимами.

Преимущества: простота реализации интерфейса ВУ, в процессе функционирования программы можно менять приоритеты ВУ. Недостатки: снижение производительности за счет непроизводительного расхода команд процессора на опрос ВУ, трудно предусмотреть аварийное или экстренное обслуживание некоторого ВУ.

Основное назначение программируемых периферийных адаптеров разработка программируемых устройств ввода/вывода для МПС. Структурная схема программируемого периферийного адаптера К580ВВ55 приведена на рисунке 2.2.



Рисунок 2.2. Структурная схема ППА К580ВВ55

Адаптер 580ВВ55 обеспечивает ввод/вывод по трем дополнительным восьмиразрядным портам РА, РВ, РС. Причем порт PC может быть использован в качестве двух четырехразрядных портом PCh – старшая тетрада порта РС, PCl – младшая тетрада порта PC.

В состав ППА входят следующие функциональные блоки:

* Буфер шины данных D7-D0
* Схема управления чтением/записью данных в регистра ППА
* Группа А, порта РА – порт ввода/вывода РА группы А
* ГруппаВ, порт РВ – порт ввода/вывода РВ группы В
* Группа С, порт РВ – порт ввода/вывода РВ группы В
* Группа С, порт РС – порт ввода/вывода PCh группы А
* Группа В, порт РС – порт ввода/вывода PCl группы В

Схемы управления портами группы А и В содержат регистр управления, который задает режимы работы портов.

Все порты оснащены буферными регистрами, через которые производится связь между ППА и внешними шинами.

## Режим прерываний

Под прерыванием понимают временную приостановку выполнения программы и переход на другую подпрограмму с возможностью возврата на прерванную. Прерывания можно классифицировать следующим образом: внутренние и внешние. Внутренние делятся на аппаратные и программные.

Микроконтроллеры семейства МК-51 обеспечивают поддержку пяти источников прерываний:двух внешних прерываний, поступающих по входам *INT0* и *INT1* (линии порта Р3: Р3.2 и Р3.3 соответственно);двух прерываний от таймеров/счетчиков*Т/С0* и *Т/С1*;прерывание от последовательного порта.

Запросы на прерывание фиксируются в регистрах специальных функций микроконтроллера: флаги *IE0, IE1, TF0, TF1* запросов на прерывание от *INT0, INT1, T/C0* и *T/C1* соответственно содержатся в регистре управления *TCON*, а флаги *RI* и *TI* запросов на прерыва­ние от последовательного порта - в регистре *SCON* управления последовательным портом (см. рис.4).

Флаги *TF0* и *TF1* устанавливаются аппаратно при переполнении соответствующего таймера/счетчика (переход *T/Cx* из состояния “все единицы” в состоянии “все нули”).

Флаги *IE0* и *IE1* устанавливаются аппаратно от внешних прерываний *IT0* и *IT1* соответственно. Установка *ITx*=0 настраивает систему прерывания на запрос по низкому уровню сигнала, *ITx*=1 - запрос на прерывание по спаду сигнала.

Флаги *TI* и *RI* устанавливаются аппаратно схемой последовательного интерфейса соответственно после окончания передачи или приема байта

Все указанные флаги запросов на прерывания программно доступны для установки и сброса. Программная установка флага запроса на прерывание приводит к такой же реакции микроконтроллера, что и аппаратная установка того же самого флага.

Флаги *TF0* и *TF1* сбрасываются аппаратно при передаче управления программе обработки соответствующего прерывания.

Сброс флагов *IЕ0* и *IЕ1* выполняется аппаратно при обслуживании прерывания только в том случае, если прерывание было настроено на восприятие спада сигнала *INTx*. Если прерывание было настроено на восприятие уровня сигнала запроса, то сброс флага *IEx* должна выполнять программа обслуживания прерывания, воздействуя на источник прерывания для снятия им запроса.

Флаги *TI* и *RI* сбрасываются только программным путем.

Каждый вид прерывания индивидуально разрешается или запрещается установкой или сбросом соответствующих бит регистра разрешения прерывания *IE*. Этот регистр содержит также и бит общего запрещения всех прерываний.

При одновременном поступлении запросов прерывания от источников, имеющих различные приоритеты, сначала обрабатывается запрос от более приоритетного источника.

В случае одновременного поступления нескольких запросов на прерывания с одинаковым приоритетом порядок их обработки определяется аппаратными средствами микроконтроллера и не может быть изменен программно. Этот порядок соответствует последовательности опроса флагов запросов прерываний, имеющей следующий вид:*IT0 → TF0 → IT1 → TF1 → (RI,TI)*

Общая структура системы прерываний МК-51 представлена на рисунке 2.3.

Адрес

вектора

прерывания

EA

IT0

1

0

1

0

0

1

0

1

PT0

PX1

0

1

0

1

PT1

PS

23h

1Bh

13h

0Bh

03h

1

PX0

0

Порядок опроса

флагов при ра­вен­стве приоритетов

1

0 0

1

Регистр IE

ИндивидуальноеОбщее

разрешениепрерываний

Низкийуровень

приоритета

1

1

IE0

IE1

TF0

TF1

TI

1

INT0

INT1

IT1

EX0

Регистр IP

Высокийуровень

приоритета

1

0 0

1

ET0

1

0 0

1

EX1

1

0 0

1

ET1

1

0 0

1

ES

RI

Рисунок 2.3 Структура системы прерываний

При переходе на подпрограмму обработки прерывания автоматически независимо от состояния регистра *IE* запрещаются все прерывания, имеющие уровень приоритета, равный уровню приоритета об­служиваемого прерывания, то есть вложенные прерывания с равным уровнем приоритета запрещены. Таким образом, низкоприо­ри­тетное прерывание (имеющее “0” в соответствующем разряде регистра *IP*) может прерываться высокоприоритетным (имеющим “1” в соответствующем разряде регистра *IP*), но не низкоприоритетным. Обслуживание высокоприоритетного прерывания не может быть прервано другим источником.

Возврат из обработчика прерываний осуществляется с по­мощью команды *RETI*, которая восстанавливает из стека значение *PC* и логику приоритетов прерываний.

## Прямой доступ к памяти

Одним из способов обмена данными с ВУ является обмен в режиме прямого доступа к памяти (ПДП). В этом режиме обмен данными между ВУ и основной памятью микроЭВМ происходит без участия процессора. Обменом в режиме ПДП управляет не программа, выполняемая процессором, а электронные схемы, внешние по отношению к процессору. Обычно схемы, управляющие обменом в режиме ПДП, размещаются в специальном контроллере, который называется контроллером прямого доступа к памяти.

Обмен данными в режиме ПДП позволяет использовать в микроЭВМ быстродействующие внешние запоминающие устройства, такие, например, как накопители на жестких магнитных дисках, поскольку ПДП может обеспечить время обмена одним байтом данных между памятью и ВЗУ, равное циклу обращения к памяти.Для реализации режима прямого доступа к памяти необходимо обеспечить непосредственную связь контроллера ПДП и памяти микроЭВМ. В целях сокращения количества линий в шинах микроЭВМ контроллер ПДП подключается к памяти посредством шин адреса и данных системного интерфейса. При этом возникает проблема совместного использования шин системного интерфейса процессором и контроллером ПДП. Можно выделить два основных способа ее решения: реализация обмена в режиме ПДП с "захватом цикла" и в режиме ПДП с блокировкой процессора.

Существуют две разновидности прямого доступа к памяти с "захватом цикла". Наиболее простой способ организации ПДП состоит в том, что для обмена используются те машинные циклы процессора, в которых он не обменивается данными с памятью. В такие циклы контроллер ПДП может обмениваться данными с памятью, не мешая работе процессора. Однако возникает необходимость выделения таких циклов, чтобы не произошло временного перекрытия обмена ПДП с операциями обмена, инициируемыми процессором. В некоторых процессорах формируется специальный управляющий сигнал, указывающий циклы, в которых процессор не обращается к системному интерфейсу. Более распространенным является ПДП с "захватом цикла" и принудительным отключением процессора от шин системного интерфейса. Для реализации такого режима ПДП системный интерфейс микроЭВМ дополняется двумя линиями для передачи управляющих сигналов "Требование прямого доступа к памяти" (ТПДП) и "Предоставление прямого доступа к памяти" (ППДП).

Управляющий сигнал ТПДП формируется контроллером прямого доступа к памяти. Процессор, получив этот сигнал, приостанавливает выполнение очередной команды, не дожидаясь ее завершения, выдает на системный интерфейс управляющий сигнал ППДП и отключается от шин системного интерфейса. С этого момента все шины системного интерфейса управляются контроллером ПДП. Контроллер ПДП, используя шины системного интерфейса, осуществляет обмен одним байтом или словом данных с памятью микроЭВМ и затем, сняв сигнал ТПДП, возвращает управление системным интерфейсом процессору. Как только контроллер ПДП будет готов к обмену следующим байтом, он вновь "захватывает" цикл процессора и т.д. В промежутках между сигналами ТПДП процессор продолжает выполнять команды программы. Тем самым выполнение программы замедляется, но в меньшей степени, чем при обмене в режиме прерываний.

Применение в микроЭВМ обмена данными с ВУ в режиме ПДП всегда требует предварительной подготовки, а именно: для каждого ВУ необходимо выделить область памяти, используемую при обмене, и указать ее размер, т.е. количество записываемых в память или читаемых из памяти байт (слов) информации. Следовательно, контроллер ПДП должен обязательно иметь в своем составе регистр адреса и счетчик байт (слов). Перед началом обмена с ВУ в режиме ПДП процессор должен выполнить программу загрузки. Эта программа обеспечивает запись в указанные регистры контроллера ПДП начального адреса выделенной ВУ памяти и ее размера в байтах или словах в зависимости от того, какими порциями информации ведется обмен. Сказанное не относится к начальной загрузке программ в память в режиме ПДП. В этом случае содержимое регистра адреса и счетчика байт слов устанавливается переключателями или перемычками непосредственно на плате контроллера.

Использование БИС ПДП позволяет существенно сократить аппаратные затраты при реализации прямого доступа к памяти.



Рисунок 2.4. Реализация централизованного КПДП

## Подключение дополнительных портов

Для увеличения количества линий связи МК с объектом управления можно подключить дополнительные 4-разрядные порты P4-P7. Наиболее просто это достигается при использовании специальной ИС КР580 ВР43. В этом случае обеспечивается выполнение всех четырех команд с дополнительными портами, причем каждый вывод порта может быть настроен на ввод или вывод информации. Команды выполняются за 2 цикла. В первом цикле на выводы P4-P7 выдается управляющее слово, а во втором – через указанные выводы осуществляется обмен информацией между МК и портом.

## Система команд

### Основные команды

Система команд микроконтроллера предоставляет большие возможности обработки данных, обеспечивает реализацию логических, арифметических операций, а также управление в режиме реального времени.

В этой системе команд реализована побитная, потетрадная (4 бита), побайтовая (8 бит) и 16-разрядная обработка данных. Команды выполняются за один, два или четыре (умножение и деление) машинных цикла.

### Арифметические и логические команды

В наборе команд микроконтроллера имеются следующие арифметические операции:

* сложение ADD,
* сложение с учетом флага переноса ADDC,
* вычитание с займом SUBB,
* инкрементирование (увеличение на 1) INC,
* декрементирование (уменьшение на 1) DEC,
* десятичная коррекция DA,
* умножение MUL
* деление DIV

Действия производятся над целыми числами без знака.При операции умножения содержимое аккумулятора A умножается на содержимое регистра B, и результат размещается следующим образом: младший байт в регистре B, старший - в регистре А.В случае выполнения операции деления целое от деления помещается в аккумулятор A, остаток - в регистр В.

Система команд рассматриваемого микроконтроллера позволяет реализовать логические операции:

* И (ANL),
* ИЛИ (ORL),
* ИСКЛЮЧАЮЩЕЕ ИЛИ (XRL).

Логические операции выполняются над аккумулятором или непосредственно над портами ввода/вывода.

Существуют логические операции, которые выполняются только на аккумуляторе:

* сброс всех восьми разрядов A (CLR A);
* инвертирование всех восьми разрядов A (CPL A);
* циклический сдвиг влево и вправо без учета флага переноса (RR A; RL A);
* циклический сдвиг влево и вправо с учетом флага переноса (RRC A; RLC A);
* обмен местами старшей и младшей тетрад внутри аккумулятора (SWAP A).

### Команды ветвления и передачи управления

Команды ветвления позволяют реализовывать условные операторы и операторы циклов. В микроконтроллерах семейства MCS-51 доступны следующие команды:

* безусловный переход: LJMP, AJMP, SJMP
* вызов и возврат из подпрограммы: LCALL, ACALL, RET, RETI
* проверка содержимого аккумулятора: JZ, JNZ, CJNE, JMP
* проверка флага переноса С: JC, JNC
* проверка содержимого любого бита в битовом пространстве: JB, JNB, JBC

Команды 16-разрядных безусловных переходов и вызовов подпрограмм позволяют осуществить переход в любую точку адресного пространства памяти программ объемом до 64 Кбайт.

Команды проверки содержимого аккумулятора и флага переноса C могут быть использованы для реализации проверки различных условий. При этом содержимое не изменяется.

Косвенный переход JMP @A+DPTR в системе команд микроконтроллеров семейства MCS-51 обеспечивает ветвление программы по содержимому аккумулятора А., что позволяет реализовывать операцию перехода по заданному коду.

1. Структурная схема МПС

На чертеже ИАЛЦ 463617 004 Э1приведена структурная схема МПС, ядром которой является микроконтроллер МК1816ВЕ51.

В состав МПС входят такие основные функциональные части:

1. Микроконтроллер ВЕ51.
2. Внешняя память данных – 10 страниц по 64К, внешняя память программ – 2 страницы по 64К.
3. Контроллер прямого доступак памяти. Контроллер прямого доступа централизованный.
4. Внешние устройства – 116 единиц.
5. Контроллер приоритетного прерывания. КПП также централизованный.
6. Дополнительные порты.
7. Периферийный адаптер ВВ55 для подключения портов P4, P7.
8. Программная часть

## Подпрограмма умножения

Подпрограмма основана на байтовом умножении, реализованном в МК51 командой MUL AB. Входные операнды 16-разрядные, первый разряд – знак операнда.

Начальные данные: X=R1R0; Y=R3R3. Результат формируется в R3R2R1R0.

### Блок-схема алгоритма

Начало

R1[7]=1

Очистка бита знака для X

Установка бита знака для X

1

R3[7]=1

Очистка бита знака для Y

Установка бита знака для Y

1

Stack1 = R2\*R1+C(R2\*R0);R2\*R0

Stack2 = R3\*R1+C(R3\*R0);R3\*R0

R3R2R1R0=Stack1+Stack2

Конец

0

R3[7]=1

Установка бита знака

1

0

0

## Подпрограмма деления

Подпрограмма основана на байтовом делении, реализованном в МК51 командой DIV AB. Входные операнды 8-разрядные, первый разряд – знак операнда.

Начальные данные: X=R0; Y=R1. Результат формируется в R0, остаток от деления в R1.

### Блок-схема алгоритма

Начало

R0[7]=1

Очистка бита знака для X

Установка бита знака для X

1

R1[7]=1

Очистка бита знака для Y

Установка бита знака для Y

1

Выполнение деления DIV AB

Конец

0

R0[7]=1

Установка бита знака

1

0

## Блок-схема алгоритма вычисления основной функции

Начало

Конец

Ввод X5

Ввод X6

Проверка знаков X5, X6

Выполнение операции (X5+X6)

Выполнение операции

Ввод X1

Ввод X2

Проверка знаков

Выполнение операции (X1-X2)

Выполнение операции

Ввод X9

Выполнение операции

## Листинг программы

; Ввод X5

mov r1,#11100011b

mov r0,#11010111b

; Ввод X6

mov r3,#10110101b

mov r2,#10000110b

;Проверка знаков

acallchksn

; Выполнение операции X5+X6

acall add16

; Подготовка к выполнению операции (X5+X6)^2

acallkvadrat

; Выполнение операции (X5+X6)^2

acall umul16

; Пересылка данных в R7R6R5R4

acallsend

; Ввод X1

mov r1,#11100011b

mov r0,#11010111b

; Ввод X2

mov r3,#10110101b

mov r2,#10000110b

;Проверка знаков

acallchksn

; Выполнение операции X1-X2

acall sub16

; Подготовка к выполнению операции (X1-X2)^2

acallkvadrat

; Выполнение операции (X1-X2)^2

acall umul16

; Выполнение операции ((X1-X2)^2)+((X5+X6)^2)

acall add32

; Ввод X9

mov r4,#11100011b

; Выполнение операции (X9)^2

acall squareX9

; Выполнение операции ((X1-X2)^2)+((X5+X6)^2)+(X9)^2

acallLastadd

; Необходимости проверять знаки нет, т.к. ответ всегда положителен.

; Конец.

end

;-------------------------------------

; Процедура проверки знаков операндов.

; Входные данные: первый операнд - R1R0, второй операнд - R3R2.

; Анализируются первый бит в R1 и первый бит в R3.

;-------------------------------------

chksn: mov a,01H ;mov a,r1

jb acc.7, clearr11 ; Если 1, то операнд отрицательный.

mov a,r3

jb acc.7, clearr31 ; Если 1, то операнд отрицательный.

ret

clearr11: clr acc.7 ; Очистка знакового разряда.

mov r1,a

mov a, r0 ;Конвертация в ДК R1R0.

cpl a

add a, #1

mov r0, a

mov a,01H ;mov a,r1

cpl a

addc a, #0

mov r1, a

ret

clearr31: clr acc.7 ; Очистка знакового разряда.

mov r3,a

mov a, r2 ;Конвертация в ДК R3R2.

cpl a

add a, #1

mov r2, a

mov a, r3

cpl a

addc a, #0

mov r3, a

ret

;-------------------------------------

; Умножение 16-разрядных операндов.

; Входные данные: первый операнд - R1R0, второй операнд - R3R2.

; Результат: R3R2R1R0.

;-------------------------------------

UMUL16: push B

pushdpl

mov a, r0

mov b, r2

mulab ;multiply XL x YL

pushacc ;stackresultlowbyte

push b ;stackresulthighbyte

mov a, r0

mov b, r3

mulab ;multiply XL x YH

pop 00H

add a, r0

mov r0, a

mov a,#00h

addc a, b

movdpl, a

mov a, r2

mov b, r1

mulab ;multiply XH x YL

add a, r0

mov r0, a

mov a, dpl

addc a, b

movdpl, a

mov a,#00h

addc a, #0

pushacc ;saveintermediatecarry

mov a, r3

mov b, r1

mulab ;multiply XH x YH

add a, dpl

mov r2, a

popacc ;retrievecarry

addc a, b

mov r3, a

mov r1, 00H

pop 00H ;retrieveresultlowbyte

popdpl

pop B

ret

;-------------------------------------

; Вычитание 16-разрядных операндов.

; Входные данные: первый операнд - R1R0, второй операнд - R3R2.

; Результат: R1R0.

;-------------------------------------

SUB16: mov a, r0 ;load X lowbyteintoacc

clr C ;clearcarryflag

subb a, r2 ;subract Y lowbyte

mov r0, a ;putresultin Z lowbyte

mov a,01H ;mov a,r1 load X highintoaccumulator

subb a, r3 ;subtract Y highwithborrow

mov r1, a ;saveresultin Z highbyte

;mov C, OV ;Overflow

ret

;-------------------------------------

; Сложение 32-разрядных операндов.

; Входные данные: первый операнд - R3R2R1R0, второй операнд - R7R6R5R4.

; Результат: R3R2R1R0.

;-------------------------------------

ADD32: mov a, r0 ;load X lowbyteintoacc

add a, r4 ;add Y lowbyte

mov r0, a ;saveresult

mov a,01H ;mov a,r1 load X nextbyteintoacc

addc a, r5 ;add Y nextbytewithcarry

mov r1, a ;saveresult

mov a, r2 ;load X nextbyteintoacc

addc a, r6 ;add Y nextbyte

mov r2, a ;saveresult

mov a, r3 ;load X highbyteintoacc

addc a, r7 ;add Y highbytewithcarry

mov r3, a

;mov C, OV ;Overflow

ret

;-------------------------------------

; Сложение 16-разрядных операндов.

; Входные данные: первый операнд - R1R0, второй операнд - R3R2.

; Результат: R1R0.

;-------------------------------------

ADD16: mov a, r0 ;load X lowbyteintoacc

add a, r2 ;add Y lowbyte

mov r0, a ;putresultin Z lowbyte

mov a,01H ;mov a,r1 load X highbyteintoacc

addc a, r3 ;add Y highbytewithcarry

mov r1, a ;saveresultin Z highbyte

mov C, OV ;Overflow

ret

;-------------------------------------

; Пересылка данных из R1R0в R3R2.

;-------------------------------------

kvadrat: mov a,r0

mov r2,a

mov a,01H ;mov a,r1

mov r3,a

ret

;-------------------------------------

; Поднесение в квадрат 8-разрядного операнда.

; Входные данные: R4.

; Результат: R5R4.

;-------------------------------------

squareX9: mov a, r4 ;read X and ...

mov b, r4 ; ... Y

mulab ;multiply X and Y

mov r5, b ;saveresulthigh ...

mov r4, a ; ... andlow

ret

;-------------------------------------

; Сложение 32-разрядного и 16-разрядного операндов.

; Входные данные: первый операнд - R3R2R1R0, второй операнд - R5R4.

; Результат: R3R2R1R0.

;-------------------------------------

lastadd: mov a, r0 ;load X lowbyteintoacc

add a, r4 ;add Y lowbyte

mov r0, a ;saveresult

mov a,01H ;mov a,r1 load X nextbyteintoacc

addc a, r5 ;add Y nextbytewithcarry

mov r1, a ;saveresult

mov a, r2 ;load X nextbyteintoacc

addc a, #00h ;add Y nextbyte

mov r2, a ;saveresult

mov a, r3 ;load X highbyteintoacc

addc a, #00h ;add Y highbytewithcarry

mov r3, a

;mov C, OV ;Overflow

ret

;-------------------------------------

; Пересылка данных из R3R2R1R0ВR7R6R5R4.

;-------------------------------------

send: mov a,r3

mov r7,a

mov a,r2

mov r6,a

mov a,01H ;mov a,r1

mov r5,a

mov a,r0

mov r4,a

ret

;-------------------------------------

; Деление 8-разрядных операндов.

; Входные данные: первый операнд - R0, второй операнд - R1.

; Результат: R0 - ответ, R1 - остаток.

;-------------------------------------

UDIV8: push b

mov a, r0 ;read X and ...

mov b, r1 ; ... Y

divab ;divide X and Y

mov r0, a ;saveresultquotient

mov r1, b ;saveremainder

pop b

ret

# ВЫВОДЫ

Разработана микропроцессорная система на основе МК-51с подключением внешней памяти данных, внешней памяти программ, периферийного адаптера, внешних устройств. МК-51 – это усовершенствованный МК-48, ряд нововведений в этом микроконтроллере позволяет облегчить написание программ под него, а также работу с портами. По сравнению с ВЕ48 , он имеет встроенную память программ , расширенный набор команд. Программы реализации типовых процедур обработки данных выполняются в МК51 в 4 - 15 раз быстрее и занимают в 1.5 - 3 раза меньше места в памяти, чем в МК48. При разработке системы, имеющей определенное назначение, необходимо ответственно отнестись к выбору микроконтроллера со всеми необходимыми возможностями и параметрами.

# ЛИТЕРАТУРА

1. <http://ru.wikipedia.org/wiki/Intel_8051>
2. <http://digital.sibsutis.ru/content.htm>
3. <http://www.intuit.ru/department/hardware/microarch/8/1.html>
4. <http://grantronics.com.au/docs/8051inst.pdf>
5. Пухальский Г.И. Проектирование микропроцессорных систем. СП: Политехника,2001.-544с.
6. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум.- К.: Высш.шк. 1989. - 124 с.
7. GhoshalSubrata 8051 Microcontroller :Internals, Instructions, Programming&Interfacing