Национальный технический университет Украины

«Киевский Политехнический Институт»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

Расчетно-графическая работа

по дисциплине

«Архитектура компьютера»

Выполнила:

Ст. Потушинский Ярослав Сергеевич

Группа ІО-81

Факультет ИВТ  
Зачетная книжка № 8117

Номер технического задания 1111110110101

Допущен к защите\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

(подпись руководителя)

Киев 2010 г.

Описание альбома

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *N п.п.* | *Формат* | *Обозначение* | | | *Наименование* | *К. листов* | | *№ экз.* | | *Примечание* | | |
|  |  |  | | | Документация общая |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | | 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 | | | Устройство ввода-вывода | 1 | |  | |  | | |
|  |  |  | | | *Схема функциональная* |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  | А4 | ИАЛЦ 463617.003ПЗ | | | Пояснительная записка | 23 | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  | | |  |  | |  | |  | | |
|  |  |  |  |  | *ИАЛЦ 463617.001 ОА* | | | | | | | |
|  |  |  |  |  |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| *Выполнил* | | *Потушинский Я.С.* |  |  | Микропроцессорная система  *Описание альбома* | | *Лит.* | | | | *Лист* | *Листов* |
| *Проверил* | | *Ткаченко В.В.* |  |  |  | |  |  | *1* | *1* |
|  | |  |  |  | *НТУУ «КПИ» ФИВТ*  *Гр. ИВ-81* | | | | | |
| *Н. контр.* | |  |  |  |
| *Зав. каф.* | |  |  |  |

Техническое задание

Техническое задание на расчетную работу

Проектирование микропроцессорных систем

Студентка Потушинского Я. С.

Группы ИВ-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) |
| Количество ВУ | 116 |
| Функция |  |
| Вспомогательные порти, периферийные адаптеры | ВВ55, Р4,Р7 |
| Адреса для интерфейса внешнего устройства | Самостоятельно |
| Адреса для ППА | F0h, F1h, F2h, F3h |
| Внешняя память данных | по 64К(5 страниц) |
| Внешняя память программ | по 64К(2 страницы) |
| Функциональная схема | Устройство ввода + устройство вывода (интерфейсы) |

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)

[3. Система команд 15](#_Toc280296584)

[4. Программная часть 16](#_Toc280296585)

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

[4.2 Листинг программы 19](#_Toc280296591)

[5. Функциональная схема 23](#_Toc280296592)

[Выводы 23](#_Toc280296593)

**Вступление.**

Микропроцессоры (МП) представляют собой автономные функционально законченные устройства, состоящие из одной или нескольких программно-управляемых интегральных микросхем высокой степени интеграции, включающие все средства, необходимые для обработки информации и управления данными рассчитанные на совместную работу с устройствами памяти и ввода-вывода информации.

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.

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.

1. **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 квази-двунаправ­ленных.
* 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.

1. **Построение микропроцессорной системы (МПС)**
   1. **Описание архитектуры МПС**

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

* Процессор
* Основная память (ОЗУ и 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 адреса и данных объединена. КПП и КПДП централизованы. Каждое ВУ имеет фиксированный адрес в адресном пространстве периферийных устройств. Количество внешних устройств – 116. Система команд – комплексная.

* 1. **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.
   1. **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.

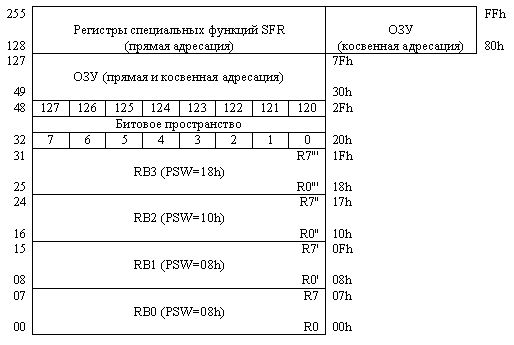


Рисунок 2.1. Адресное пространство внутренней памяти данных

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.

* 1. **Внешние устройства (ВУ) и периферийный программируемый адаптер (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.

* 1. **Режим прерываний**

Под прерыванием понимают временную приостановку выполнения программы или 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/счётчиков*/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.

* 1. **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

* 1. **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.

1. **Система команд**
   1. **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.

* 1. **Арифметические и логические команды**

Группа команд арифметических операций. Данная группа состоит из 8 основных команд и позволяет выполнять следующие операции над 8-битными целыми двоичными числами без знака:

|  |  |
| --- | --- |
| Название команд | Мнемокод |
| 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.

Группа команд логических операций. Данная группа состоит из 28 команд и позволяет выполнять следующие операции над байтами:

|  |  |
| --- | --- |
| Название команд | Мнемокод |
| Логическое «И» | ANL |
| Логическое «ИЛИ» | ORL |
| Исключающее «ИЛИ» | XRL |
| 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 |

* 1. **Команды передачи управления**

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

|  |  |
| --- | --- |
| Название команд | Мнемокод |
| Безусловный переход | LJMP, AJMP, SJMP |
| Вызов и возврат из подпрограммы | LCALL, ACALL, RET, RETI |
| Проверка содержимого аккумулятора | JZ, JNZ, CJNE, JMP |
| Проверка содержимого любого бита в битовом пространстве | JB, JNB, JBC |
| Проверка флага переноса С | JC, JNC |

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. **Программная часть**

Задача реализовать функцию . Программа основана на использовании арифметических операций байтового умножения, делений и вычитания, реализованных в МК51 командами:

- MUL, DIV, SUBB

* 1. **Блок-схема алгоритма**

Программа содержит арифметические операции выполненные по-блочно.

X1 – R1.R2

X2 – R3.R4 >> (X1-X2) – R5.R6

X3 – R1.R2

X4 – R3.R4 >> (X3\*X4) – R10.R11.R12.R13

X7 – R1.R2

X8 – R3.R4 >> (X7/X8) – R11.R12

(X1-X2) + (X3\*X4) + (X7/X8) >> R1.R2.R3.R4



* 1. **Листинг программы**

;Ввод X1

mov R1, #6Fh

mov R2, #15h

;Ввод X2

mov R3, #40h

mov R4, #09h

;Проверка знаков

acall checksign

;Выполнение операции (x1-x2)

acall substraction

;Ввод Х3

mov R1, #7Fh

mov R2, #12h

;Ввод X4

mov R3, #39h

mov R4, #08h

;Проверка знаков

acall checksign

;Выполнение операции умножения (х3 \* х4)

acall multiplication

;Ввод Х7

mov R1, #8Fh

mov R2, #13h

;Ввод Х8

mov R3, #3Fh

mov R4, #10h

;Проверка знаков

acall checksign

;Выполненик операции деления (х7/х8)

acall divide

;Выполненик операции сложения (х1+х2)+(х3\*х4)+(х7/х8)

acall addzz

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

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

;Проверка знаков операндов. |

;Проверяются первый бит R1 и R3|

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

checksign:

mov a, R1

jb acc.7, clearR1; Если 1 то операнд отрицательный

chkR3:

mov a, R3

jb acc.7, clearR3; Если 1 то операнд отрицательный

ret

clearR1:

clr acc.7 ;Очистка знакового разряда

;Конвертация числа в ДК

mov R1, a

mov a, R2

cpl a

add a, #1h

mov R2, a

mov a, R1

cpl a

addc a, #0h

mov R1, a

ajmp chkR3

clearR3:

clr acc.7 ;Очистка знакового разряда

;Конвертация числа в ДК

mov R3, a

mov a, R4

cpl a

add a, #1h

mov R4, a

mov a, R3

cpl a

addc a, #0h

mov R3, a

ret

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

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

;Операция вычитания |

;(x1 - x2) |

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

substraction:

clr c; очистка знака переноса

mov a, R2; Загрузка младшых воьсми битов вычитаемого числа

subb a, R4; Вычитание младшых воьсми битов от R2

mov R6, a; Выгрузка результата в R6

mov a, R1; Загрузка старших 8-ми битов X1

subb a, R3; Вычитание старших восьми битов от X1

mov R5, a;

ret

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

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

;Операция умножения |

;(х3 \* х4) |

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

multiplication:

clr c

push b

mov a, R2

mov b, R4

mul ab

mov R13, a

mov a, R1

mov b, R4

mul ab

addc a, #00h

mov R12, a

mov a, R2

mov b, R3

mul ab

addc a, #00h

mov R11, a

mov a, R1

mov b, R3

mul ab

mov R10, a

ret

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

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

;Операция деления |

;(х7/х8) |

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

divide:

push b

mov a, R2

mov b, R4

div ab

mov R15, a

mov a, R1

mov b, R3

div ab

addc a,#00h

mov R14, a

ret

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

;(x1-x2)+(x3\*x4)+(x7/x8)

addzz:

mov a, R6

add a, R15

mov R4, a

mov a, R5

add a, R14

mov R3, a

mov a, R13

add a, R4

mov R4, a

mov a, R12

add a, R3

mov R3, a

mov a, R11

addc a, #00h

mov R2, a

mov a, R10

addc a, #00h

mov R1, a

ret

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

end

1. **Функциональная схема устройства ввода-вывода**

# 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