**ЗАДАНИЕ НА КУРСОВую РАБОТу**

1.1. Общие требования к курсовой работе

Каждый студент должен выполнить в требуемом объеме работу в соответствии с индивидуальным заданием, приобрести навыки проектирования аппаратной части МПС и разработать программу обработки данных в соответствии с заданием.

1. Задание на курсовое проектирование.

Требуется разработать микропроцессорную систему управления объектом (МПС). Объект задан видом и количеством данных поступающих с объекта, потребным ресурсом для обработки данных, видом и количеством управляющих сигналов.

Вариант задания определяется по цифрам учебного шифра студента в соответствии с табл. 1 и табл. 2.

Для выполнения курсовой работы необходимо:

* изучить рекомендуемую литературу и настоящие методические указания;
* определить свой вариант задания;
* разработать функциональную схему МПС;
* составить подпрограмму обработки данных, отладить ее с помощью программного эмулятора и получить листинг программы.

Таблица 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Цифра | Число каналов | | Объем памяти, Кбайт | |
| шифра | ввода | вывода | ПЗУ | ОЗУ |
| 0 | 8 | 20 | 0,5 | 16 |
| 1 | 16 | 16 | 1 | 16 |
| 2 | 24 | 12 | 2 | 12 |
| 3 | 32 | 8 | 4 | 8 |
| 4 | 44 | 4 | 8 | 16 |
| 5 | 12 | 18 | 12 | 10 |
| 6 | 20 | 14 | 16 | 8 |
| 7 | 28 | 10 | 16 | 2 |
| 8 | 40 | 6 | 32 | 4 |
| 9 | 48 | 2 | 40 | 1 |

Примечание.

1. По последней цифре определяется число каналов ввода и вывода информации МПС для взаимодействия с объектом управления.

2. По предпоследней цифре - объем внешней памяти программ (ПЗУ) и данных (ОЗУ).

Блок-схема алгоритма обработки данных представлена на рис. 1.



Рис. 1. Блок-схема алгоритма обработки данных

В соответствии c блоком 2 алгоритма должна быть произведена предварительная установка микроконтроллера (МК-51), т.е. определены начальный адрес программы, константы, приоритет и источники прерывания, режим работы таймеров/счетчиков и последовательного порта, должен быть очищен аккумулятор и регистры-счетчики и задан вид сигнала прерывания (по фронту). Блок 3 программы переводит МК-51 в режим ожидания прихода прерывания. Внешнее прерывание поступает со входа , если сумма двух последних цифр шифра четная, или , если сумма двух последних цифр шифра нечетная. После прихода прерывания МК-51 читает байт данных (Date) из порта P1 (блок 4) и производит его сравнение с константой (Const) (блок 5). По результатам сравнения МК-51 либо выводит принятый байт через последовательный порт (блок 6), либо формирует на заданной линии порта P3 импульс заданной длительности (блок 7). После этого осуществляется переход в режим ожидания прихода следующего прерывания.

Частота синхронизации МК-51, скорость передачи данных через последовательный порт, длительность импульса, банк памяти, где будут располагаться регистры – счетчики, номер линии порта P3 определяются в соответствии с табл. 2.

Таблица 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Цифра  шифра | Частота синхронизации, МГц | Длительность импульса, мс | Номер банка памяти | Номер линии порта P3 | Скорость передачи данных через УАПП, бит/с |
| 0 | 4 | 60 | 0 | 0 | 110 |
| 1 | 5 | 55 | 1 | 1 | 300 |
| 2 | 6 | 50 | 2 | 4 | 1200 |
| 3 | 7 | 45 | 3 | 5 | 2400 |
| 4 | 8 | 40 | 0 | 6 | 4800 |
| 5 | 9 | 35 | 1 | 7 | 7200 |
| 6 | 10 | 30 | 2 | 0 | 9600 |
| 7 | 10,5 | 25 | 3 | 1 | 14400 |
| 8 | 11 | 20 | 0 | 4 | 24000 |
| 9 | 12 | 15 | 1 | 5 | 28800 |

Примечание.

1. По последней цифре определяется частота синхронизации, длительность импульса и скорость передачи данных.

2. По предпоследней цифре – номера банка памяти и линии порта Р3.

1. Объем и содержание курсового работаа

Курсовая работа состоит из пояснительной записки и графической части.

Пояснительная записка, объемом 15…20 страниц, должна включать:

1. Задание на проектирование и исходные данные.
2. Теоретический вопрос.
3. Описание принципа работы разрабатываемой МПС, включающее функциональные схемы, временные диаграммы работы и т. д.
4. Блок-схему алгоритма работы МПС, реализующую заданную подпрограмму.
5. Текст программы на языке Ассемблер с необходимыми комментариями.

Графическая часть должна содержать функциональную схему разрабатываемой МПС.

Графические материалы должны быть оформлены с соблюдением требований ЕСКД.

Разработку электрической схемы МПС рекомендуется выполнять с использованием САПР, а разработку и отладку программы с помощью эмулятора однокристальных микроконтроллеров Single-Chip machine, который можно скачать по адресу: [http://dca.narod.ru/simulation/mk51.htm.](http://dca.narod.ru/simulation/mk51.htm.%20%20)

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ОСНОВНЫМ РАЗДЕЛАМ КУРСОВОЙ** **РАБОТЫ**

2.1. Теоретический вопрос

В этой части работы необходимо изложить особенности построения МПС на основе выбранного микроконтроллера под заданные условия функционирования системы.

Работу над данной частью рекомендуется начать с изучения лекционного материала и учебной литературы [1, 3, 4]. Не следует переписывать вопрос целиком из каких-либо монографий или учебников. В тексте обязательно должны быть ссылки на используемые литературные источники.

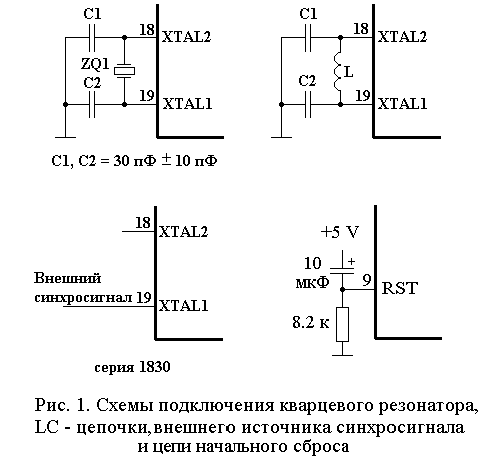
1. Разработка МПС под заданные условия работы

Проектирование МПС на основе микроконтроллеров - достаточно простая инженерная задача, так как многие решения являются стандартными и их можно найти в литературных источниках. Однако для многих применений число линий ввода/вывода оказывается недостаточным, поскольку требуется проводить опрос значительного количества датчиков и осуществлять управление многими исполнительными механизмами. В условиях применения внешней памяти программ и данных эта проблема еще более обостряется т.к. для связи с внешними объектами остается всего лишь два порта ввода/вывода. Кроме этого, в процессе разработки МПС приходится решать задачу согласования уровней сигналов, формируемых датчиками, нагрузочной способности микроконтроллера и выходных ключей, скорости работы остальных компонентов МПС. С учетом изложенного, рассмотрим типовые решения сформулированных задач.

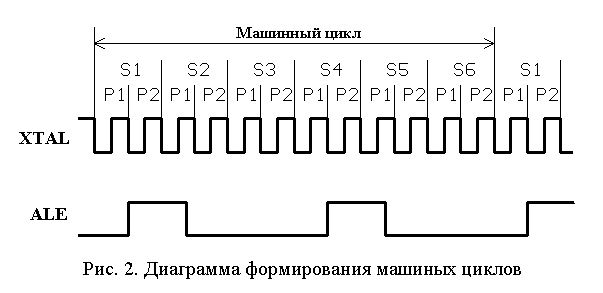
1. Синхронизация МК-51

Работу всех внутренних блоков МК-51 синхронизирует устройство выработки временных интервалов, которое в свою очередь управляется внутренним тактовым генератором. Период следования синхроимпульсов тактового генератора определяется частотой кварцевого резонатора, LC - цепочки или периодом следования сигналов внешнего источника. Схемы подключения кварцевого резонатора, LC - цепочки и внешнего источника приведены на рис. 2.

Практически все команды МК-51 выполняются за один или два машинных цикла, который имеет фиксированную длительность и состоит из шести состояний S1...S6. В свою очередь каждое состояние включает в себя две фазы Р1 и Р2, длительность которой равна периоду следования тактовых импульсов t. Таким образом, длительность каждого машинного цикла ТЦ равна двенадцати периодам тактовых импульсов , где - частота синхронизации МК-51.



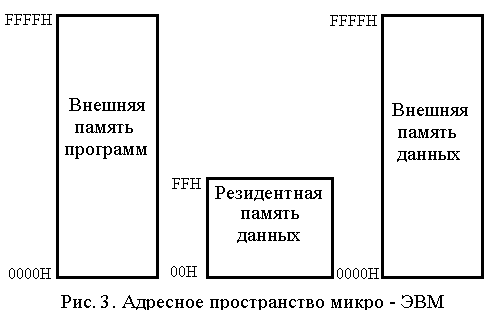
Для синхронизации работы остальных компонентов МПС можно использовать сигнал ALE, который дважды формируется в течении одного машинного цикла. Рис. 2 иллюстрирует сказанное.



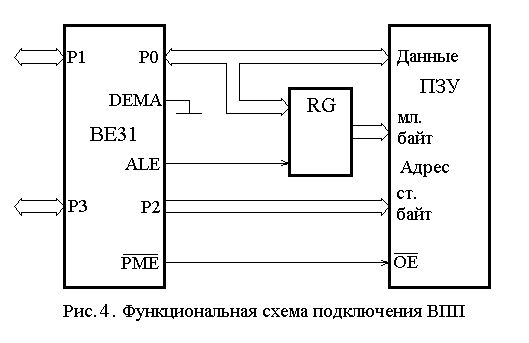
1. Использование внешней памяти

Все МК-51 серии 1830 (1816) имеют несколько адресных пространств, функционально и логически разделенных за счет разницы в механизмах адресации и сигналах управления записью и чтением:

* внешняя память программ (ВПП);
* резидентная память данных (РПД);
* внешняя память данных (ВПД).

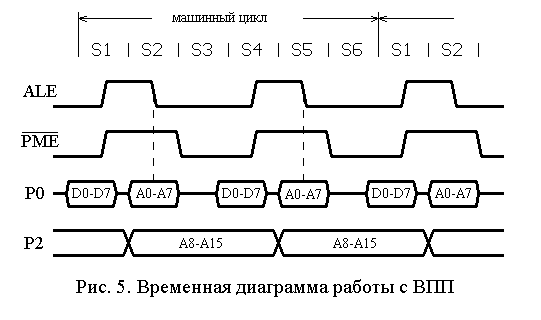


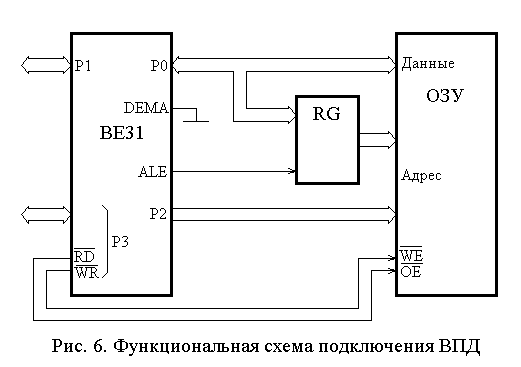
Структура адресного пространства ВЕ31 показана на рис. 3. Слева приведены адреса соответствующих областей памяти.



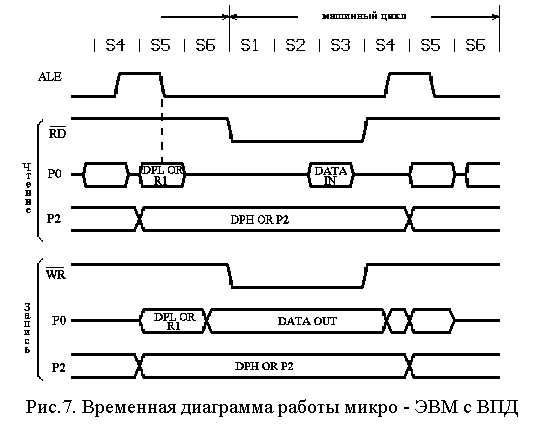
Микро - ЭВМ ВЕ31 не имеет внутренней памяти программ. Поэтому для ее работы необходимо использовать ВПП, объем которой может достигать 64 Кбайта. Функциональная схема подключения ВПП к МК-51 показана на рис. 4.

При обращении к ВПП всегда формируется 16 - разрядный адрес, младший байт которого выдается через порт Р0, а старший - через порт Р2. Причем порт Р0 используется в режиме временного мультиплексирования: в начале каждого машинного цикла обращения к ВПП (фаза S2Р1) через порт Р0 выдается младший байт адреса, который должен быть зафиксирован во внешнем регистре RG по отрицательному фронту сигнала ALE (рис. 5); низкий уровень сигнала PME, формируемый в течении фаз S3Р1 ... S4Р1 машинного цикла, разрешает выборку байта данных из ПЗУ, который затем поступает на линии порта Р0 и вводится в МК-51. Следует особо подчеркнуть, что на вывод DEMA ВЕ31 должен быть подан низкий уровень напряжения.





В некоторых МПС, рассчитанных на обработку больших массивов данных, РПД может оказаться недостаточно. В этом случае возникает необходимость использования внешнего оперативного запоминающего устройства (ОЗУ), которое может быть подключено к МК-51 так, как это показано на рис. 6. При этом обращение к ВПД возможно только с помощью команд MOVX. Команды MOVX @Ri, A и MOVX A, @Ri формируют восьмиразрядный адрес, который выдается на ОЗУ через порт Р0. Команды MOVX @DPTR, A и MOVX A, @DPTR формируют 16-ти разрядный адрес, младший байт которого выдается через порт Р0, а старший - через порт Р2. Так же как и при работе с ВПП, байт адреса, выдаваемый через порт Р0, должен быть зафиксирован во внешнем регистре по отрицательному фронту сигнала ALE, т.к. в дальнейшем линии порта Р0 используются для чтения или записи информации в/из МК-51. При этом чтение информации стробируется сигналом RD, формируемым ВЕ31, а запись - сигналом WR. На рис. 7 показана временная диаграмма работы МК-51 в режимах обмена данными с ВПД.



* 1. Разработка и отладка подпрограммы обработки данных
     1. Правила разработки программ на языке Ассемблер

Разработка прикладного программного обеспечения на языке Ассемблере - творческая задача, требующая от программиста отличного знания программистской модели МК-51, состава и формата команд, способов адресации операндов и т.д. Однако существуют формальные правила составления программ, соблюдение которых позволяет даже начинающему разработчику составлять работоспособные программы. Подробно эти правила рассмотрены в [1], согласно которым для получения текста исходной программы необходимо выполнить следующую последовательность действий:

* составить подробное описание исходной задачи;
* выполнить инженерную интерпретацию задачи, желательно с привлечением того или иного аппарата формализации (сети Петри, графа автомата и т.п.);
* разработать блок-схемы алгоритма работы МПС;
* разработать детализированные блок-схемы алгоритмов отдельных процедур, выделенных на основе модульного принципа составления программ;
* распределить рабочие регистры и память МК-51;
* составить текст исходной программы.

Исходный текст программы на языке Ассемблера имеет определенный формат, состоящий, обычно, из четырех полей:

МЕТКА МНЕМОНИКА ОПЕРАНД КОММЕНТАРИЙ.

Поля отделяются друг от друга произвольным числом пробелов.

**Метка**. Метка - это имя, состоящее из букв латинского алфавита и стоящее в начале строки. После метки ставится двоеточие. В директивах Ассемблера EQU, DB и DW метка принимает значение выражения, стоящего за директивой. В остальных случаях метка принимает значение $ (текущее значение счетчика команд). Перед директивой ORG использование меток запрещено.

**Мнемоника.** В поле Мнемоника записывается мнемоническое обозначение команды МК-51 или псевдокоманды ассемблера.

**Операнды.** В этом поле указываются операнды, участвующие в операции. Команды ассемблера могут быть без-, одно- или двухоперандными. Операнды разделяются запятой.

Операнд может быть задан непосредственно или в виде его адреса (прямого или косвенного). Непосредственному операнду обязательно должен предшествовать префикс ( # ). В качестве непосредственного операнда можно указывать число или символическое имя. Прямой адрес операнда может быть задан мнемоническим обозначением, числом или символическим именем. Указанием на косвенную адресацию служит префикс @. В командах передачи управления операндом может быть число, метка, косвенный адрес или выражение.

Используемые в качестве операндов символические имена должны быть определены в программе, а числа представлены с указанием системы счисления, для чего используется суффикс (буква, стоящая после числа): В - для двоичной, Q - для восьмеричной, H - для шестнадцатеричной. Число без суффикса по умолчанию считается десятичным.

**Комментарий.** Поле комментария всегда начинается после символа (;) и игнорируется Ассемблером. В нем допускается использовать любые символы. Чаще всего это поле используется программистом для пояснения логической организации программы.

**Директивы Ассемблера.** Строка программы может содержать директиву или команду. Директивы, в отличие от команд, не исполняются МК-51 и предназначены, в основном, для управления трансляцией программы. Ниже приводится список директив Ассемблера, которые поддерживает редактор, встроенный в эмулятор микроконтроллера:

ORG - изменение текущего значения счетчика команд;

EQU - определение имени;

DB - определение байта;

DW - определение слова (2 байта);

После составления текста программы необходимо получить объектный код, т.е. набор двоичной информации, содержащий коды команд и данных. Для простых программ объектный код может быть получен вручную. Однако для более сложных программ требуются специальные средства, позволяющие осуществить трансляцию программы в автоматическом режиме. В настоящее время для МК-51 серии 1830 разработано значительное количество кросс - средств, которые не только обеспечивают ввод и трансляцию программ составленных на языке Ассемблер, но и позволяющие проводить отладку программ.

Отладку разработанной программы рекомендуется выполнять с помощью эмулятора однокристальных микроконтроллеров Single-Chip machine.

После загрузки объектного модуля можно просмотреть и модифицировать содержимое регистров МК-51, ячеек памяти и флагов, вывести на печать или дисковые носители дисассемблированный текст, дампы памяти и т.д.

2.3.2. Разработка подпрограммы вывода информации через УАПП

Последовательный порт МК-51 может использоваться в качестве универсального асинхронного приемо-передатчика (УАПП) с фиксированной или переменной скоростью последовательного обмена и возможностью дуплексного включения. Скорость последовательного обмена УАПП в режимах 1 и 3 определяется по формуле:

, (1)

где - значение бита SMOD регистра управления мощностью PCON;

- частота синхронизации МК-51;

- десятичное значение содержимого регистра TH1. Если необходим последовательный обмен с очень низкой скоростью, то можно использовать Т/С 1 в режиме 16-ти разрядного таймера (режим 1), разрешив при этом прерывание от Т/С 1 с целью перезагрузки TH1 и TL1 в подпрограмме обслуживания прерывания.

Для использования Т/С 1 в качестве источника для задания частот  и  необходимо:

1. Запретить прерывания от Т/С 1;
2. Запрограммировать работу Т/С 1 в качестве таймера или счетчика, установив при этом для него один из режимов 0, 1 или 2;
3. Запустить Т/С 1 на счет.

Обычно для установки скорости передачи данных через последовательный порт таймер Т/С 1 включается в режим 2 (режим автозагрузки).

В качестве примера рассмотрим фрагмент подпрограммы, предназначенной для передачи данных через последовательный порт со скоростью 500 бит/с. Частота синхронизации МК-51 составляет , а бит SMOD = 0.

CLR TR1 ;останов таймера Т/С 1;

MOV TH1,#0DBH ;автозагружаемое значение для получения;

;скорости 500 бит/с, рассчитанное по;

;формуле (1) ;

MOV SCON,#11011100B ;установка режима 3 УАПП;

MOV TMOD,#00100000B ;установка режима 2 Т/С 1;

SETB TR1 ;запуск таймера;

LABEL:

JNB T1,LABEL ;ожидание окончания передачи;

CLR T1 ;очистка флага передачи;

MOV SBUF,A ;выдача 9 бит данных в последовательный;

;порт;

Флаг прерывания передатчика Т1 регистра SCON устанавливается аппаратно в конце времени выдачи 8-го бита в режиме 0 или в начале стоп бита в других режимах.

2.3.3. Формирование импульса заданной длительности

Во многих применениях МК-51 необходимо сформировать импульс заданной длительности на той или иной линии порта. Обычно для этих целей используется либо таймер/счетчик, либо регистры РПД. Блок-схема алгоритма формирования временной задержки большой длительности показана на рис. 8.



Рис. 8. Блок-схема алгоритма формирования временной задержки

Данный алгоритм для формирования временной задержки использует так называемые вложенные циклы. На начальном этапе в регистр R1 загружается число “большого” цикла X (блок 2), а в регистр R2 – число “малого” цикла Y (блок 3). Эти две команды выполняются за один машинный цикл. С помощью блоков 4, 5 подпрограммы содержимое регистра R2 уменьшается до 0 и осуществляется переход к блоку 6 подпрограммы. Этот цикл удобнее организовать с помощью команды DJNZ R, $, которая выполняется МК-51 за 2 машинных цикла. Аналогично организуется и “большой” цикл (блоки 6, 7). Таким образом, длительность временной задержки, формируемой с помощью этого алгоритма, в машинных циклах можно рассчитать по формуле (2):

. (2)

Время задержки в секундах рассчитывается по формуле (3):

. (3)

Если задана длительность импульса, формируемого с помощью рассмотренного алгоритма, то необходимо подобрать такие X и Y, чтобы выполнялось условие (4):

, (4)

где - заданная длительность импульса.

Для того чтобы МК-51 сформировал импульс с точностью плюс/минус 1 цикл в подпрограмму необходимо добавить команду “нет операции” (NOP), число которых можно рассчитать по формуле (5):

. (5)

Если число команд NOP, необходимых для “подгонки” задержки, получится больше 5, то рекомендуется для этих целей использовать локальный цикл, который вставляется в тело подпрограммы после блока 7.

**Литература**

1. Горелик А.В., Горелик В.Ю., Ермаков А.Е., Ермакова О.П. Микропроцессорные информационно-управляющие системы железнодорожного транспорта. – М.: МИИТ, 2011. – 230 с.
2. Магда Ю.С. Микроконтроллеры серии 8051: Практический подход –М.: ДМК, 2008. 223 с.
3. Однокристальные микроЭВМ./ Боборыкин А.В., Липовецкий Г.П. и др. М.: МИКАП, 1994. - 400 с.
4. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах. М.: Энергоатомиздат, 1990. - 224 с.