Контрольные вопросы

1. Как усовершенствовать принципиальную схему рис. 2.1?

Резистор R1 можно исключить, заменив его внутренним нагрузочным резистором микроконтроллера. Дело в том, что микроконтроллеры серии AVR имеют встроенные нагрузочные резисторы для каждого разряда порта. Главное при написании программы – не забыть включить программным путем соответствующий резистор.

Кварцевый резонатор Q1 обеспечивает работу встроенного тактового генератора. Конденсаторы С2 и СЗ – это цепи согласования кварцевого резонатора. Элементы Cl, R2 – это стандартная цепь начального сброса. Такая цепь обеспечивает сброс микроконтроллера в момент включения питания. Еще недавно подобная цепь была обязательным атрибутом любой микропроцессорной системы. Однако современные микроэлектронные технологии позволяют исключить обе эти цепи. Дело в том, что большинство микроконтроллеров AVR, кроме тактового генератора с внешним кварцевым резонатором, содержат внутренний RC-генератор, не требующий никаких внешних цепей. Если не предъявляются высокие требования к точности и стабильности частоты задающего генератора, то микросхему можно перевести в режим внутреннего RC-генератора и отказаться как от внешнего кварца (Q1), так и от согласующих конденсаторов (С2 и СЗ). Цепь начального сброса тоже можно исключить. Любой микроконтроллер AVR имеет внутреннюю систему сброса, которая в большинстве случаев обеспечивает стабильный сброс при включении питания.

2. Как и для чего осуществляется настройка портов микроконтроллера?

установить начальное значение для вершины стека микроконтроллера;

настроить порт В на вывод информации;

подать на выход РВ.0 сигнал логической единицы (потушить светодиод);

сконфигурировать порт D на ввод;

включить внутренние нагрузочные резисторы порта D.

3. Дайте словесный алгоритм работы микропроцессорного устройства

прочитать состояние младшего разряда порта PD (PD.0);

если значение этого разряда равно единице, выключить светодиод;

если значение разряда PD.0 равно нулю, включить светодиод;

перейти на начало цикла.

4. Опишите порядок оформления программ на ассемблере

программа записывается в несколько колонок;

аналогичные элементы разных команд принято размещать друг под другом;

самая первая (левая) колонка зарезервирована для меток;

если метка отсутствует, место в колонке пустует;

следующая колонка предназначена для записи операторов;

затем идет колонка для операндов;

оставшееся пространство (крайняя колонка справа) предназначено для комментариев.

5. Что такое оператор и операнды ассемблерной команды?

Оператор - это некий эквивалент реальной команды микроконтроллера и в процессе трансляции заменяется соответствующим машинным кодом, который помещается в файл результата трансляции.

6. Сравните функции оператора и директивы в языке Ассемблер.

Если оператор - это некий эквивалент реальной команды микроконтроллера и в процессе трансляции заменяется соответствующим машинным кодом, который помещается в файл результата трансляции, то директива, хотя по форме и напоминает оператор, но не является командой процессора.

Директивы - это специальные вспомогательные команды для транслятора, определяющие режимы трансляции и реализующие различные вспомогательные функции.

Контрольные вопросы

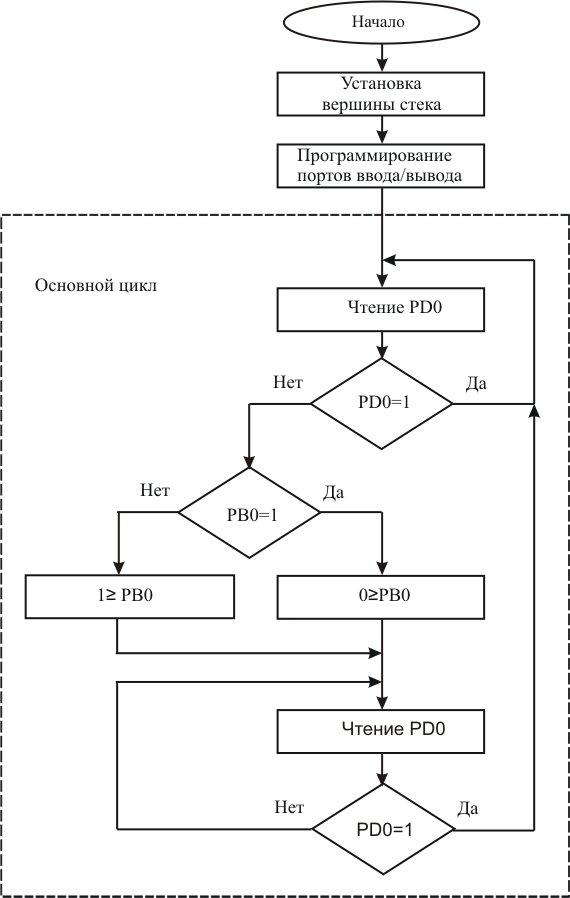
1. Как осуществляется настройка портов микроконтроллера?

* установить начальное значение для вершины стека микроконтроллера;
* настроить порт В на вывод информации;
* подать на выход РВ.0 сигнал логической единицы (потушить светодиод);
* сконфигурировать порт D на ввод;
* включить внутренние нагрузочные резисторы порта D.

3. Дайте словесный алгоритм работы микропроцессорного устройства;

* Прочитать состояние младшего разряда порта PD (PD.0).
* Если значение этого разряда равно единице, перейти к началу цикла.
* Если значение разряда PD.0 равно нулю, изменить состояние выхода РВ.О на противоположное.
* Перейти к началу цикла.

4. Представьте граф-схему алгоритма работы МП-устройства;



5. Дайте характеристику командам передачи управления sbrc и sbrs;

**sbrc**

Команда из группы условных переходов. Вызывает пропуск следующей за ней команды, если соответствующий разряд РОН сброшен. У команды два параметра. Первый параметр – **имя регистра общего назначения,** второй параметр – **номер проверяемого бита.**

**sbrs**

Команда, обратная предыдущей. Пропускает следующую команду, если соответствующий разряд РОН установлен в единицу. Имеет те же два параметра, что и команда sbrc.

6. Дайте характеристику командам sbi и сbi операций с разрядами;

**sbi**

Сброс в ноль одного из разрядов порта ввода-вывода. Команда имеет два параметра: имя порта и номер сбрасываемого разряда.

**сbi**

Установка в единицу одного из разрядов порта ввода-вывода. Имеет такие же два параметра, как и предыдущая команда.

1. Дребезг контактов и его влияние на работу схемы;

При работе микроконтроллера с датчиками, имеющими механические или электромеханические контакты, (кнопки, клавиши, реле и клавиатуры), возникает явление, называемое дребезгом. Это явление заключается в том, что при замыкании контактов возможно появление отскока контактов, которое приводит к переходному процессу. При этом сигнал с контакта может быть прочитан микроконтроллером как случайная последовательность нулей и единиц.

2. Опишите способы борьбы с дребезгом;

Подавить это нежелательное явление можно схемотехническими средствами с использованием буферного триггера, но чаще это делается программным путем.

3. Как реализуется временная задержка программными методами?

Обнаружив первый же нулевой уровень на входе, программа должна перейти в режим ожидания. В режиме ожидания программа приостанавливает все свои действия и просто отрабатывает задержку. Время задержки должно быть выбрано таким образом, чтобы оно превышало время дребезга контактов. Такую же процедуру задержки нужно ввести в том месте программы, где она находится в цикле ожидания отпускания кнопки, т.е. появления высокого потенциала.

4. Дайте характеристику командам push и pop работы со стеком;

**push**

Запись содержимого регистра общего назначения в стек. У данного оператора всего один операнд – имя регистра, содержимое которого нужно поместить в стек.

**pop**

Извлечение информации из стека. У этого оператора тоже всего один операнд – имя регистра, в который помещается информация, извлекаемая из стека.

5. Дайте характеристику командам rcall и ret работы с подпрограммами;

**rcall**

Переход к подпрограмме. У этого оператора всего один параметр – относительный адрес начала подпрограммы. Относительный адрес – это просто смещение относительно текущего адреса. Выполняя команду rcall, микроконтроллер запоминает в стеке текущий адрес программы из счетчика команд и переходит по адресу, определяемому смещением.

**ret**

Команда выхода из подпрограммы. По этой команде микроконтроллер извлекает из стека адрес, записанный туда при выполнении команды rcall, и осуществляет передачу управления по этому адресу.

6. Дайте характеристику команде условного перехода brne;

**brne**

Оператор условного перехода (переход по условию). У этого оператора всего один параметр – относительный адрес перехода. Условие перехода звучит как «не равно».

7. Как рассчитывать программно-управляемые временные задержки?

Предположим, что в управляющей программе необходимо реализовать временную задержку 153 мкс. Для ее получения требуется определить число Х, загружаемое в рабочий регистр. Определение числа Х выполняется на основе расчета времени выполнения команд, образующих данную подпрограмму (см. листинг 3.1). При этом необходимо учитывать, что команды push и pop, ldi и ret выполняются однократно, а число повторений команд dec и brne равно числу Х. Кроме того, обращение к подпрограмме временной задержки осуществляется по команде rcall wait, время исполнения которой также необходимо учитывать при подсчете временной задержки. В сводной таблице команд Ассемблера микроконтроллеров AVR указывается, за сколько тактов исполняется каждая команда: rcall – 3 такта, push – 2, ldi – 1, dec – 1, brne – 2 (команда brne выполняется за один такт, если не вызывает перехода, и за два такта, если вызывает переход), pop – 2, ret – 4 такта.

Тактовая частота кварцевого резонатора у нас равна 4 МГц. Длительность одного такта равна 0,25 мкс. Таким образом, подпрограмма выполняется за время (3 + 2 + 1 + (1+2) Х +2 +4) 0,25 мкс = (12 +3Х) 0,25 мкс= (3+0,75Х) мкс. Из уравнения 3+0,75Х = 153 получим Х = 200

В данном случае при загрузке в регистр числа 200 требуемая временная задержка (153 мкс) реализуется точно. Если число Х получается дробным, то временную задержку можно реализовать лишь приблизительно.