**СОДЕРЖАНИЕ**

Введение

1. Словесное описание работы системы

2. Предварительное распределение памяти

3. Алгоритм функционирования микропроцессорной системы

4. Распределение ресурсов

5. Программа работы системы

6. Контрольный пример

Заключение

Список использованной литературы

#### ВВЕДЕНИЕ

В данной курсовой работе описывается отбраковка резисторов на производстве. Резисторы сортируются по допускам и раскладываются в соответствующие контейнеры. Если сопротивление не входит ни в один диапазон допуска, он помещается в отдельный контейнер и включает сигнал, что попался брак. Система построена на микропроцессоре К1816ВЕ48.

Измерение сопротивления производиться посредством измерения падения напряжения на исследуемом резисторе при пропускании через него фиксированного тока.

1. **СЛОВЕСНОЕ ОПИСАНИЕ РАБОТЫ СИСТЕМЫ**

Принципиальная схема системы приведена на рис. 1.1

Работает система следующим образом:

В некоторый момент времени процессор подает роботу команду установить резистор и начинает ждать. Робот, когда установит резистор в измеряющее устройство, сигнализирует об этом процессору. Процессор выходит из режима ожидания и выдает команду начать преобразование и снова начинает ждать. АЦП, завершив преобразования падения напряжения на измеряемом резисторе в цифровой код, подает сигнал процессору. Процессор считывает с АЦП цифровой код и приступает к сравнению его с записанным в память эталонным сопротивлением. В результате вычислений процессор определяет к какой группе по отклонению от номинала относится измеряемый резистор и выдает соответствующую команду роботу- поместить резистор в один из пяти контейнеров с отклонениями

Далее цикл повторяется сначала.

1. **ПРЕДВАРИТЕЛЬНОЕ РАСПРЕДЕЛЕНИЕ ПАМЯТИ**

Предварительное распределение памяти в системе показано на рис. 2.1. Т. к. программа, управляющая системой, скорее всего, будет сравнительно небольшой, то она вся поместиться во внутреннем ПЗУ процессора (памяти компьютера), поэтому на рисунке изображена только эта память; внешние ПЗУ не нужны и поэтому распределение для них не показано.

Система будет обрабатывать сравнительно небольшой объем данных, поэтому показания на схеме распределения памяти данных область “ОЗУ данных” скорее всего, останется незадействованной.

|  |  |  |  |
| --- | --- | --- | --- |
| 3FFhYYYhXXXh000h | Свободно | 3Fh20h1Fh18h17h08h07h00h  | ОЗУ данных |
| Банк регистров RB1 |
| Подпрограмма |
| 8-уровневый стек |
| Программа, управляющая работой системы |
| Банк регистров RB0 |

**Рис. 2.1.**Память команд Память данных

**3.АЛГОРИТМ ФУНКЦИОНИРОВАНИЯ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ**

Укрупненная структурная схема алгоритма программы, управляющей процессором, изображена на рис. 3.1.

При включении системы вначале она принудительно переходит в режим занесения эталонного сопротивления. Процессор считывает с АЦП значение сопротивления, записывает его в память и обнуляет все счетчики, в которых ведется учет резисторов с определенным допуском.

Далее следует установка и измерение сопротивления очередного резистора. Считанное с АЦП значение сопротивления подвергается обработке процессором и вычисляется процент отклонения сопротивления резистора от эталонного. В зависимости от значения этого отклонения процессором выдается команда роботу на размещение резистора в определенном контейнере и увеличение на единицу соответствующего счетчика резисторов.

Далее анализируется состояние переключателя “ЭТАЛОН”. Если он замкнут, то снова производится замена в памяти эталонного сопротивления и обнуление счетчиков. Если этот переключатель разомкнут, то система начинает обработку следующего резистора.

Установка эталона или обычное измерение?

Установка эталона

Обычное измерение

Размещение резистора в нужном контейнере и увеличение соответствующего счетчика на единицу

Определение группы по допуску

Измерение сопротивления

Обнуление счетчиков; запись измеренного сопротивления в память как эталон

# Измерение сопротивления

**4. РАСПРЕДЕЛЕНИЕ РЕСУРСОВ**

В регистре R0 банка 0 находится текущее эталонное сопротивление. Остальные регистры банка 0 используются по мере надобности для хранения промежуточных результатов и временного сохранения содержимого аккумулятора.

В регистре R0 банка 1 находится счетчик количества резисторов с сопротивлением вне допуска.

В регистре R2 банка 1 находится счетчик количества резисторов с отклонением не более 10% от эталона.

В регистре R2 банка 1 находится счетчик количества резисторов с отклонением не более 10% от эталона.

В регистре R2 банка 1 находится счетчик количества резисторов с отклонением не более 10% от эталона.

В регистре R2 банка 1 находится счетчик количества резисторов с отклонением не более 10% от эталона.

В регистре R2 банка 1 находится счетчик количества резисторов с отклонением не более 10% от эталона.

Через вывод Р10 процессору сообщается режим работы:

1. сортировка резисторов по допускам;
2. смена эталонного сопротивления.

Через вывод Р20 роботу выдается команда “установить резистор”. Активное состояние – 1.

Через вывод Р21 роботу выдается команда положить резистор в контейнер “ВНЕ ДОПУСКА”. Активное состояние – 1.

Через вывод Р22 осуществляется запуск АЦП. Активное состояние – 0.

Через выводы Р25, Р26, Р27, Р28 и Р29 роботу выдается команда положить резистор в контейнер с допуском соответственно. Активное состояние – 1.

На вывод Т0 от робота поступает 1, если резистор установлен.

На вывод Т1 от АЦП поступает 1, когда данные готовы к считыванию.

Программа в памяти начинается с адреса 000h.

**5. ПРОГРАММА РАБОТЫ СИСТЕМЫ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Адрес | Код | Количество циклов | Метка | Мнемоника | Комментарий |
| 00000200300400800А00С00D00F0100111201301501701801901A01B01D01E02002102302402602702802A02B02C02D02E02F03003103203303403503703803A03B03C | 14 70A8D5D8 00BA 00BB 00C514 70331768AB53 80C6 1BFB0733ABBC 64FBBA 006BE6 241AEC 20A997BB 08F9F7A9FAF7AAF833176AA7F6 38F8EB 2AF9F7A9 | 21122212111122111121212121 | Change: Call Meas  MOV R0.A SEL RB1 MOV R0, 00h MOV R2, 00h MOV R3, 00h SEL RB0 Work: CALL Meas CPL A.INC A ADD A,R0 MOV R3,A ANL A,80H.JZ Mul.MOV A,R3 DEC A CPL A MOV R3,A. Mul: MOV R4,64h MOV A,R3 MOV R2, 00h M1: ADD A, R3 JNC M2 INC R2 M2: DJNZ R4, M1 MOV R1, A  CLR C MOV R3,08h M3: MOV A, R1 RLC A MOV R1, A MOV A, R2 RLC A MOV R2, A MOV A, R0 CPL A.INC A ADD A, R2 CPL C JC M4 ADD A, R0 M4: DJNZ R3, М3 MOV A, R1RLC AMOV R1, A | ;Вызов подпрограммы измерения сопротивления резистора ;Запомним в R0 эталонное сопротивление.;Подключим банк регистров RB1.;Обнуление счетчика;Обнуление счетчика;Обнуление счетчика;Подключим банк регистров RB0.;Измерение сопротивления резистора.;Инвертирование содержимого аккумулятора;А=А+1⇒А в дополнительном коде.;Вычитаем из эталонного сопротивления измеренное: А=R0-А.;Спасаем А регистре R3;Проверяем знаковый бит;Если А>0, то переходим на метку Mul;Восстановим содержимое А.;А=А-1;Инвертирование А; А- в прямом коде.;Сохраним содержимое А;В R4- число повторений цикла- 10010;В аккумуляторе- разность между измеренным и эталонным ;сопротивлениями, взятая по модулю.;Обнуление регистра R2;А=А+R3.;Если А+R3≤FFh, то переходим на М2.;Учитываем перенос (А+R3>FFh).;Уменьшаем счетчик цикла. Если R4>0- повторяем цикл.;Копируем аккумулятор в регистр R1;После умножения в регистре R1 будет младший байт;произведения, в R2- старший, т. е. R2R1=⏐Nэт-N⏐\*100.;Разделим содержимое этой пары регистров на содержимой;регистра R0 (эталон).;Обнуляем флаг переноса.;Счетчик цикла.;ВА- младший байт произведения.;Циклический сдвиг влево через флаг переноса.;R1=А;ВА- младший байт произведения.;Циклический сдвиг влево через флаг переноса;R2=А;ВА- эталонное вопротивление (делитель).;Инвертирование содержимого аккумулятораА=А+1⇒ А в дополнительном коде;А=R2-А; Инвертирование флага переноса;Если флаг С установлен, переходим на М4.;А=А+ R0;Уменьшаем счетчик цикла. Если R3>0, то повторяем цикл. ;А=R1;циклический сдвиг влево через флаг переноса.; В R1- результат деления, т. е. отклонение в процентах измеренного сопротивления от эталонного.;проверка на принадлежность к группе по допуску. |
| 03D03F04104204304404604704904A04C04E04F05005105305405605705805905B05C05E05F060062 | 03 F6E6 49D518C523 063A04 5CF903 FBE6 56D51AC523 443A04 5CD51BC523 243A23 043A0912 0004 0D | 221112221221112221112222222 | P 10:P 5:Mode: | ADDJNCSELIBCSELMOVOUTLJMPMOVADDJNCSTLINCSELMOVOUTLJMPSELINCSELMOVOUTLMOVOUTLINJB0JMP | A. F6hP10RB1R0RB0A, 06hP2, AModeA, R1A, FBhP5RB1R2RB0A, 44hP2, AModeRB1R3RB0A, 24hP2, AA, 04hP2, AA, P1ChangeWork | ;А=А-1010;Если А<1010, то переходим на Р10;Подключаем банк регистров RB1 ;Увеличиваем счетчик резисторов “ВНЕ ДОПУСКА”;Подключаем банк регистров RB0;Устанавливаем 1-й и 2-й биты в единицу;Команда роботу–положить резистор в контейнер “вне допуска”;Пропускаем остальные отклонения;A=R1 ;A=A–5;Если А<5, то переходим на Р5;Подключаем банк регисторов RB1;Увеличиваем счетчик резисторов с отклонением до 10%;Подключаем банк регистров RB0;Устанавливаем в единицу 2–й и 6–й биты;Команда роботу – положить резистор в контейнер “10%”; Пропускаем отклонение 5%; Подключаем банк регистров RB1;Увеличиваем счетчик резисторов с отклонением до 5%; Подключаем банк регистров RB0; Устанавливаем в единицу 2–й и 5–й биты; Команда роботу – положить резистор в контейнер “10%”; Устанавливаем в единицу только 2–й бит;Снимаем все активные сигналы с порта Р2;Читаем порт Р1;Нулевой бит установлен в единицу–команда смены эталона.;Проверяем следующий резистор. |
| 07007207307507707907В07С07Е07D | 23 053A26 739A 008A 0456 790912 7B0883 | 2222222222 | Meas:W1:W2:W3: | MOVOUTLJNT0ANLORLJT1INJB0INSRET | A, 05hP2, AW1P2,00hP2,04hW2A, P1W3A, Bus | ; Устанавливаем в единицу 0–й и 2–й биты; Команда роботу –установить резистор;Ждем установки резистора;На АЦП-сигнал “начать измерение”;Убираем все активные сигналы с порта Р2;Ждем окончания измерения;Ждем выключения режима; “эталонное сопротивление”;Считываем с АЦП значение сопротивления;Возврат в основную программу. |

К полученной программе трудно применить термин “быстродействие”, на это есть несколько причин:

* В программе есть несколько задержек на неопределенное время ( эти задержки определяются роботом);
* Программа имеет сильно разветвленную структуру, а выбор ветви в некоторых случаях зависит от внешних условий;
* Программа представляет собой бесконечный цикл, т. е. программа выполняется с момента включения системы и до выключения.

Для того, чтобы все-таки оценить быстродействие, сделаем некоторые допущения: предположим, что внешние устройства совершенно не затормаживают систему, т. е. если система выдает запрос во внешние устройства, результат приходит мгновенно. Таким образом, будет посчитано быстродействие собственно программы.

Для определенности примем, что система работает в режиме сортировки, эталонное значение уже занесено в память и в измеритель установлен резистор с 10% допуском.

Так как программа зациклена, посчитаем количество машинных циклов за один проход программы:

2+2×10+1×4+2+2+1×4+2+1+2+1+(1+2+1+2)×100+1+1+2+(1×11+2+1+2)×8+1+1+1+2+2+2+1+ 2+2+1+1+1+2+2+2+1+1+1+2×7=812 циклов

К процессору подключен кварцевый резонатор на частоту 6 МГц, следовательно тактовая частота процессора равна 6/3=2 МГц; время одного такта равно 0,5 мкс. Один машинный цикл равен пяти тактам, т. е. 2,5 мкс.

Время выполнения одного цикла программы равно 812×2,5=2030 мкс ≈ 2 мс.

**6. КОНТРОЛЬНЫЙ ПРИМЕР**

Напряжение, подаваемое на АЦП, равно

U=Iэт×R; Iэт = 0,025 А

Входные данные:

Rэт = 51Ом; Rизм = 53 Ом; Uэт = Iэт×Rэт = 0,025 А×51Ом = 1,28 В

С АЦП в память запишется число 128 = 80h, т. е. (R0) = 80h

U= Iэт×Rэт = 0,025 А×53 Ом = 1,33 В

С АЦП в аккумулятор запишеться число 133 = 85h.

Находим модель разности эталонного и измеренного сопротивлений:

⏐Rэт –Rизм ⏐=⏐128-133⏐=⏐-5⏐=5

Отклонение в процентах находиться по формуле:

и именно по этой формуле работает написанная выше программа.

Умножаем разность на 100:

⏐Rэт –Rизм ⏐×100=500

Делим полученное число на Rэт/800/102 = 3, т. к. деление целочисленное.

Итак, в программе получается, что Rизм имеет отклонение 3% от



номинала. На самом деле Rизм имеет отклонение

Число, полученное программой, и число, рассчитанное непосредственно, достаточно близки друг к другу.

Система поместит данный резистор в контейнер с допуском 5%.

**ЗАКЛЮЧЕНИЕ**

В процессе работы была разработана система, способная рассортировать партию резисторов по допускам 5, 10% и брак. Система также подсчитывает число резисторов каждого допуска.

В устройстве предусмотрена возможность смены эталонного сопротивления, записанного в памяти, при смене партии резисторов.

При каждом попадании резистора с отклонением более 10% система подает световой сигнал.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Микропроцессоры. В 3–х кн. Кн. 1. Архитектура и проектирование микро–ЭМВ. Организация вычислительных процессов: Учебник для вузов. Под редакцией Л.Н. Преснухина. – М.: Высшая школа, 1986г.
2. Калабеков Б.А. Микропроцессоры и их применение в системах передачи и обработки сигналов: Учебное пособие для вузов. – М.: Радио и связь, 1988г.
3. В.Н. Пильщиков. Программирование на языке ассемблер IBM PC. – М.: Диалог МИФИ, 1994г.