

УДК
621.398
Е-172

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

МОСКОВСКИЙ ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)

Ю.Н. ЕВЛАНОВ, В.А. НОВИКОВ, А.А. ШАТОХИН

ПРИМЕНЕНИЕ ОДНОКРИСТАЛЬНОГО МИКРОКОНТРОЛЛЕРА 80С552
В ИЗМЕРИТЕЛЬНОЙ ТЕХНИКЕ

Лабораторные работы

Методическое пособие по курсу
«Схемотехника и программное обеспечение электронных средств измере-
ний»
для студентов, обучающихся по направлению "Информатика и вычислитель-
ная техника"

УДК
621.398
Е-172
УДК: 621.398.726(072)(075.8)

Утверждено учебным управлением МЭИ
Подготовлено на кафедре Информационно-измерительной техники МЭИ

Рецензент: канд. техн. наук, доц. Н.А. Серов

Евланов Ю.Н., Новиков В.А., Шатохин А.А. Применение однокристалльного микроконтроллера 80C552 в измерительной технике: Лабораторные работы. Методическое пособие по курсу «Схемотехника и программное обеспечение электронных средств измерений». - М.: Издательство МЭИ, 2002. – 20 с.

Представлены четыре лабораторные работы: «Управление цифровым отсчетным устройством», «Таймер», «Тестирование интерфейсного устройства», «Ввод с клавиатуры». Описание каждой работы включает задание, методические указания и контрольные вопросы. В приложении приведена инструкция по использованию лабораторного модуля КИТ-552, выполненного на основе однокристалльного микроконтроллера 80C552.

Продолжительность лабораторных занятий для работ №1–3 – 4 часа, для работы № 4 – 8 часов.

Методическое пособие предназначено для студентов старших курсов специальности «Вычислительные машины, комплексы, системы и сети» (специализация «Вычислительно-измерительные системы»), изучающих дисциплину «Схемотехника и программное обеспечение электронных средств измерений».

Учебное издание

Юрий Николаевич Евланов
Виктор Александрович Новиков
Александр Алексеевич Шатохин

ПРИМЕНЕНИЕ ОДНОКРИСТАЛЬНОГО МИКРОКОНТРОЛЛЕРА 80C552
В ИЗМЕРИТЕЛЬНОЙ ТЕХНИКЕ

Лабораторные работы
Методическое пособие по курсу
«Схемотехника и программное обеспечение электронных средств измерений»
для студентов, обучающихся по направлению "Информатика и вычислительная техника"

Редактор издательства Е.Н. Касьянова
ЛР № 020528 от 05.06.97

Темплан издания МЭИ 2002(1), метод.

Подписано к печати 08.05.02

Формат 60×84/16

Печ.л. 1,25

Тираж 50 Изд.№ 9 Заказ

Издательство МЭИ, 111250, Москва, Красноказарменная, д.14

© Московский энергетический институт, 2002

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	3
ПРЕДИСЛОВИЕ.....	4
ЛАБОРАТОРНАЯ РАБОТА №1	5
<i>Методические указания.....</i>	<i>6</i>
<i>Контрольные вопросы</i>	<i>6</i>
ЛАБОРАТОРНАЯ РАБОТА №2	8
<i>Методические указания.....</i>	<i>8</i>
<i>Контрольные вопросы</i>	<i>9</i>
ЛАБОРАТОРНАЯ РАБОТА №3	10
<i>Методические указания.....</i>	<i>10</i>
<i>Контрольные вопросы</i>	<i>11</i>
ЛАБОРАТОРНАЯ РАБОТА №4	13
<i>Методические указания.....</i>	<i>14</i>
<i>Контрольные вопросы</i>	<i>16</i>
ЛИТЕРАТУРА	19
ПРИЛОЖЕНИЕ.....	20

ПРЕДИСЛОВИЕ

Лабораторный практикум «Применение однокристалльного микроконтроллера 80С552 в измерительной технике» предназначен для выработки у студентов специализации "Вычислительно-измерительные системы" практических навыков проектирования функциональных узлов средств измерений на основе однокристалльных микроконтроллеров (ОМК). Необходимо подчеркнуть, что все лабораторные работы связаны с использованием реальных устройств (ОМК, программной памяти, цифрового отсчетного устройства и т.д.), что позволяет приблизить учебный процесс к настоящему проектированию.

Основой практикума являются специальные курсы, читаемые студентам указанной специализации, а также методическое пособие [1]. Лабораторные работы рассчитаны на выполнение их фронтальным способом по единой методике в последовательности, соответствующей их нумерации. При этом предполагается написание студентами прикладных программ на языке ассемблера, создание с помощью инструментальных программ (ассемблера и редактора связей) загружаемых программных файлов, отладку и испытание созданных программных средств с помощью лабораторного модуля КИТ-552. В качестве пособия, содержащего подробное описание системы команд МК семейства МК51 и примеры программирования, может быть рекомендован справочник [2].

В заданиях и методических указаниях лабораторных работ для удобства изложения те или иные программы, подпрограммы, переменные обозначены символическими именами, применение которых студентами необязательно.

Перед выполнением работ необходимо ознакомиться с содержащейся в Приложении инструкцией по использованию лабораторного модуля КИТ-552.

Лабораторная работа №1

УПРАВЛЕНИЕ ЦИФРОВЫМ ОТСЧЕТНЫМ УСТРОЙСТВОМ

Задание

1. Создать программный модуль *display.asm*. Написать и включить в него подпрограмму *LoadDisp*, обеспечивающую загрузку в ЦОУ LDM205AR 40-разрядного «сегментного» кода из 5-байтного буфера, расположенного во внутреннем ОЗУ ОМК; начальный адрес буфера передается подпрограмме через регистр R0.

2. Создать программный модуль *test_1.asm*. Написать и включить в него циклическую тестовую программу *Test_1*, каждый цикл которой должен состоять из: 1) загрузки в 5-байтный буфер *bufDisp* тестового кода x_i ($i = 1, \dots, n$); 2) вызова подпрограммы *LoadDisp*; 3) задержки T_3 .

Значения x_i , n , T_3 указываются преподавателем.

3. С помощью ассемблера получить объектные коды модулей *test_1* и *display* и, объединив их посредством редактора связей, создать загружаемый файл *test_1.tsk*.

Проверить работу подпрограммы *LoadDisp*. При необходимости выполнить отладку.

4. Написать подпрограмму *DispNum*, обеспечивающую вывод на ЦОУ LDM205AR 5-разрядного числа, представленного в десятичной или шестнадцатеричной системе счисления; коды разрядов этого числа должны находиться в 5-байтном буфере, расположенном во внутреннем ОЗУ ОМК; начальный адрес буфера передается подпрограмме через регистр R0; подпрограмма не должна изменять содержимое буфера. Включить подпрограмму *DispNum* в модуль *display.asm*.

5. Создать программный модуль *test_2.asm*. Написать и включить в него циклическую тестовую программу *Test_2*, каждый цикл которой должен состоять из: 1) загрузки в 5-байтный буфер *bufNum* тестового кода y_i ($i = 1, \dots, m$); 2) вызова подпрограммы *DispNum*; 3) задержки T_3 .

Значения y_i , m , T_3 указываются преподавателем.

6. С помощью ассемблера получить объектные коды модулей *test_2* и *display* и, объединив их посредством редактора связей, создать загружаемый файл *test_2.tsk*.

Проверить работу подпрограммы *DispNum*. При необходимости выполнить отладку.

7. Составить отчет о работе, включающий задание, схемы алгоритмов и листинги отлаженных программ.

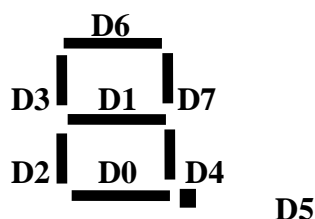
Методические указания

1. Цифровое отсчетное устройство (ЦОУ) LDM205AR представляет собой модуль, включающий контроллер и 5 светодиодных семисегментных индикаторов.

Отображаемая информация последовательно вводится в сдвиговый регистр контроллера через вход DIN; горящему сегменту соответствует логический ноль; синхронизация – по фронту сигнала, поданного на вход CLK. Первым передается старший бит (D7) младшего разряда ЦОУ. После ввода последнего, 40-го, бита (D0 старшего разряда ЦОУ), данные переписываются в выходной регистр в момент возникновения фронта сигнала на входе LOAD.

Входы DIN, CLK, LOAD соединены, соответственно, с выводами P1.0, P1.1, P1.2 микроконтроллера.

Схема соответствия сегментов управляющим битам:



2. Кодировка символов шестнадцатеричного числа, знака «минус» и пробела:

Символ	`0`	`1`	`2`	`3`	`4`	`5`	`6`	`7`	`8`
Код	0	1	2	3	4	5	6	7	8

Символ	`9`	`A`	`b`	`c`	`d`	`E`	`F`	`-`	` `
Код	9	10	11	12	13	14	15	16	17

Если после некоторого разряда числа должна следовать десятичная точка, то в соответствующем байте *bufNum* старший бит D7 устанавливается в состояние логической единицы.

Контрольные вопросы

1. Запишите шестнадцатеричные коды, которые необходимо разместить в *bufDisp*, чтобы получить на ЦОУ следующие символы: «8.8888».

2. Какие команды ОМК целесообразно использовать для формирования входных сигналов ЦОУ DIN, CLK, LOAD?

3. Какие команды ОМК целесообразно использовать для создания программы задержки $T_3 = (0,5 \pm 0,1)$ с?

4. Какой метод преобразования двоичного кода символа, помещенного в *bufNum*, в его «сегментный» код Вы использовали в данной работе?

Лабораторная работа №2

ТАЙМЕР

Задание

1. Создать программный модуль *timer.asm*. Написать и включить в него подпрограмму *Timer*, обеспечивающую выработку с заданной погрешностью δ_t сигналов – меток времени, а также отображение на ЦОУ с помощью подпрограммы *DispNum* времени t , отсчитанного от момента пуска программы.

Диапазон значений t – (00 мин. 00,0 с .. 59 мин. 59,9 с); $|\delta_t| \leq (0,01..0,1)\%$ (по указанию преподавателя).

2. С помощью ассемблера получить объектный код модуля *timer* и, объединив его посредством редактора связей с объектным кодом модуля *display*, создать загружаемый файл *timer.tsk*.

Проверить работу программы *Timer*. При необходимости выполнить отладку.

3. Экспериментально определить значение погрешности δ_t . Сравнить результаты расчета и опыта. В случае невыполнения требования задания к погрешности скорректировать программу *Timer* и повторить эксперимент.

4. Составить отчет о работе, включающий задание, расчет погрешности δ_t , схемы алгоритмов и листинги отлаженных программ.

Методические указания

1. Программа *Timer* должна обеспечивать отображение текущего времени (отсчитанного от момента пуска) в формате ММ.СС.с, где ММ, СС, с – соответственно, минуты, секунды и десятые доли секунды.

Кроме того, программа *Timer* должна вырабатывать метки времени - импульсы на выводе P1.3 с периодом следования 10 мс и длительностью (100...500) мкс.

При расчете погрешности δ_t отклонением реального значения тактовой частоты ОМК от номинального можно пренебречь.

Для формирования меток времени рекомендуется воспользоваться одним из следующих способов:

а) использование прерываний от таймера-счетчика Т/С0 (Т/С1), настроенного на работу в качестве 8-разрядного таймера с автоматической перезагрузкой; в данном случае для получения необходимого значения периода меток времени потребуется дополнительный программный таймер;

б) использование прерываний от таймера-счетчика T2, настроенного на работу в качестве 16-разрядного таймера с автоматической перезагрузкой.

2. Подпрограмма *DispNum* входит в состав модуля *display*, созданного в результате выполнения лабораторной работы №1.

3. Для проверки работоспособности программы *Timer* можно воспользоваться наручными часами с секундомером. При этом следует одновременно запустить проверяемую программу и секундомер; затем, по истечении (10...15) минут сравнить показания и оценить (грубо) погрешность δ_t .

4. Для точного определения погрешности δ_t необходимо измерить цифровым частотомером, погрешность которого достаточно мала, период меток времени T_m , после чего рассчитать погрешность:

$$\delta_t = 100\% (10\text{мс} - T_m) / 10\text{мс} .$$

Контрольные вопросы

1. Какие значения кода предварительной установки таймера-счетчика T/C0 (T/C1), настроенного на работу в качестве 8-разрядного таймера с автоматической перезагрузкой, можно использовать для получения требуемых меток времени?

2. Какой код предварительной установки таймера-счетчика T2, настроенного на работу в качестве 16-разрядного таймера с автоматической перезагрузкой, необходимо использовать для получения требуемых меток времени?

3. Чем определяется систематическая составляющая погрешности δ_t ?

4. Чем определяется случайная составляющая погрешности δ_t ?

5. Какую оценку можно дать погрешности δ_t , если по истечении 10 минут после одновременного запуска программы *Timer* и секундомера их показания совпали с точностью до секунды, а погрешность секундомера пренебрежимо мала?

Лабораторная работа №3

ТЕСТИРОВАНИЕ ИНТЕРФЕЙСНОГО УСТРОЙСТВА

Задание

1. Создать программный модуль *test_com.asm*. Написать и включить в него программу *TestCom*, обеспечивающую определение технического состояния интерфейсного устройства, предназначенного для связи ОМК с компьютером.

2. Написать подпрограмму *DispRes*, обеспечивающую вывод на ЦОУ сообщений о результатах диагностирования. Включить написанную подпрограмму в ранее созданный модуль *display.asm*.

3. С помощью ассемблера получить объектные коды модулей *test_com* и *display* и, объединив их посредством редактора связей, создать загружаемый файл *test_com.tsk*.

Проверить работу программы *TestCom* для следующих состояний интерфейсного устройства: 1) неисправен передатчик ОМК; 2) имеется обрыв в канале связи; 3) интерфейсное устройство исправно; 4) в канале связи возникают искажения.

При необходимости выполнить отладку.

4. Составить отчет о работе, включающий задание, схему алгоритма и листинг отлаженной программы.

Методические указания

1. Интерфейсное устройство, предназначенное для связи ОМК с компьютером, включает встроенный в ОМК универсальный асинхронный приемопередатчик (УАПП), преобразователь уровней ТТЛ/КМОП ↔ RS-232C и 9-контактный разъем для подключения соединительного кабеля.

2. В инициализирующей части программы *TestCom* необходимо предусмотреть настройку УАПП для работы в режиме 1 со скоростью 19,2 кБод.

3. С целью проверки программы в модуль *test_com.asm* нужно включить подпрограмму обработки прерываний от УАПП, сбрасывающую флаг передатчика.

4. Тестовый цикл должен содержать:

а) загрузку в буфер передатчика кода 55H;

б) ожидание установки флага передатчика TI с контролем времени;

в) проверку флага приемника RI и содержимого буфера приемника;

г) передачу через аккумулятор результирующего кода подпрограмме *DispRes*:

- 0 – устройство исправно,
- 1 – неисправен передатчик ОМК,
- 2 – обрыв в канале связи,
- 3 – искажения в канале связи.

5. Подпрограмма *DispRes* в зависимости от кода, помещенного в аккумулятор при ее вызове, выводит на ЦОУ соответствующее сообщение.

Результирующий код	Текст сообщения
0	PASS
1	Err_1
2	Err_2
3	Err_3
>3	-----

Подпрограмме помимо результирующего кода необходимо также передать через регистр R0 начальный адрес 5-байтного рабочего буфера, расположенного во внутреннем ОЗУ ОМК.

6. Проверку программы *TestCom* следует проводить в четыре этапа:

1) временно (только для этого этапа проверки) разрешить прерывания от приемопередатчика; результатом работы программы должно быть сообщение "Err_1";

2) после загрузки программы в ОЗУ КИТ-552 отключить интерфейсный кабель от его платы и запустить программу; при этом на ЦОУ должно появиться сообщение "Err_2";

3) соединить перемычкой гнезда 2 и 3 разъема последовательного порта (выполняет преподаватель); при этом на ЦОУ должно появиться сообщение "PASS";

4) включить между гнездами 2 и 3 разъема последовательного порта вместо перемычки имитатор линии задержки (выполняет преподаватель); при этом на ЦОУ должно появиться сообщение "Err_3".

7. В работе используется модуль *display*, созданный в результате выполнения лабораторной работы №1.

Контрольные вопросы

1. Какие операции необходимы для требуемой в данной лабораторной работе настройки УАПЧ ОМК?

2. Как рассчитывается значение максимально возможного времени ожидания установки флага передатчика $T_{П, \max}$?

3. Почему, по Вашему мнению, в качестве тестового кода в лабораторной работе используется код 55Н?

4. Как в лабораторной работе имитируется неисправность передатчика ОМК?

Лабораторная работа №4

ВВОД С КЛАВИАТУРЫ

Задание

1. Создать программный модуль *keyboard.asm*. Написать подпрограмму *CheckKeys*, обеспечивающую выявление нажатия на любую из 16 кнопок клавиатуры и передающую информацию о нажатиях в двухбайтный буфер *bufKeys*, расположенный в битовой области внутреннего ОЗУ ОМК. Включить подпрограмму *CheckKeys* в состав модуля *keyboard.asm*.

2. Создать программный модуль *test_1.asm*. Написать тестовую программу *Test_1*, которая должна содержать: 1) разрешение прерываний через каждые 10 мс от одного из таймеров-счетчиков ОМК, работающего в режиме таймера; 2) обработку прерывания – вызов подпрограммы *CheckKeys*; 3) основной цикл, включающий анализ состояния *bufKeys* и вывод на ЦОУ символа нажатой кнопки с помощью подпрограммы *DispNum*. Включить программу *Test_1* в состав модуля *test_1.asm*.

3. С помощью ассемблера получить объектные коды модулей *test_1* и *keyboard* и, объединив их, а также объектный код модуля *display* посредством редактора связей, создать загружаемый файл *test_1.tsk*.

Проверить работу подпрограммы *CheckKeys*. При необходимости выполнить отладку.

4. Экспериментально определить минимально возможное время реакции программы на нажатие при условии сохранения нечувствительности к «дребезгу» контактов.

5. Написать подпрограмму *InpNum*, входным параметром которой является передаваемый через регистр R1 начальный адрес 5-байтного буфера, расположенного во внутреннем ОЗУ ОМК. Подпрограмма должна возвращать в заданном буфере коды введенных символов. Включить подпрограмму *InpNum* в состав модуля *keyboard.asm*

6. Создать программный модуль *test_2.asm*. Написать программу *Test_2*, которая должна включать: 1) ввод и сохранение во внутреннем ОЗУ ОМК до 10 наборов кодов символов; 2) просмотр введенных наборов символов; 3) контроль корректности введенного набора символов (по указанию преподавателя). Включить программу *Test_2* в состав модуля *test_2.asm*.

7. С помощью ассемблера получить объектные коды модулей *test_2* и *keyboard* и, объединив их, а также объектный код модуля *display* посредством редактора связей, создать загружаемый файл *test_2.tsk*.

Проверить работу программы *Test_2*. При необходимости выполнить отладку.

8. Составить отчет о работе, включающий задание, схемы алгоритмов и листинги отлаженных программ.

Методические указания

1. Используемая в лабораторной работе клавиатура имеет матричную структуру, представленную на следующем рисунке.

P4.0	P4.1	P4.2	P4.3	
↓	↓	↓	↓	
1	2	3	A	→ P4.4
4	5	6	B	→ P4.5
7	8	9	C	→ P4.6
*	0	#	D	→ P4.7

Матрица соединена с портом 4 ОМК. Столбцы матрицы соединены с выводами P4.0...P4.3 (использующимися как выходы), а строки – с выводами P4.4...P4.7 (использующимися как входы).

Сканирующий сигнал (логический ноль) подается на один из столбцов, затем выполняется считывание сигналов строк. Нажатие каких-либо кнопок из выбранного столбца приводит к считыванию логических нулей в соответствующих строках. Поочередно просканировав все четыре столбца, можно определить состояние всех кнопок панели.

Используемый в лабораторной работе ассемблер *x8051.exe* не рассчитан на обращение к порту 4 (адрес соответствующего регистра специальных функций СОН). Для того чтобы отключить в программе *x8051.exe* контроль правильности адресации внутреннего ОЗУ, нужно воспользоваться директивой **RAMCHK OFF**.

2. Алгоритм выявления нажатия кнопки, рекомендуемый для использования в подпрограмме *CheckKeys*:

а) периодическое (с периодом T , задаваемым вызывающей программой) определение x_i – состояния кнопки в момент времени $t_i=t_0+iT$ (логический ноль соответствует нажатой кнопке), $i=0,1,2,\dots$, t_i – время запуска программы;

б) определение логической функции y_i - флага нажатия:

$$y_i := (x_i \text{ xor } x_{i-1}) \text{ and not } (x_i);$$

в) запоминание текущего состояния кнопки в буфере предыдущего состояния:

$$x_{i-1} := x_i.$$

3. При вводе информации посредством кнопок возможны искажения из-за вибрации («дребезга») контактов при нажатии и отпускании. Продолжительность «дребезга» $T_{др}$ в зависимости от конструктивных особенностей кнопок и материала контактов может принимать значения в диапазоне (0,01...0,1)с.

Следствием «дребезга» может быть многократная установка флага нажатия u_i при однократном воздействии на кнопку. Этого нежелательного явления можно избежать, если выбрать значение периода вызова подпрограммы *CheckKeys* так, чтобы выполнялось следующее неравенство: $T > T_{др}$. При этом сигнал, поступающий от кнопки, будет профильтрован, и ввод с клавиатуры станет нечувствительным к «дребезгу» контактов.

4. Помимо 10 цифровых кнопок в подпрограмме *InpNum* используются и остальные 6 кнопок клавиатуры, которым придаются следующие функции:

- A – ESC (отказ),
- B – BS (стирание последнего набранного символа),
- C – SPACE (пробел),
- D – ENT (ввод),
- * – POINT (десятичная точка),
- # – MINUS (знак минус).

5. Интерфейс пользователя подпрограммы *InpNum*:

- а) сначала все разряды ЦОУ гасятся;
- б) при нажатии на любую из цифровых кнопок, а также на кнопки SPACE и MINUS, соответствующий символ появляется в первом, считая слева, свободном разряде ЦОУ;
- в) при нажатии кнопки POINT включается сегмент-точка в последнем, считая слева, заполненном разряде ЦОУ;
- г) с помощью кнопки BS последний введенный символ стирается;
- д) при нажатии на любую из кнопок ESC и ENT осуществляется возврат из подпрограммы, причем флаг нажатия кнопки ESC не сбрасывается.

Схема алгоритма подпрограммы *InpNum* представлена на рис.1. На рисунке использованы следующие обозначения:

Digits – переменная булевского типа, определяемая логическим выражением $Digits = Digit_0 \text{ or } Digit_1 \text{ or } .. Digit_9$, где $Digit_0, Digit_1, .. Digit_9$ – флаги нажатия цифровых кнопок;

Minus, Space, Point, Ent, Esc – соответственно, флаги нажатия кнопок MINUS, SPACE, POINT, ENT, ESC;

$A_0...A_4$ – адресное пространство во внутреннем ОЗУ ОМК, занимаемое 5-байтным буфером, предназначенным для вводимого числа;

N_j ($j=0,1,..9$), N_{Space} , N_{Minus} – коды символов, соответствующих кнопкам.

6. Интерфейс пользователя программы *Test_2*:

- а) исходное состояние – «ожидание», при этом на ЦОУ выводится порядковый номер j (от 0 до 9) очередного набора символов: ` - j - - `;

- б) при нажатии на кнопку ENT вызывается подпрограмма *InpNum*;
- в) при возврате из подпрограммы *InpNum* осуществляется переход в состояние ожидания и, если не была нажата кнопка ESC, полученный набор кодов введенных символов запоминается в кольцевом буфере наборов;
- г) при нажатии на любую цифровую кнопку на ЦОУ выводится набор символов с соответствующим порядковым номером;
- д) при отображении какого-либо набора из буфера возможен просмотр любого набора (цифровые кнопки) и возврат в состояние ожидания (кнопка ESC).

Схема алгоритма программы *Test_2* представлена на рис.2.

7. Подпрограмма *DispNum* входит в состав модуля *display*, созданного в результате выполнения лабораторной работы №1.

8. Кодировка символов, соответствующих кнопкам клавиатуры, содержится в методическом указании 2 лабораторной работы №1.

Контрольные вопросы

1. Какова аппаратная реализация ввода с клавиатуры, используемая в данной лабораторной работе?
2. Какие команды ОМК необходимо использовать, чтобы считать состояние первой строки клавиатуры?
3. Какие логические операции Вы применили для выявления нажатия кнопки?
4. Что такое «дребезг» контактов и как он влияет на ввод с клавиатуры?
5. Какой способ исключения влияния «дребезга» на ввод с клавиатуры Вы использовали в данной лабораторной работе?
6. Чем ограничено снизу время реакции на нажатие кнопок?

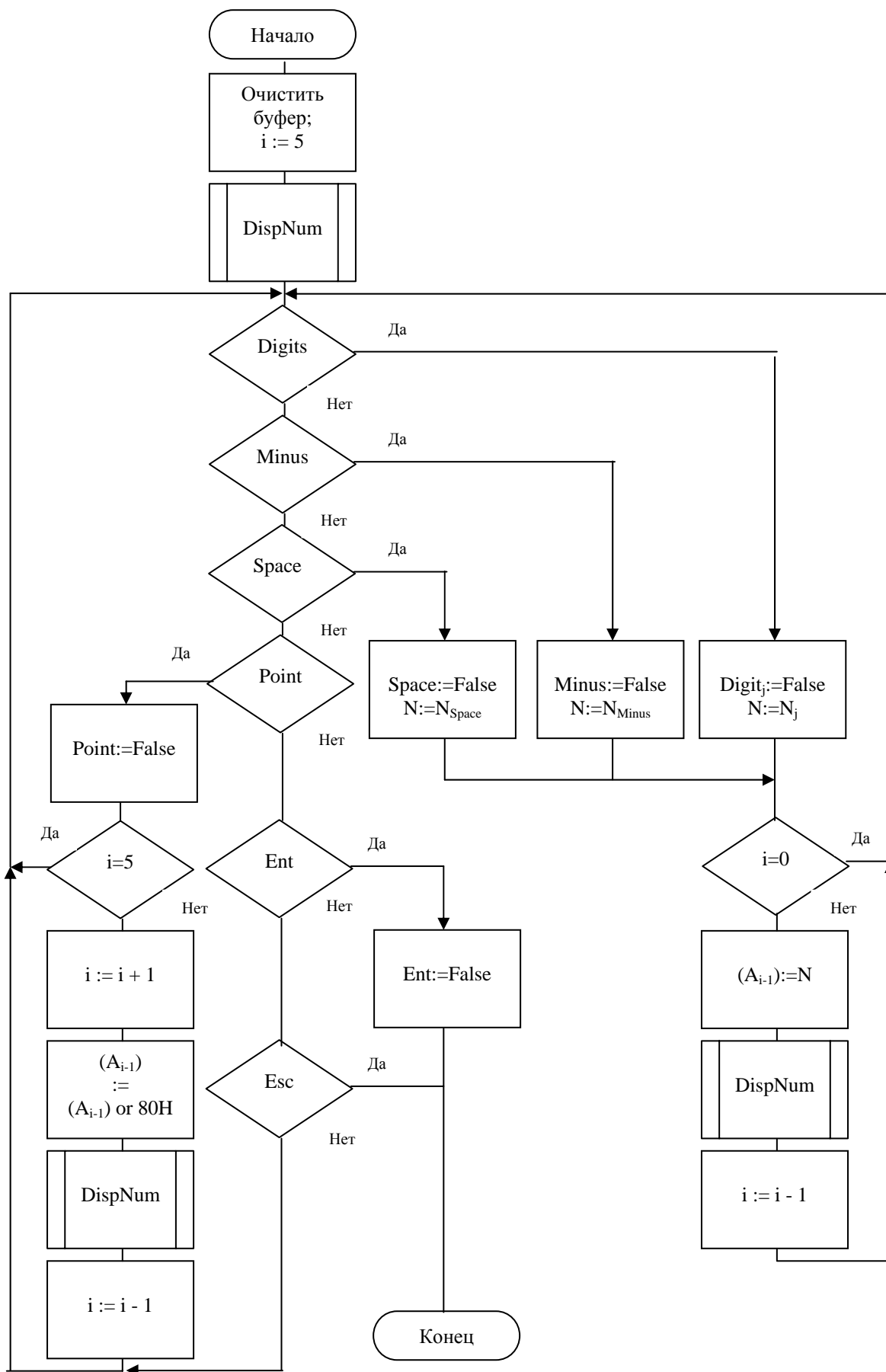


Рис. 1. Схема алгоритма подпрограммы *InpNum*

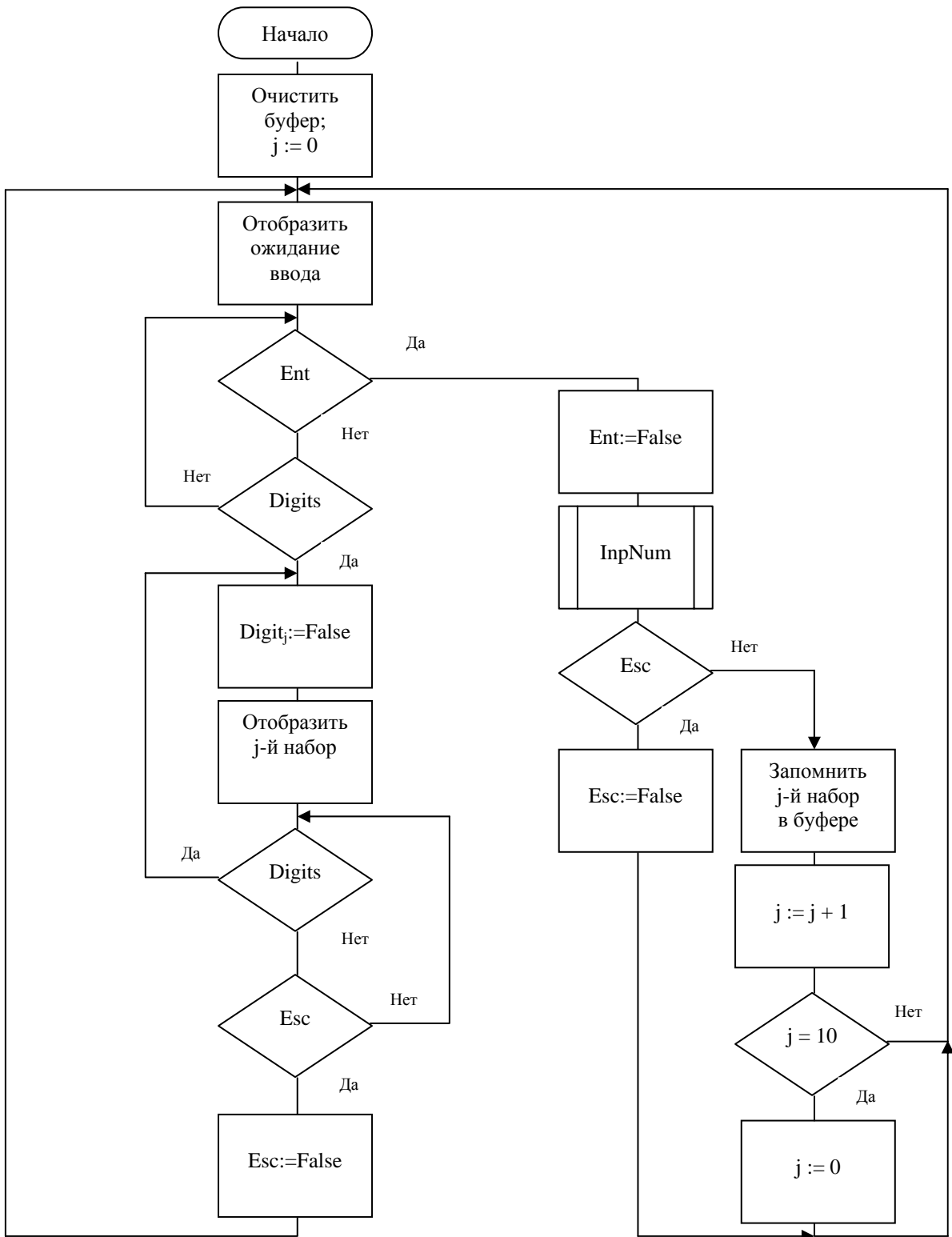


Рис. 2. Схема алгоритма программы *Test_2*

ЛИТЕРАТУРА

1. **Евланов Ю.Н., Новиков В.А., Шатохин А.А.** Однокристалльный микроконтроллер 80C552: Методическое пособие по курсу «Схемотехника и программное обеспечение электронных средств измерений». – М.: Издательство МЭИ, 2001. – 60 с.
2. **Однокристалльные микроЭВМ/А.В. Боборыкин, Г.П. Липовецкий, Г.В. Литвинский и др.** – М.: БИНОМ, 1994. – 400 с.

ПРИЛОЖЕНИЕ

ИНСТРУКЦИЯ

по использованию лабораторного модуля КИТ-552

1. С дискеты, полученной у преподавателя, скопируйте в Ваш рабочий каталог программу **loadprog.exe**.

2. Используя инструментальные программные средства (текстовый редактор, ассемблер, редактор связей), создайте загружаемый программный файл (например, под именем *myprog.tsk*).

3. Подключите КИТ-552 с помощью соединительного кабеля к свободному СОМ-порту компьютера. Подайте напряжение +5В на плату КИТ-552. При правильной работе должен загореться индикатор «Master», указывающий на готовность КИТ-552 к приему программы пользователя.

4. Вызовите к исполнению программу **loadprog.exe**.

4.1. Программа **loadprog.exe** предложит имя загружаемого файла и номер СОМ-порта, сохраненные в памяти компьютера после предыдущего исполнения программы. Если требуется изменить имя загружаемого файла и (или) номер СОМ-порта, нажмите клавишу «n» («N»), после чего введите новые параметры; в противном случае нажмите клавишу «y» («Y»).

4.2. После исполнения программы **loadprog.exe** при положительном исходе на дисплее появится сообщение «Выполнено 100%», при отрицательном - сообщение об ошибке. В последнем случае следует обратиться за помощью к преподавателю.

5. Для запуска загруженной программы нажмите кнопку «U/M», при этом загорится индикатор «User» и программа начнет выполняться.

6. Для перезапуска загруженной программы нажмите кнопку «RESET».

7. Для загрузки новой (или скорректированной) программы нажатием кнопки «U/M» переведите КИТ-552 в режим «Master» (при этом ранее записанная в ОЗУ КИТ-552 программа будет стерта), после чего повторите действия, соответствующие п.п. 4, 5.

8. Тактовая частота ОМК $f_{osc} = 11,059$ МГц.

9. При продолжительных паузах в использовании КИТ-552 его следует отключать от сети электропитания во избежание перегрева сетевого адаптера.