

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тверской государственный технический университет»
(ТвГТУ)
Факультет Информационных Технологий
Кафедра ЭВМ

Отчёт к лабораторной работе № 2
по дисциплине: «Проектирование специализированных компьютерных
систем»
на тему: «Команды передачи управления. Организация циклов»

Выполнил: студент группы
Б.ИВТ.ВМКСС-19.03
Макаров М.К.
Принял: Филиппов Р.Н.

Тверь,
2023

Цель работы

Изучение организации пространства памяти программ микроконтроллера КМ1816ВЕ51, программных средств управления ходом выполнения программы, приобретение навыков программирования циклических алгоритмов в кодах микроконтроллера.

Задание

Вариант №18

НОМЕР	ADR1	ADR2	N	ADR3
14	4D3	1F2	A	5B

Часть 1:

Пусть в памяти программ, начиная с ячейки ADR2, расположена таблица кодов длиной N. Записать в кодах МК К1816ВЕ51 программу, которая выполняет пересылку данного массива в RAM, начиная с адреса ADR3. Программа должна начинаться с ячейки ADR1. Коды данных задать произвольно.

Часть 2:

Собрать схему, в результате работы которой:

- с клавиатуры вводятся числа;
- записываются во внешнюю память МК (пространство XSEG).

Теоретическая часть

Пространство памяти программ CSEG

Для хранения программ и неизменяемых данных в МК 1816ВЕ51 используется логическое однородное линейное пространство памяти CSEG объемом 64Кбайт. Память программ адресуется 16-разрядным счетчиком РС. Младшие 4Кбайт этого пространства соответствует встроенному EPROM

микроконтроллера, остальные 60 Кбайт реализуются внешним относительно МК схемами.

В пространстве CSEG выделяются следующие точки (адреса):

0000H - RESET - стартовый адрес при сбросе системы;

0003H - EXTI0 - внешнее прерывание 0;

000BH - TIMER0 - прерывание таймера/счетчика 0;

0013H - EXTI1 - внешнее прерывание 1;

001BH - TIMER1 - прерывание таймера / счетчика 1;

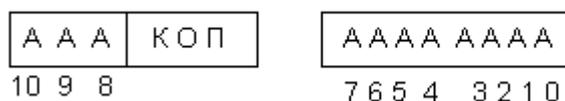
0023H - SINT - прерывание последовательного порта;

002BH - TIMER2 - прерывание таймера 2 (для MCS 52).

В CSEG определены два способа передачи управления:

1. Прямая адресация с помощью 16- разрядного прямого адреса ad16;
2. Относительная адресация, имеющая два варианта: с помощью 8-разрядного смещения куд (целое двоичное со знаком) относительно PC [(A)+PC0] или DPTR [(A)+(DPTR)].

Для двух команд (AJMP, ACALL) предусмотрена страничная адресация в CSEG, с помощью 11-разрядного адреса ad11. В этом случае 8 младших разрядов адреса располагаются во втором байте команды, а 3 старших - в трех старших разрядах первого байта команды:



Номер страницы задается пятью старшими разрядами программного счетчика PC.

Необходимо подчеркнуть также, что в группе пересылки существуют специальные команды MOVC A,@A+DPTR и MOVC A,@A+PC, которые позволяют считывать содержимое памяти программ программ. Как правило эта возможность используется для организации таблиц констант в CSEG.

Команды передачи управления

Группу команд передачи управления образуют команды безусловного перехода, безусловного перехода, команды вызова подпрограмм и команды возврата из подпрограммы.

В большинстве команд используется прямая адресация.

Характеристики команд приведены в следующей таблице:

№	МНЕМОНИКА	КОП	Б	Ц	ОПЕРАЦИЯ
	LJMP ad16	00000010	3	2	(PC) := ad16
	AJMP ad11	aaa00001	2	2	(PC) := (PC) + 2 (PC[0-10]) := ad11
	SJMP rel	10000000	2	2	(PC) := (PC) + 2 (PC) := (PC) + rel
	JMP @A+DPTR	01110011	1	2	(PC) := (A) + (DPTR)
	JZ rel	01100000	2	2	(PC) := (PC) + 2; если (A) == 0 to (PC) := (PC)+rel
	JNZ rel	01110000	2	2	(PC) := (PC) + 2; если (A) != 0 to (PC) := (PC)+rel
	JC rel	01000000	2	2	(PC) := (PC) + 2; если (C) == 1 to (PC) := (PC)+rel
	JNC rel	01010000	2	2	(PC) := (PC) + 2; если (C) == 0 to (PC) := (PC)+rel
	JB bit, rel	00100000	3	2	(PC) := (PC) + 3; если (b) == 1 to (PC) := (PC)+rel
	JNB bit,rel	00110000	3	2	(PC) := (PC) + 3; если (b) == 0 to (PC) := (PC)+rel
	JBC bit, rel	00010000	3	2	(PC) := (PC) + 3; если (b) == 1 to (PC) := (PC)+rel, (b) :=0
	DJNZ Rn, rel	11011rrr	2	2	(PC) := (PC)+ 2, (Rn) := (Rn) - 1; если (Rn) != 0 to (PC) := (PC)+rel
	DJNZ ad, rel	11010101	3	2	(PC) := (PC)+ 3, (ad) := (ad) - 1; если (ad) != 0 to (PC) := (PC)+rel
	CJNZ A, ad, rel	10110101	3	2	(PC) := (PC)+ 3; если (A) != ad, to (PC) := (PC)+rel;

					если (A)<(ad), то (C):=1, иначе (C) := 0;
	CJNE A, #d, rel	10110100	3	2	(PC) := (PC)+ 3; если (A) != #d, то (PC) := (PC)+rel; если (A)<(#d), то (C):=1, иначе (C) := 0;
	CJNZ Rn, #d, rel	10111rrr	3	2	(PC) := (PC)+ 3; если (Rn) != #d, то (PC) := (PC)+rel; если (Rn)<(ad), то (C):=1, иначе (C) := 0;
	CJNZ @Ri, #d, rel	1011011i	3	2	(PC) := (PC)+ 3; если ((Ri)) != #d, то (PC) := (PC)+rel; если ((Ri))<(#d), то (C):=1, иначе (C) := 0;
	LCALL ad16	00010010	3	2	(PC) := (PC)+3, (SP) := (SP) + 1 ((SP)) := (PCL), (SP) := (SP)+1 ((SP)) := (PCH), (PC) := ad16
	ACALL ad11	aaa10001	2	2	(PC) := (PC)+2, (SP) := (SP) + 1 ((SP)) := (PCL), (SP) := (SP)+1 ((SP)):= (PCH), (PC[0-10]):= ad11
	RET	00100010	1	2	(PCH) :=((SP)), (SP) := (SP) + 1, (PCL) := ((SP)), (SP) := (SP) -1
	RETI	00110010	1	2	(PCH) :=((SP)), (SP) := (SP) + 1, (PCL) := ((SP)), (SP) := (SP) -1
	NOP	00000000	1	1	(PC) := (PC) + 1

Таблица 1. Группа команд передачи управления

Практическая часть

Часть 1

Для выполнения данной задачи сначала необходимо добавить на схему микроконтроллер 80C51, в который далее нужно будет загружать код программы в шестнадцатеричном формате. Можно воспользоваться схемой из предыдущей лабораторной работы, просто заменив в ней код на языке ассемблера.

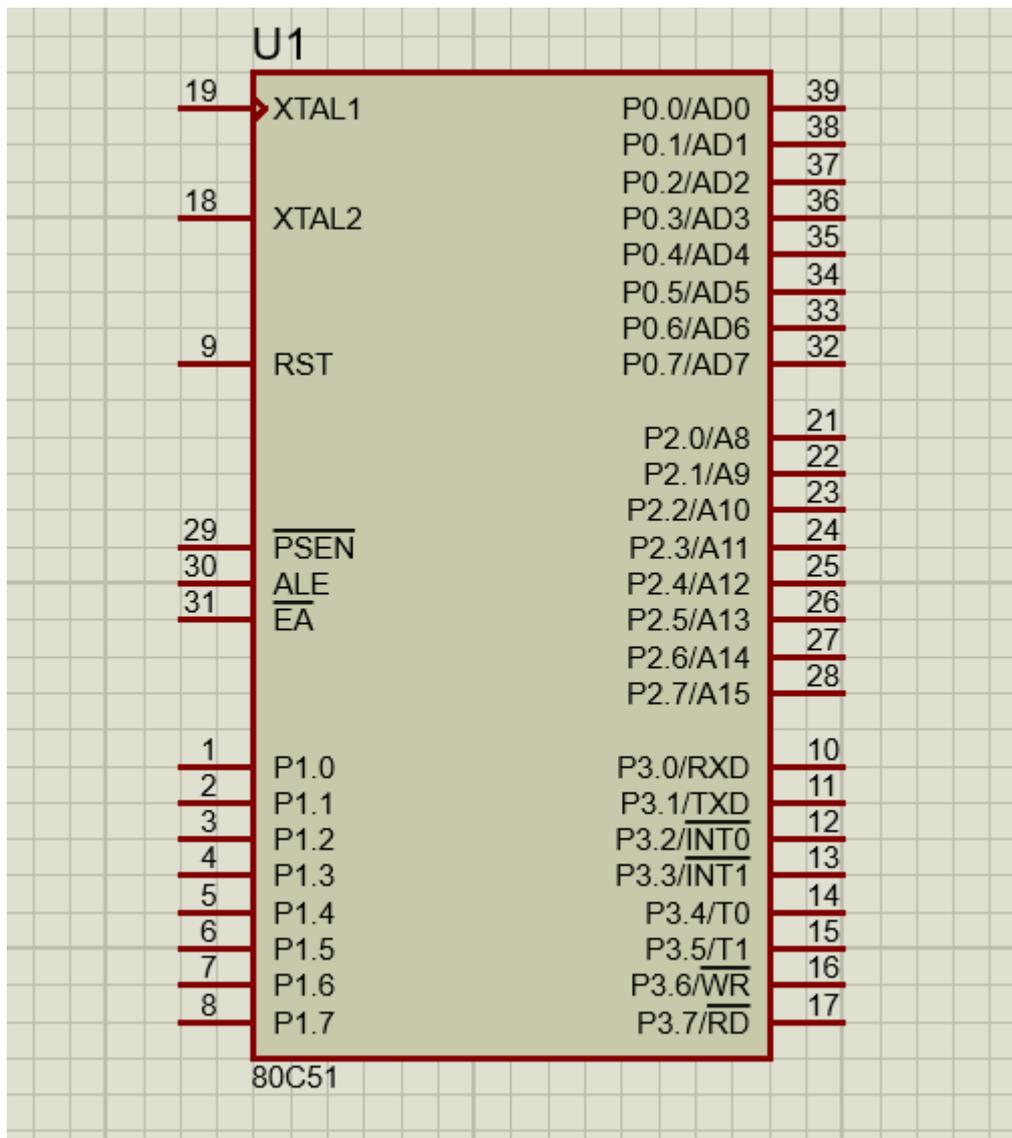


Рисунок 1. Добавление микроконтроллера 80C51 в проект.

Код программы, который выводит строку «Yakovlev»:

```

jmp start
org                1F2H
db                '$Makarov',0
org                4D3H
start:
MOV DPTR,#01F2H
MOV R0,#5BH
MOV R1,#0AH
m2:                MOV A,R1
MOVC A,@A+DPTR
MOV @R0,A
DJNZ R0,m1
m1:                DJNZ R1,m2
NOP
RET
END

```

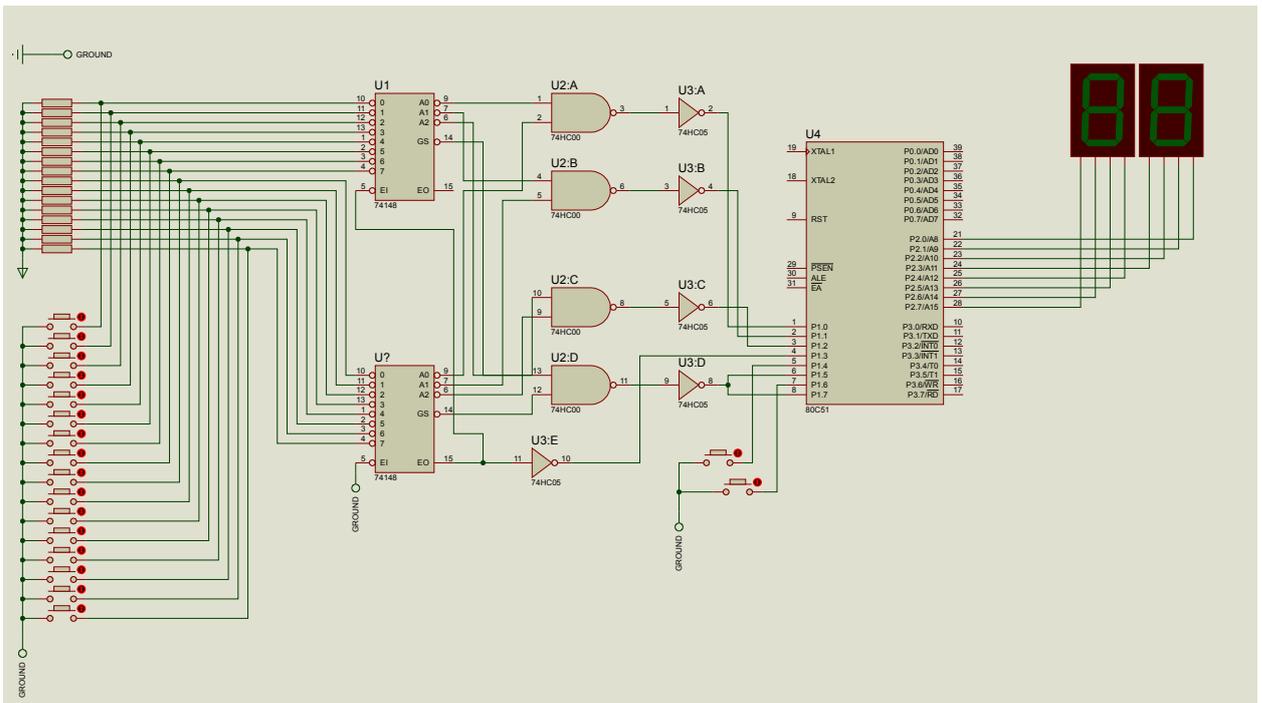



Рисунок 4. Схема клавиатуры с микроконтроллером и семисегментными индикаторами.

Ниже представлен модернизированный код программы предыдущей лабораторной работы, который позволяет выводить числа на индикаторы:

```
ORG 0255h
```

```
start:
```

```
MOV DPTR, #0781
MOV P2, #0000000b
```

```
chk:          JB      p1.5,chk          ;проверка нажатия на число
              SETB   p1.7
```

```
m2:          SWAP   A
```

```
bit0:        JB      P1.0,bit1         ;считывание заданного числа побитно
              ADD   A,#00000001b
```

```
bit1:        JB      P1.1,bit2         ;считывание заданного числа побитно
              ADD   A,#00000010b
```

```
bit2:        JB      P1.2,bit3         ;считывание заданного числа побитно
              ADD   A,#00000100b
```

```
bit3:        JB      P1.3,m1           ;считывание заданного числа побитно
              ADD   A,#00001000b
```

```
m1:          CALL   delay
              CALL   delay
              CALL   delay
```

;ожидание дальнейших действий

```
number:      JB      p1.7,rst
              JMP   m2
```

```
rst:         JB      p1.6,mem          ;сброс введённого числа
              CLR   A
```

```

                JMP      chk          ;возврат на ожидание ввода числа
mem:           JB       p1.4,number  ;запоминание числа
                MOVX    @DPTR,A
                MOV     P2, A
                INC     DPTR
                CLR     A
                JMP     chk
delay:        MOV     R7,#255
ZD0:         MOV     R6,#255
                MOV     R5,#255
ZD1:         DJNZ    R6,ZD1
ZD2:         DJNZ    R5,ZD2
                DJNZ    R7,ZD0
                RET
END

```

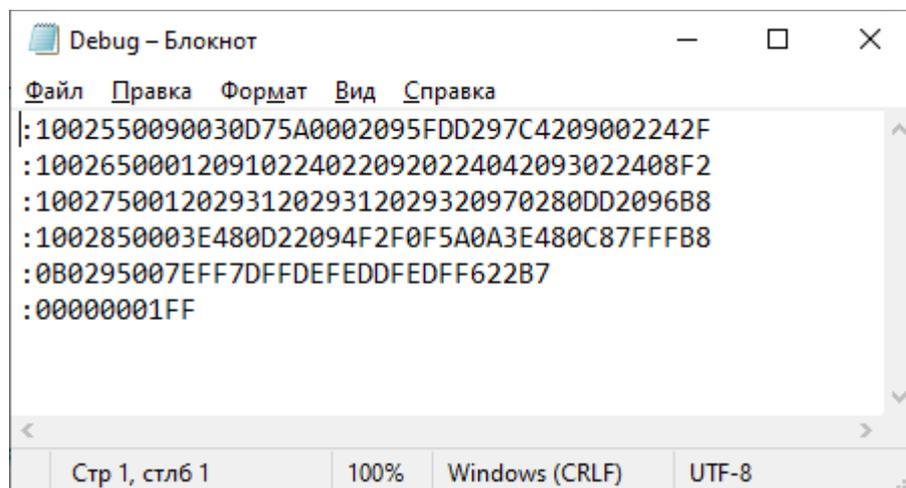


Рисунок 5. Содержимое файла Debug.hex.

Результат ввода и запоминания числа выглядит следующим образом:

8051 CPU Internal (IDATA) Memory - U4										
00	00	00	00	00	00	00	00	00	00	7E 02 00 00
0C	00	00	00	00	00	00	00	00	00	00 00 00 00
18	00	00	00	00	00	00	00	00	00	00 00 00 00
24	00	00	00	00	00	00	00	00	00	00 00 00 00
30	00	00	00	00	00	00	00	00	00	00 00 00 00
3C	00	00	00	00	00	00	00	00	00	00 00 00 00
48	00	00	00	00	00	00	00	00	00	00 00 00 00
54	00	00	00	00	00	00	00	00	00	00 00 00 00
60	00	00	00	00	00	00	00	00	00	00 00 00 00
6C	00	00	00	00	00	00	00	00	00	00 00 00 00
78	00	00	00	00	00	00	00	00	00	

Рисунок 8. Окно ячеек памяти (IDATA).

Вывод

В результате выполнения данной лабораторной работы были изучены организация пространства памяти программ микроконтроллера КМ1816ВЕ51 и программных средств управления ходом выполнения программы. Приобретены навыки программирования циклических алгоритмов в кодах микроконтроллера.

В первой части лабораторной работы была написана простая программа на языке ассемблера для ознакомления с работой микроконтроллера.

Во второй части построена схема шестнадцатеричной клавиатуры с микроконтроллером и двумя семисегментными индикаторами, на которые выводились шестнадцатеричные числа. В микроконтроллер также была записана программа на языке ассемблера, в результате сборки которой можно было наблюдать перемещение введённых данных по регистрам и битам микроконтроллера с помощью инструментов отладки программного средства Proteus, а также их отображение на индикаторах.