



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
**«МИРЭА – Российский технологический университет»**  
**РТУ МИРЭА**

---

Институт информационных технологий  
Кафедра вычислительной техники

**Отчет по лабораторной работе №1**  
**по дисциплине**

**«Разработка и программирование микропроцессорных систем»**

**Тема работы:**  
**«Изучение системы команд микроконтроллера семейства MCS-51»**

**Студент группы:** ИВБО-02-19 А. М. Сосунов

**Преподаватель:** ассистент кафедры ВТ Р. Э. Семенов

Работа выполнена «\_\_\_\_\_» \_\_\_\_\_ 2021 г.

«Зачтено» «\_\_\_\_\_» \_\_\_\_\_ 2021 г.

Москва 2021

## **Цель работы**

1. Изучить общие правила написания программ на ассемблере для семейства MCS-51.
2. Изучить систему команд микроконтроллера 87C52X2: форматы представления данных и команд, способы адресации операндов, команды операций с данными, признаки результата операций, команды операций управления.
3. Приобрести навыки программирования микроконтроллера 87C52X2 в эмуляторе «MCU 8051 IDE».

## **Задание**

В ходе данной лабораторной работы было предложено выполнить следующее задание. Определить адрес наименьшего числа в массиве, расположенном во внутренней памяти микроконтроллера 87C52X2.

## **Ход работы**

Для выполнения данного задания напомним программу на языке ассемблера, в которой инициализируем массив из десяти элементов, расположенных по адресам от 0x20 до 0x29, отдельно выделим три регистра R0, R1 и R2 для вспомогательных операций. Результат программы сформируем в аккумуляторе A.

## **Используемые регистры и ячейки памяти**

Назначение регистров и ячеек памяти, использованных в ходе выполнения работы, приведено в таблице 1.

## **Алгоритм решения задачи**

Опишем алгоритм работы программы по нахождению адреса минимального элемента массива (см. рисунок 1).

Таблица 1 — Назначение регистров

Регистр	Назначение
R0	Адрес текущего элемента массива. В начальный момент содержит адрес начала массива – 32 (0x20).
R1	Адрес текущего минимального элемента массива. В начальный момент содержит адрес первого элемента массива – 32 (0x20)
R2	Счетчик итераций. В начальный момент содержит число элементов массива – 10.
A	Участвует в вспомогательных операциях. Хранит результат работы программы

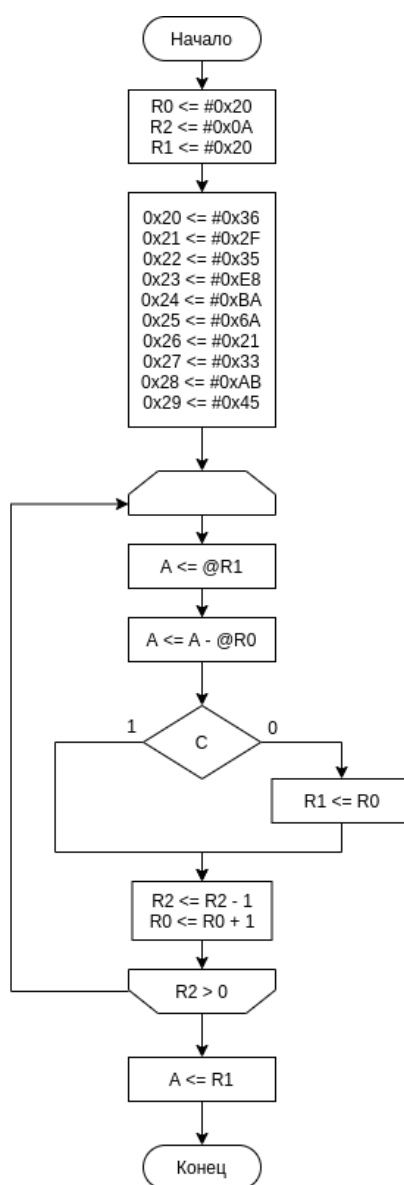


Рисунок 1 — Алгоритм работы программы

## Описание используемых команд

Приведем описание команд, использованных в ходе написания программы.

**Команда MOV @Rn, A**, где  $n = 0..7$

Записывает содержимое регистра-аккумулятора A во внутреннюю память данных по адресу, который содержится в регистре Rn.

**MOV Rn, #data**, где  $n = 0..7$

Записывает в регистр Rn данные #data, которые содержатся непосредственно в команде.

**MOV <direct>, #data**

Пересылка по прямому адресу константы.

**MOV A, Rn**, где  $n = 0..7$

Записывает в регистр-аккумулятор A содержимое регистра Rn.

**SUBB A, Rn**

Команда «вычитание с заемом» вычитает указанную переменную вместе с флагом переноса из содержимого аккумулятора, засылая результат в аккумулятор A.

**DJNZ Rn, <метка>**, где  $n = 0..7$

Команда «декремент и переход, если не равно нулю» выполняет вычитание «1» из указанной ячейки и осуществляет ветвление по вычисляемому адресу, если результат не равен нулю.

**SJMP <метка>**

Команда выполняет относительный переход по адресу в пределах +2 Кслов (4 Кбайт) текущего состояния счетчика команд. В ассемблере вместо относительных операндов используются метки.

## Код программы

Приведем исходный код программы на языке ассемблера (см. листинг 1).

*Листинг 1 — Описание устройства ресинхронизации данных*

```
USING 0 ; Use register bank 0
2 ORG 0h
  Jmp Start ; Start of the program
4 ORG 03h
  LJmp Ext0Srv ; Handler address EXT INT0
```

```

6 ORG 0BH ; Handler address TIMER INT 0
  LJump Time0Srv ;
8 ORG 13H ; Handler address EXT INT 1
  LJump Ext1Srv ;
10 ORG 1BH ; Handler address TIMER INT 1
  LJump Time1Srv ;
12 ORG 23H ; Interrupt handler address from
  LJump RiTiSrv ; Serial port
14 ; --- End of header ----
  Start:
16
  MOV R0, #0x20 ; Address of the current array element
18 MOV R1, #0x20 ; Address of the current minimum element
  MOV R2, #0x0A ; Iteration counter
20
22 MOV 0x20, #0x36 ; Initializing memory
  MOV 0x21, #0x2F
24 MOV 0x22, #0x35
  MOV 0x23, #0xE8
26 MOV 0x24, #0xBA
  MOV 0x25, #0x6A
28 MOV 0x26, #0x21
  MOV 0x27, #0x33
30 MOV 0x28, #0xAB
  MOV 0x29, #0x45
32 Loop1:
34   MOV A, @R1 ;Compare current and minimum elements
    SUBB A, @R0
36   JC NotMin
    MOV A, R0
38   MOV R1, A
  NotMin:
40
    INC R0
42   DJNZ R2, Loop1
    MOV A, R1
44 Loop:
    SJMP Loop
46 ; Interrupt handlers
  Ext1Srv: RETI
48 Time0Srv: RETI
  Time1Srv: RETI
50 RiTiSrv: RETI
  Ext0Srv: RETI
52 RetIE0: RETI
  END

```

## **Вывод**

В ходе данной лабораторной работы нами были рассмотрена и изучена система команд микроконтроллера семейства MCS-51, приобретены навыки программирования микроконтроллера 87C52X2 в эмуляторе «MCU 8051 IDE». Полученные знания были применены для написания программы по нахождению адреса минимального элемента в массиве.