

**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»**

**ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ ПРОГРАММИРОВАНИЯ
НА ЯЗЫКЕ АССЕМБЛЕРА**

САМАРА 2010

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»

**ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ ПРОГРАММИРОВАНИЯ
НА ЯЗЫКЕ АССЕМБЛЕРА**

*Утверждено Редакционно-издательским советом университета
в качестве методических указаний к лабораторному практикуму*

САМАРА
Издательство СГАУ
2010

УДК 621.397.132

Составитель А.В. Данилов

Рецензент канд. техн. наук, доц. Ю.С. Дмитриев

Примеры решения задач программирования на языке ассемблера: метод. указания к лаб. практикуму / сост. *А.В. Данилов.* – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2010. – 16 с.

Приводятся примеры программ, реализующих типовые процедуры обработки данных и управления объектами. Рассмотрены рекомендации к выполнению лабораторных работ по дисциплине «Электроника и микропроцессорная техника (Микропроцессоры)» и система команд однокристального микроконтроллера типа Intel 8048.

Методические указания рекомендуются для студентов, обучающихся по специальности 200401 по курсу «Электроника и микропроцессорная техника (Микропроцессоры)», подготовлены на кафедре РТУ.

* *
*

Целью работы является приобретение навыков программирования на языке ассемблера микропроцессорных устройств управления, изучение ресурсов и особенностей функционирования микроконтроллера типа Intel 8048, а также его системы команд, способов адресации и технических возможностей.

Приобретению таких навыков способствуют приводимые в данной работе примеры программ и рекомендации к выполнению лабораторных работ по дисциплине «Электроника и микропроцессорная техника (Микропроцессоры)».

Примеры реализации простейших стандартных подпрограмм

Приведенные ниже программы написаны в виде подпрограмм, где:

R0, R1 – регистры-указатели данных (data pointer)

R2 – регистр-расширитель аккумулятора

R3 – циклический счетчик (loop)

R0-R7 – рабочие регистры памяти данных

RX0 = R0

AEX = R2

Пример 1

Сложение двух однобайтовых чисел

Сложить два 8-разрядных числа **X** и **Y**, результат сложения поместить в регистр **R7**.

ADD8: MOV A, #X ;Непосредственная передача числа **X** в
;аккумулятор
ADD A, #Y ;Непосредственное сложение числа **Y** с
;содержимым аккумулятора
MOV R7, A ;Результат сложения поместить в регистр **R7**

Пример 2

Сложение двух двухбайтовых чисел

Сложить два 16-разрядных числа **X** и **Y**, результат сложения поместить в регистр **R6** (**High** – старший байт) и регистр **R7** (**Low** – младший байт).

ADD16: MOV A, #X Low
ADD A, #Y Low
MOV R7, A
MOV A, #X High
ADDC A, #Y High ;Непосредственное сложение старшего
;байта числа **Y** со старшим байтом
;числа **X** совместно со значением
;разряда переноса **C**
MOV R6, A ;Результат сложения поместить в
;регистр **R6**

Пример 3

Сложение двух двухбайтовых чисел с использованием косвенной адресации

DADD: DEC RX0 ;Получение младшего байта и сложение
ADD A, @RX0 ;его с содержимым аккумулятора
INC RX0 ;Получение старшего байта и сложение
XCH A, AEX ;его с содержимым **AEX**
ADDC A, @RX0
XCH A, AEX

Пример 4

Сложение двух двоично-десятичных чисел

Сложить двоично-десятичное число, цифра младшего значащего разряда которого расположена в ячейке BETA, с двоично-десятичным числом, цифра младшего значащего разряда которого расположена в ячейке ALPHA.

Количество пар цифр двоично-десятичного числа (его длина) составляет COUNT. Для данного примера принимаем, что оба числа имеют одинаковую длину и четное количество цифр (или, если нечетное количество цифр, цифра старшего значащего разряда равна нулю).

```
ADDBCD: MOV R0, ALPHA ;Загрузка в регистр R0 адреса младшего
          ;значащего разряда (ALPHA)
        MOV R1, BETA  ;Загрузка в регистр R1 адреса младшего
          ;значащего разряда (BETA)
        MOV R2, COUNT ;Загрузка числа COUNT в регистр R2
          ;для организации счетчика
        CLR C
LOOP:   MOV A, R0      ;Подпрограмма сложения
        ADDC A, R1
        DA A
        MOV R0, A     ;Запоминание суммы
        DEC R0        ;Декрементирование регистра указателя
        DEC R1        ;адреса
        DJNZ R2, LOOP ;Цикл подпрограммы LOOP
```

Пример 5

Вычитание однобайтового числа из однобайтового числа

Выполняется операция вычитания однобайтового числа из однобайтового числа с использованием операций сложения с двойным инвертированием. Результат вычисления – разность загрузить в регистр R7.

```

SUB8: MOV A, #SUBHND ;Загрузка вычитаемого в аккумулятор
      CPL A           ;инверсия вычитаемого
      INC A           ;A = A + 1
      ADD A, #MINEND ;получение разности
      MOV R7, A

```

Пример 6

Вычитание двухбайтового числа из двухбайтового числа

Найти разность двух 16-ти разрядных чисел и поместить ее в регистр R3 (старший байт – HIGH) и в регистр R4 (младший байт – LOW). Использование команды ADD вместо INC обеспечивает выполнение операции двойного инвертирования. Команда INC не влияет на бит переноса. Команды в подпрограмме снабдить комментариями самостоятельно.

```

SUB16: MOV A, #SUBLOW
      CPL A
      ADD A, #1           ;Дополнение до двух
      MOV R4, A
      MOV A, #SUBH1
      CPL A
      ADDC A, #0
      MOV R3, A
      MOV A, R4
      ADD A, #MINLOW
      MOV R4, A
      MOV A, R3
      ADDC A, #MINHI
      MOV R3, A

```

Пример 7

Вычисление двухбайтового числа из двухбайтового числа с использованием косвенной адресации

Поместить старший байт в регистр AEX, младший – в аккумулятор.
Команды в подпрограмме снабдить комментариями самостоятельно.

```
DMIN:  DEC RX0  
        CPL A  
        ADD A, @RX0  
        CPL A  
        INC RX0  
        XCH A, AEX  
        CPL A  
        ADDC A, @RX0  
        CPL A  
        XCH A, AEX
```

Пример 8

Загрузка двухбайтового числа из ОЗУ

Команды в подпрограмме снабдить комментариями самостоятельно.

```
DLD:  DEC RX0  
        MOV A, @RX0  
        INC RX0  
        XCH A, AEX  
        MOV A, @RX0  
        XCH A, AEX
```


Пример 9

Запись двухбайтового числа в ОЗУ

DST: DEC RX0 ;Передача содержимого аккумулятора в
MOV @RX0, A ;регистр, содержащий младший байт
INC RX0 ;Передача содержимого регистра AEX в
XCH A, AEX ;регистр, содержащий старший байт
MOV @RX0, A
XCH A, AEX

Пример 10

Сравнение без знака содержимого ячейки памяти с содержимым аккумулятора

Сравнить без учета знака содержимое ячейки памяти по адресу X и содержимое аккумулятора.

Содержимое аккумулятора сохранить в регистре временного хранения R5.

COMPAR: MOV R5, A
MOV R0, #X
CPL A
INC A
ADD A, @R0
JZ EQUAL ;Переход, если содержимое A равно
;содержимому ячейки памяти
JNC ACCGT ;Переход, если содержимое A больше
;содержимого ячейки памяти
JC ACCLT ;Переход, если содержимое A меньше
;содержимого ячейки памяти

ЛАБОРАТОРНАЯ РАБОТА № 1

(2-х час.) по курсу «Электроника и МП техника (Микропроцессоры)»,
раздел ОМЭВМ i8048 и ее аналоги

Цель работы: изучение функций эмулятора ОМЭВМ серии K1816 типа KM1816BE48, KM1816BE49 и овладение навыками программирования этих средств на языке ассемблера.

Задание: сравнить два однобайтовых слова.

Процедуры: сравниваются однобайтовые входные данные, поступающие из разных портов. Если сравниваемые величины равны, то производится установка флага пользователя F0 в "1" и вывод данных на порт данных BUS. Если сравниваемые входные данные не равны, то флаг F0 сбрасывается в "0" и большее из двух чисел передается на порт данных.

Порядок выполнения работы:

1. Начертить структуру алгоритма программы сравнения двух однобайтовых слов и представить преподавателю.

2. Написать, снабдив комментариями, и отладить при помощи средств эмулятора ОМЭВМ серии K1816 программу сравнения двух однобайтовых слов (решение задания оформить программой, которая пишется в любом текстовом редакторе, и после ее трансляции при помощи файла C49.exe и устранения ошибок (если они есть) загрузить Ваш файл с расширением *.hex в отладчик fd48.exe). Результаты представить преподавателю.

Ориентировочное число операторов в программе равно 18.

ЛАБОРАТОРНАЯ РАБОТА № 2

(2-х час.) по курсу МПУ, раздел ОМЭВМ i8048 и ее аналоги

Цель работы: изучение функций эмулятора ОМЭВМ серии K1816 типа KM1816BE48, KM1816BE49 и овладение навыками программирования этих средств.

Задание: перемножить два 8-разрядных числа. Результат – 16-разрядное число загрузить в регистр **R2 (High – старший байт)** и регистр **R3 (Low – младший байт)**.

Процедуры: организация циклического счетчика (8+1) в регистре **R5**. Загрузка множимого в регистр **R6**. Загрузка множителя и младшего байта произведения в регистр **R3**. Очистка аккумулятора **A** и признака переноса. Циклический сдвиг вправо с переносом:

```
LOOP:  RRC A           ;  
        XCH A, R3      ;  
        RRC A           ;  
        XCH A, R3      ;  
        JNC NOADD      ;  
        ADD A, R6       ;
```

```
NOADD: DJNZ R5, LOOP ;
```

Загрузка старшего байта произведения в регистр **R2**.

Порядок выполнения работы:

1. Начертить структуру алгоритма программы перемножения двух 8-разрядных чисел и представить преподавателю.

2. Написать, снабдив комментариями, и отладить при помощи средств эмулятора ОМЭВМ серии K1816 программу перемножения двух однобайтовых слов. Решение задания оформить программой,

которая пишется в любом текстовом редакторе, и после ее трансляции при помощи файла C49.exe и устранения ошибок (если они есть) загрузить Ваш файл с расширением *.hex в отладчик fd48.exe. Результаты представить преподавателю.

Ориентировочное число операторов равно 13.

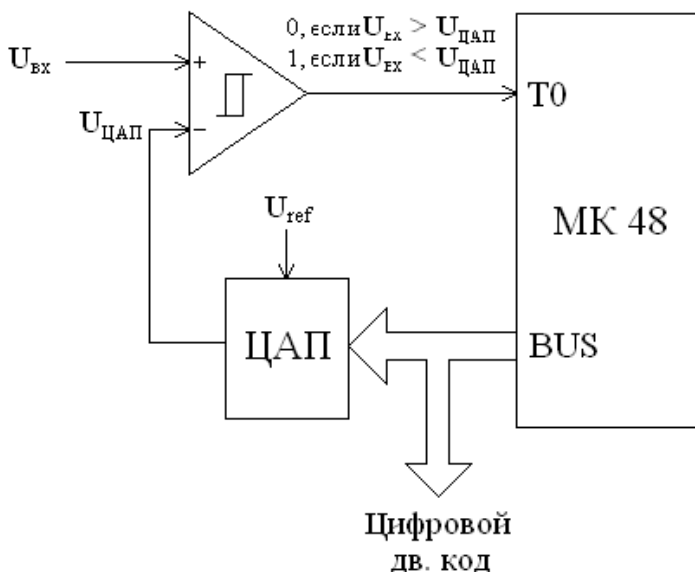
ЛАБОРАТОРНАЯ РАБОТА № 3

(4-х час.) по курсу МПУ, раздел ОМЭВМ i8048 и ее аналоги

Цель работы: изучение функций эмулятора ОМЭВМ серии К1816 типа КМ1816ВЕ48, КМ1816ВЕ49 и овладение навыками программирования этих средств.

Задание: реализовать программным способом АЦП последовательных приближений на базе ОМЭВМ, ЦАП и компаратора.

Структурная схема АЦП последовательных приближений на базе ОМЭВМ КМ1816ВЕ48, 8-разрядного ЦАП и компаратора имеет вид:



Микроконтроллер выдает 8-разрядные данные на ЦАП через порт BUS. Выходной сигнал $U_{\text{ЦАП}}$, снимаемый с выхода ЦАП, сравнивается с входным аналоговым сигналом $U_{\text{вх}}$ компаратором, который соответственно формирует логический уровень напряжения нуля или единицы. Результатом сравнения, если входной аналоговый сигнал меньше сигнала с ЦАП, является "1" и "0", – если входной аналоговый сигнал больше сигнала с ЦАП. Таким образом, этот соответствующий логический уровень напряжения с выхода компаратора поступает в микроконтроллер в качестве управляющего сигнала, причем, либо через порт P1, либо через входную линию T0 (вход, тестируемый по командам условного перехода JTO или JNTO).

При таком способе построения АЦП последовательных приближений осуществляется следующая последовательность операций: в старший разряд ЦАП устанавливается "1", и если на выходе компаратора появится "0", то она фиксируется в старшем разряде кода преобразуемого сигнала. Если на выходе компаратора "1", то "1", записанная в старший разряд ЦАП, сбрасывается в "0" и в старшем разряде кода преобразуемого сигнала запишется "0". Далее записывается "1" в следующий по порядку убывания разряд ЦАП, вновь производится сравнение с входным аналоговым сигналом и т.д.

Таким образом, формируются значения каждого разряда цифрового кода на входе ЦАП в порядке убывания номера разряда, при этом в цифровом коде преобразуемого сигнала записывается либо "1", либо "0".

Процедуры: записать в регистр-счетчик число используемых разрядов в ЦАП. Очистить аккумулятор А, рабочие регистры R5, R6 и флаг переноса С. Установить бит переноса в "1". Организовать циклический сдвиг от старшего разряда к младшему для поразрядного тестирования ЦАП в цикле работы регистра-счетчика:

LOOP: MOV A, R5 ;Сместить тестируемый разряд, значение
;которого хранится в R5, вправо от старшего
RRC A ;двоичного разряда к младшему (циклический
;сдвиг вправо с переносом)
MOV R5, A
ORL A, R6 ;Добавить к значению тестируемого разряда
;значение, хранящееся в R6
.....

Тестировать новое значение сигнала: если флаг установлен в "1", то сбрасывается в "0" единица последнего тестируемого разряда.

.....
MOV R6, A ;Если флаг установлен в "0", новое значение
;кода сохраняется

Декрементировать содержимое регистра-счетчика и, если получается не "0", переходить в цикле к следующему разряду (вход в цикл по метке LOOP). Код преобразования входного сигнала выводить на порт BUS.

Порядок выполнения работы:

1. Начертить алгоритмическую структуру принципа работы АЦП последовательного приближения на базе ОМЭВМ, ЦАП, компаратора и представить преподавателю.

2. Написать, снабдив комментариями, и отладить при помощи средств эмулятора ОМЭВМ серии К1816 программу работы АЦП (решение задания оформить программой, которая пишется в любом текстовом редакторе, и после ее трансляции при помощи файла С49.exe и устранения ошибок загрузить Ваш файл с расширением *.hex в отладчик fd48.exe). Результаты представить преподавателю.

Ориентировочное число операторов равно 14.

Система команд микроконтроллера Intel 8048

Система команд микроконтроллера Intel 8048 включает 96 базовых команд пересылки данных, арифметических и логических операций, передачи управления и управления режимами работы и ориентирована на реализацию процедур управления. К достоинствам системы команд следует отнести эффективный ввод/вывод (включая маскирование и возможность управления отдельными битами портов) и возможность ветвления по значению отдельных бит.

По формату команды могут быть однобайтовыми и двухбайтовыми. Они выполняются соответственно за один машинный цикл (2,5 мкс) и за два машинных цикла (5 мкс) при частоте кварца 6 МГц.

Учебное издание

**ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ ПРОГРАММИРОВАНИЯ
НА ЯЗЫКЕ АССЕМБЛЕРА**

Методические указания к лабораторному практикуму

Составитель *Данилов Анатолий Викторович*

Редактор Т.С. Петренко
Доверстка Т.С. Петренко

Подписано в печать 05.04.2010. Формат 60×84 1/16.

Бумага офсетная. Печать офсетная.

Печ. л. 1,0

Тираж 100 экз. Заказ . Арт. С - М14/2010

Самарский государственный
аэрокосмический университет.
443086, Самара, Московское шоссе, 34.

Изд-во Самарского государственного
аэрокосмического университета.
443086, Самара, Московское шоссе, 34.