

Коми Республикаса велёдан, наука да том йöz политика министерство
Министерство образования, науки и молодежной политики Республики Коми
Государственное профессиональное образовательное учреждение
«Сыктывкарский целлюлозно – бумажный техникум»

РАССМОТРЕНО
на заседании ПЦК информационных
дисциплин ГПОУ «СЦБТ»
Протокол № ____
« ____ » ____ г.
Председатель ПЦК _____ В.А. Шулепов
Преподаватель _____ Д.Д. Расов

УТВЕРЖДАЮ
Зам.директора по УПР
ГПОУ «СЦБТ»

« ____ » ____ 20 ____ г.
Е.В. Соколова

Методические указания по выполнению лабораторных работ

по учебной дисциплине МДК02.01 «Микропроцессорные системы»
для специальности
09.02.01 Компьютерные системы и комплексы

Программирование микропроцессора i8086 на языке Ассемблер

Сыктывкар 2018

ЛАБОРАТОРНАЯ РАБОТА №9

СПОСОБЫ И МЕТОДЫ ВЫВОДА ЧИСЕЛ

Цель работы: Освоить методы вывода чисел в двоичном, шестнадцатеричном и десятичном коде.

1. Вывод двоичного кода числа, записанного в регистр DH.

1.1. Методика выполнения.

Нужно последовательно проанализировать биты числа. В данной работе ограничимся байтом, который будем хранить в **DH**. Если бит нулевой (сброшен), то нужно вывести '0', если установлен, то '1'.

Алгоритм решения задачи:

- проанализировать значение одного бита;
- вывести значение бита;
- перейти к следующему биту. И так 8 раз (БАЙТ!) – ЦИКЛ

Анализ бита: **При анализе** значения программисты обычно используют команду **TEST**, но в данной лабораторной работе мы будем использовать следующую команду **SHL**.

Команда **SHL** осуществляет сдвиг влево всех битов операнда. Старший бит операнда поступает в флаг **CF**. Если команда записана в формате **SHL операнд**, 1 сдвиг осуществляется на 1 бит. В младший бит операнда загружается 0. Если команда записана в формате **SHL операнд,CL**, сдвиг осуществляется на число битов, указанное в регистре-счетчике **CL**, при этом в процессе последовательных сдвигов старшие биты операнда, пройдя через флаг **CF**, теряются, а младшие заполняются нулями

В качестве операнда команды **SHL** можно указывать любой регистр (кроме сегментного) или ячейку памяти размером, как в байт, так и в слово. Не допускается использовать в качестве операнда непосредственное значение.

Каждый сдвиг влево эквивалентен умножению знакового числа на 2, поэтому команду **SHL** удобно использовать для возведения операнда в степень 2.

Команда воздействует на флаги **OF**, **SF**, **ZF**, **PF** и **CF**.

Прореагировать на значение флага можно с помощью команды **JNC <метка>**

Осуществляется переход на метку, если флаг **CF** равен нулю, иначе выполняется команда, непосредственно идущая после команды.

Пример №1.1

.model tiny	; модель памяти, в которой сегменты кода, данных и стека объединены.
.code	; сегмент кода, который содержит данные.
org 100h	; начало СОМ-файла
begin:	; метка начала кода программы
mov DH,<55>	; заносим в регистр dh - любой ASCII-код символа
mov AH,2	; помещаем номер функции DOS "вывод строки (2)" в регистр AH.
mov CX,8	; инициализация переменной цикла
@1: mov DL,'0'	; заносим в DL код символа '0'
shl DH, 1	; сдвиг на 1 бит
jnc @2	; переход, если '0'
inc DL	; используем тот факт, что код символа '1' на единицу больше кода символа '0'
@2: int 21h	; вызов прерывания DOS - вызов символа;
LOOP @1	; переходим на метку @1
int 21h	; вызов прерывания DOS - вызов символа;
end begin	; метка окончания кода программы

Замечание: в описанном способе анализа значение исходного числа теряется. Иначе следует использовать команду **ROL**, но 8 раз для байта.

2. Вывод значения байта в шестнадцатеричной системе счисления

2.1. Методика выполнения.

Алгоритм решения задачи:

Допустим, что байт, значение которого нужно вывести, находится в регистре **DH**, и имеется таблица символов "0123456789ABCDEF". Байт состоит из двух шестнадцатеричных цифр. С учетом этого задачу можно решить так: нужно вывести на экран два символа из этой таблицы. Сначала - символ с номером, равным старшему полубайту числа, а потом с номером, равным младшему полубайту.

Для решения задачи нужно решить две небольшие проблемы:

- Записать в **AL** символ с нужным номером. Воспользуемся регистровым косвенным режимом адресации со смещением. Для этого значение каждого полубайта следует записывать в **BX**;
- Записать в **BX** значение полубайта.

Простейший способ решения задачи.

Пример №2.1

.model tiny	; модель памяти, в которой сегменты кода, данных и стека объединены.
.code	; сегмент кода, который содержит данные.
org 100h	; начало СОМ-файла
begin:	; метка начала кода программы
mov dh, 10	; заносим в регистр dh число 10
mov bl, dh	; заносим в регистр bl число 10
xor bh, bh	; Обнуление вх
and bl, 0F0h	; осуществлялем логическое (побитовое) умножение bl на 0f0h.
shr bl, 4	; сдвиг в право на 4 бита
mov al, table [bx]	; заносим в регистр al значение строки данных
int 29h	; вызов прерывания DOS - вызов символа
mov bl, dh	; заносим в регистр bl значение регистра dh
and bl, 0Fh	; осуществлялем логическое (побитовое) умножение bl на 0fh.

<code>mov al, table [bx]</code>	; заносим в регистр al значение строки данных
<code>int 29h</code>	; вызов прерывания DOS - вызов символа;
<code>mov al, 13</code>	; заносим в регистр al число 13
<code>int 29h</code>	; вызов прерывания DOS - вызов символа;
<code>mov al, 10</code>	; заносим в регистр al число 10
<code>int 29h</code>	; вызов прерывания DOS - вызов символа
<code>ret</code>	; функция DOS "завершить программу"
<code>table db '0123456789ABCDEF'</code>	; строка с содержащая выводимые данные.
<code>end begin</code>	; метка окончания кода программы

3. Вывод значения байта в десятичной системе счисления

3.1. Методика выполнения

Алгоритм решения задачи:

Будем считать, что байт, значение которого нужно вывести, находится в регистре **DH**. Однако теперь применим другой способ вывода символа цифры на экран: используем тот факт, что коды символов, обозначающих цифры, отличаются от них на **30h**. Но проблема здесь другая: заранее не известно, сколько цифр нужно отобразить, одну или три. Байт может принимать значение от 0 до 255. И есть еще одна проблема. При записи числа с применением позиционной системы записи в некоторой системе счисления поступают следующим образом: вычисляют и записывают остатки от деления числа на основание системы. Так поступают до тех пор, пока частное от деления не станет равным нулю. Затем остатки выписывают в порядке, обратном тому, как они получены.

Пример:

число = 251.

Делим на 10. Частное 25, остаток "1".

Делим на 10. Частное 2, остаток "5".

Делим на 10. Частное 0, остаток "2".

Нужно вывести на экран "2", "5", "1".

Задача решается с использованием стека программы. Остатки будем помещать в стек программы с помощью оператора **PUSH**. Одновременно будем подсчитывать число остатков, помещенных в стек. Счетчик - **CX**. Потом его используем для организации цикла, в котором будем извлекать остатки из стека оператором **POP**. Стек организован таким образом, что оператор **POP** извлекает последнее слово, которое было помещено туда оператором **PUSH**. Отметим, что оператор **PUSH** помещает в стек слово (**WORD**) или двойное слово (**DWORD**). Аналогично работает и оператор **POP**.

3.1.1 Команда PUSH (занесение операнда в стек).

Команда **push** уменьшает на 2 содержимое указателя стека **SP** и заносит на эту новую вершину двухбайтовый операнд-источник (проталкивает в стек новое данное). Проталкивать в стек можно только целые слова (не байты). Программа должна строго следить за тем, чтобы каждой команде проталкивания в стек **push** отвечала обратная команда выталкивания из стека **pop**. Если стек используется для временного хранения некоторых данных, то извлекать эти данные из стека следует в порядке, обратном их сохранению.

В качестве операнда-источника может использоваться любой 16-разрядный регистр (включая сегментный) или ячейка памяти. Команда **push** не воздействует на флаги процессора.

Пара команд **push** - **pop** часто используется для пересылки данного из регистра в регистр (особенно, в сегментный) через стек.

Пример 1

push ES: mem	; Сохранение содержимого
push DS	; слова памяти mem из
push BP	; дополнительного сегмента, а также регистров DS и
BP	
pop PP	; Восстановление из стека
pop DS	; трех operandов
pop ES: mem	; в обратном порядке

Пример 2

push DS ; Пересылка DS через стек
pop ES ; Теперь ES=DS

3.1.2 Простейший способ решения задачи вывода значения байта в десятеричной системе счисления.

Пример № 3.1

.model tiny	; модель памяти, в которой сегменты кода, данных и стека объединены.
.code	; сегмент кода, который содержит данные.
org 100h	; начало СОМ-файла
begin:	; метка начала кода программы
mov dh, 33	; заносим в регистр dh число 33
mov al, dh	; заносим в регистр al число 33
xor ah, ah	; обнуление ah
mov bl, 10	; заносим в регистр bl число 10
xor cx, cx	; обнуление cx (счетчик)
@1:	; устанавливаем метку @1
div bl	; делим целое число (регистр ax) без знака, на число (регистр bl)
push ax	; пересыпаем ax через стек
inc cx	; увеличиваем на 1 cx (счетчик)
xor ah, ah	; обнуление ah
or ax, ax	; логического (побитового) сложения само на себя
jnz @1	; если флаг zf не ноль, то переходим на метку @1
@2:	; вызов прерывания DOS - вызов символа;
pop ax	; заносим в регистр al число 10
xchg ah, al	; обмен данными между operandами ah и al
add al, 30h	; прибавляем к al число 30h
int 29h	; вызов прерывания DOS - вызов символа
loop @2	; реализация цикла - переходим на метку @2
mov al, 13	; заносим в регистр al число 13
int 29h	; вызов прерывания DOS - вызов символа
mov al, 10	; заносим в регистр al число 10
int 29h	; вызов прерывания DOS - вызов символа
ret	; функция DOS "завершить программу"
end begin	; метка окончания кода программы

4. Задание для выполнения.

- 4.1. С помощью редактора эмулятора EMU 8086 напишите программы, исходный текст которых приводится в примерах данной лабораторной работы.

- 4.2. Создайте исполняемые файлы типа ***.com**.
- 4.3. Изучите работу полученных программ.
- 4.4. Напишите программу для вывода на экран содержимого регистра **DS** (на основе **примера №2.1**). Сравните результат работы своей программы и того, что показывает отладчик.
- 4.5. Опишите работу команд **DIV, PUSH, POP, SHL, TEST**.
- 4.6. Установите (найдите адреса и запишите), где находятся числа, помещенные в стек.
- 4.7. Напишите программу для вывода на экран содержимого регистра **CS** (на основе **примера №3.1**).
- 4.8. Предложите другие способы решения поставленных задач.

5. Контрольные вопросы

- 5.1. Преимущества использования команды **SHL** вместо **TEST** (**пример №1.1**)?
- 5.2. Чем отличаются команды **SHL dx,1** и **SHL dx, cl**?
- 5.3. Как переслать содержимое X в стек и получить обратно?
- 5.4. Опишите методику вывода значения байта в десятеричной системе счисления?
- 5.5. Опишите методику вывода значения байта в шестнадцатеричной системе счисления?
- 5.6. Опишите методику вывода двоичного кода числа, записанного в регистр X
- 5.7. Стек. Принцип работы. Команды работы со стеком.
- 5.8. Укажите отличия в работе тандема команд:

push DS

pop ES

от

push DS

pop ES