

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Филиал
«Минский радиотехнический колледж»

Учебная дисциплина «Встраиваемые микропроцессорные системы»

Инструкция
по выполнению лабораторной работы
«Программирование на языке Ассемблер. Изучение команд логических операций и сдвига»

Минск
2017

Лабораторная работа № 8

Тема работы: «Программирование на языке Ассемблер. Изучение команд логических операций и сдвига»

1. Цель работы:

Формирование практических навыков по использованию команд логических операций и сдвига.

2. Задание

Изучить команды логических операций и сдвига. Написать на языке ассемблера, выполнить и исследовать с помощью отладчика программу для работы с логическими командами, командами сдвига, командами побитовой обработки.

3. Оснащение работы

Техническое задание, ПК, эмулятор DOSBox.

4. Основные теоретические сведения

Команды побитовой обработки данных манипулируют группами битов в регистрах или ячейках памяти.

Существует 3 группы команд побитовой обработки:

- логические команды;
- команды сдвига;
- команды циклического сдвига.

4.1. Логические команды. К логическим командам относятся команды AND, OR, XOR, TEST, NOT. Формат первых четырех команд:

команда приемник,источник

Формат команды NOT:

NOT приемник

Логические команды обрабатывают один байт или слово в регистре или в памяти и устанавливают флаги CF, OF, PF, SF, ZF (флаг AF не определен). При этом обработка байта (слова) осуществляется непосредственно бит за битом.

Логические операции часто используются для установки в двоичные разряды необходимых значений. Существуют следующие способы получения подобных результатов:

- для установки в данный разряд 0 необходимо логически умножить разряд на 0 (команда AND);
- для установки в данный разряд 1 необходимо логически сложить разряд с 1 (команда OR).

Например, предположим, что требуется, не меняя остальные 7 битов байта, присвоить пятому байту значение 1. Для этого достаточно логически сложить исходный байт с байтом, содержащим 00100000.

Рассмотрим действия логических команд:

AND: Если оба из сравниваемых битов приемника и источника равны

единице, то результирующий бит равен единице; во всех остальных случаях результат равен нулю.

OR: Если хотя бы один из двух сравниваемых битов приемника и источника равен единице, то результирующий бит приемника равен единице; если сравниваемые биты равны нулю, то результат равен нулю.

XOR: Если один из сравниваемых битов равен нулю, а другой равен единице, то результирующий бит равен единице; если оба бита одинаковы, то результат равен нулю.

NOT: Устанавливает обратное значение битов в байте или слове, регистре или ячейке памяти.

TEST: Действует как AND, устанавливает флаги, но не меняет биты.

Примеры,

```
MOV AL,11000101B
```

```
MOV BH,01011100B
```

```
AND AL,BH      ; устанавливает в AL 010001000
```

```
OR CL,CL       ; устанавливает флаги CF и ZF
```

4.2. Команды сдвига. Команды сдвига осуществляют сдвиг 8- или 16-битового содержимого регистров на одну или несколько позиций влево или вправо. Формат:

команда приемник,счетчик

В этих командах содержимое приемника сдвигается на величину, задаваемую счетчиком. При этом счетчик может быть цифрой 1 или значением без знака в регистре CL.

Особенностью команд сдвига является то, что они помещают во флаг переноса CF значение бита, сдвинутого за один из концов приемника.

Команды сдвига делятся на команды логического, арифметического и циклического сдвига.

4.2.1. Команды арифметического сдвига.

SAL / SAR – сдвинуть арифметически влево / вправо

Команда SAL не сохраняет знак операнда, но заносит 1 во флаг переполнения CF и 0 в освобождающийся бит. Команда SAR сохраняет знак операнда, занося его при каждом сдвиге в старший бит операнда. Например,

```
MOV BL,10110100B
```

```
MOV AL,BL      ; пусть CF=1
```

```
SAL AL,1       ; AL=01101000, CF=1
```

```
MOV AL,BL
```

```
SAR AL,1       ; AL=11011010, CF=0
```

4.2.2. Команды логического сдвига:

SHL / SHR – сдвинуть логически влево / вправо

Команда SHL идентична команде SAL. Команда SHR аналогична SAR, но сдвигает операнд вправо, занося 0 в старший бит операнда. Примеры,

```
MOV BL,10110100B
```

```
MOV AL,BL      ; пусть CF=1
```

```
SHL AL,1       ; AL=01101000, CF=1
```

MOV AL,BL

SHR AL,1 ; AL=01011010, CF=0

Поскольку сдвиг операнда на 1 бит удваивает значение операнда, а сдвиг на 1 бит вправо уменьшает его вдвое, то команды сдвига можно использовать в качестве команд быстрого умножения и деления.

Примеры,

MOV CL,2

SHL AX,CL ; умножить число без знака на 4

4.3. Команды циклического сдвига:

ROL / ROR – сдвинуть циклически влево / вправо

При выполнении этих команд вышедший за пределы операнда бит входит в него с другого конца и помещается во флаг переноса. Примеры:

MOV BL,10110100B ; пусть CF=1

MOV AL,BL

ROL AL,1 ; AL=01101001, CF=1

RCL / RCR – сдвинуть циклически вместе с флагом переноса влево/вправо.

При выполнении этих команд осуществляется сдвиг, в противоположный конец операнда помещается значение флага CF и затем в CF помещается вышедший за пределы операнда бит.

Примеры:

MOV BL,10110100B ; пусть CF=1

MOV AL,BL

RCR AL,1 ; AL=11011010, CF=0

4.4. Простейшие способы ввода символов с клавиатуры и вывода символов на экран.

Для вывода символов на экран в текущей позиции курсора, необходимо:

- определить область данных для вывода, при этом последним символом области вывода должен быть знак \$;
- занести в область данных для вывода необходимые данные;
- установить в регистре AH значение 09;
- занести в регистр DX адрес области данных для вывода;
- указать команду прерывания INT 21H.

Пример. Для вывода на экран текста 'Введите данные' необходимы следующие команды: STR DB 'Введите данные','\$'

...

MOV AH,09

LEA DX,STR

Для ввода последовательности символов с клавиатуры необходимо:

- определить область памяти в сегменте данных для вводимых символов, причем длина этой области должна быть на 2 байта больше максимальной длины строки с учетом того, что последний введенный символ всегда будет 'RETURN' (код 0DH). Первый байт содержит максимальное число символов, а второй байт будет содержать число реально введенных символов до нажатия

клавиши 'RETURN';

- установить в регистре AH значение 10;
- поместить в регистр DX адрес области памяти;
- указать команду прерывания INT 21H.

Пример. Для ввода с клавиатуры строки максимальной длиной в 50 символов необходимы следующие команды:

```
STR DB 50,52 DUP(?)
```

```
...
```

```
MOV AH,10
```

```
LEA DX,STR
```

```
INT 21H
```

4.5. Пример программы. Следующая программа вводит строку символов латинского алфавита и заменяет все строчные буквы на прописные, а прописные – на строчные. Результат выводится на экран. Например, строка Table будет заменена на строку tABLe и выведена на экран.

```
STACKSG SEGMENT PARA STACK 'stack'
```

```
DB 64 DUP(?)
```

```
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'
```

```
STR DB 10,12 DUP(' ')
```

```
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'
```

```
ASSUME CS:CODESG,DS:DATASG,SS:STACKSG
```

```
ENTRY PROC FAR
```

```
PUSH DS
```

```
SUB AX,AX
```

```
PUSH AX
```

```
MOV AX,DATASG
```

```
MOV DS,AX
```

```
; Ввод исходной строки
```

```
MOV AH,10
```

```
LEA DX,STR
```

```
INT 21H
```

```
; Блок замены символа 'A' на 'a'
```

```
MOV SI,2
```

```
SUB CX,CX ; CX=CX-CX
```

```
MOV CL,STR+1 ; длина строки
```

```
M1: MOV AL,STR[SI] ; занести в AL первый символ
```

```
CMP AL,'A' ; < 'A' ?
```

```
JB NEXT1
```

```
CMP AL,'z' ; > 'z' ?
```

```
JA NEXT1
```

```

        CMP AL,'a'      ; < 'a' ?
        JB  MARK1
;
        XOR AL,00100000B      ; заменить
        MOV STR[SI],AL ; занести новый символ в строку
        JMP SHORT NEXT1
MARK1: OR  AL,00100000B      ; заменить
        MOV STR[SI],AL      ; занести новый символ в строку
NEXT1: INC SI ; перейти к следующему символу (SI=SI+1)
        LOOP M1
; Печать результирующей строки
PRINT1:
        MOV Ah,09h
        LEA  DX,STR
        SUB  BX,BX
        MOV BL,STR+1
        MOV STR[BX+2],'$'
        MOV STR,0Ah
        MOV STR+1,0Dh
        INT  21H
        RET
ENTRY  ENDP
CODESG ENDS
        END  ENTRY

```

5. Порядок выполнения работы

1. Ввести с терминала строку символов. Вывести на экран ее двоичное представление.
2. Ввести с терминала строку символов. Вывести на экран количество единичных битов третьего символа строки.
3. Дано двоичное число. Вывести на экран его шестнадцатеричное представление.
4. Дано двоичное число. Вывести на экран его восьмеричное представление.
5. Ввести с терминала строку символов. Вывести на экран количество нулевых битов второго символа строки.

6. Форма отчета о работе

Лабораторная работа № ____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. Классификация команд побитовой обработки данных.
2. Перечислите логические команды.
3. Формат логических команд.
4. Какие флаги формируются при выполнении логических команд?
5. Охарактеризуйте команду AND.
6. Охарактеризуйте команду OR.
7. Охарактеризуйте команду XOR.
8. Охарактеризуйте команду NOT.
9. Охарактеризуйте команду TEST.
10. Какую функцию выполняют команды сдвига? Особенности их выполнения.
11. Какие флаги формируются при выполнении команд сдвига?
12. Особенности выполнения команд арифметического сдвига, команд логического сдвига, команд циклического сдвига

8. Рекомендуемая литература

Финогенов, К. Г. Основы языка Ассемблера [Текст] / К. Г. Финогенов. – М.: Радио и связь, 2000.

Финогенов, К. Г. Использование языка Ассемблера [Текст]: учеб. пособие для вузов / К.Г. Финогенов. – М.: Горячая линия Телеком, 2004.

Юров, В. И. Assembler [Текст]: учеб. пособие для вузов / В. И. Юров. 2-е изд. – СПб.: Питер, 2007.