# Лабораторная работа №4. Логические команды и команды манипулирования битами

**Цель работы**

Изучение команд передачи управления и получение навыка работы с ними.

**Постановка задачи**

1. Занести в память 32 разрядное шестнадцатеричное число в соответствии с вариантом из таблицы 6;
2. Подсчитать количество нулей и единиц в данном числе двумя разными способами;
3. Подсчитать количество парных нулей и парных единиц в данном числе;
4. В младшем байте числа обменять между собой биты 0-7, 1-6, 2-5, 3-4.

Результаты каждого пункта сохранить в памяти и распечатать.

**Содержание отчета**

Отчет по лабораторной работе должен содержать номер, название, цель лабораторной работы, задание на лабораторную работу, текст программы на языке ассемблера, распечатки формы в активном состоянии с результатами выполнения лабораторной работы в виде дампа памяти и фрагмента программы, содержащей ассемблерную вставку, а также контрольный пример и выводы по работе.

**Варианты заданий**

Таблица 6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер  варианта | Число (HEX) | Номер  варианта | Число(HEX) | Номер  варианта | Число(HEX) |
| 1 | **DEFBCDEF** | 21 | **23456ABC** | 41 | **4467ADEC** |
| 2 | **264535АВ** | 22 | **34578ABC** | 42 | **BCDE5587** |
| 3 | **DFAC3421** | 23 | **DABC47AE** | 43 | **75421DDE** |
| 4 | **1254FEDB** | 24 | **BDCE4578** | 44 | **23794685** |
| 5 | **12546FD1** | 25 | **58697EFF** | 45 | **CEF59764** |
| 6 | **FDCB5478** | 26 | **22546781** | 46 | **162894BB** |
| 7 | **ABCDEF45** | 27 | **FADCDAD5** | 47 | **FEADCBDE** |
| 8 | **AAABCD45** | 28 | **FEAD45AD** | 48 | **CCBDEA43** |
| 9 | **BC41424D** | 29 | **12451346** | 49 | **1457BCEC** |
| 10 | **1245BC41** | 30 | **ADAECB44** | 50 | **CBDEA785** |
| 11 | **DA2AD355** | 31 | **578FAEDB** | 51 | **CBDAE145** |
| 12 | **47AC2EFA** | 32 | **ABCD1542** | 52 | **CBAFE432** |
| 13 | **BCD45AE1** | 33 | **1679EADB** | 53 | **CBFAE342** |
| 14 | **14523678** | 34 | **1346FAED** | 54 | **1678ADEC** |
| 15 | **ABCDEF11** | 35 | **ABABDFAB** | 55 | **47215F57** |
| 16 | **BCDEF44A** | 36 | **4678FEAB** | 56 | **ADCBE875** |
| 17 | **1245DBCC** | 37 | **CB45AE2A** | 57 | **BCDE22EA** |
| 18 | **ABE4578A** | 38 | **25467BCA** | 58 | **4875ADCB** |
| 19 | **8754AD4F** | 39 | **12478BCE** | 59 | **BCDEFA2B** |
| 20 | **12345678** | 40 | **BCDAE14A** | 60 | **17563BCB** |

**Краткие теоретические сведения**

Различают команды, выполняющие логические операции и команды манипулирования битами. Команды выполняют операции с битами байта, слова или двойного слова, поэтому еще называются побитовыми (поразрядными) командами.

Теоретической базой для логических операций является булева алгебра, которая впервые была исследована Дж. Булем (1815-1864). Она базируется на высказывани­ях. Высказывание – это законченное предложение, о котором можно определенноска­зать, что его содержание истинно или ложно. Операции булевой алгебры определены для таких высказываний. Всевозможные наборы входных значений и значений булевой функции на этих наборах входных значений составляют таблицу истинности данной булевой функции.

К командам, выполняющим логические операции, и командам манипулирования битами относятся:

AND – Операция логического умножения «И»;

OR – Операция логического сложения «ИЛИ»;

XOR (eXclusive OR) – Операция «исключающее ИЛИ»;

NOT – Операция двоичной инверсии;

SAL (Shift Arithmetic Left) – Арифметический сдвиг влево;

SAR (Shift Arithmetic Right) – Арифметический сдвиг вправо;

SHL (SHift Left) – Сдвиг влево;

SHR (SHift Right) – Сдвиг вправо;

SHLD (SHift Left Double) – Сдвиг влево двойной точности;

SHRD (SHift Right Double) – Сдвиг вправо двойной точности;

ROL (ROtate Left) – Циклический сдвиг влево;

ROR (ROtate Right) – Циклический сдвиг вправо;

RLC (Rotate through Carry Left) – Циклический сдвиг влево через флаг CF;

RRC (Rotate through Carry Right) – Циклический сдвиг вправо через флаг CF;

BT (Bit Test) – Переместить бит для тестирования;

BTS (Bit Test and Set) – Протестировать бит и установить его;

BTC (Bit Test and Complement) – Протестировать бит и инвертировать его;

BTR (Bit Transfer and Reset) – Протестировать бит и сбросить его;

BSF (Bit Scan Forward) – Сканировать биты в прямом направлении;

BSR (Bit Scan Reverse) – Сканировать биты в обратном направлении;

TEST – Протестировать операнд;

CLC (Clear CF) – Очистить флаг CF;

CLD (Clear DF) – Очистить флаг DF;

CMC (Complement CF) – Инвертировать флаг CF;

STC (Set CF) – Установить флаг CF;

STD (Set DF) – Установить флаг DF;

**Команды AND,OR,XOR**

Формат команды: ANDП,И

Формат команды: ORП,И

Формат команды: XORП,И

Команды выполняют побитовые логические операции над приемником П, в качестве которого могут выступать регистр или ячейка памяти, и источником И, в качестве которого могут выступать регистр, ячейка памяти или константа. Источник и приемник не могут быть одновременно ячейками памяти. Результат операции помещается в приемник П. Таблицы истинности для каждой функции представлены в таблице 7.

таблица 7.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **AND** | | |  | **OR** | | |  | **XOR** | | |  | **NOT** | |
| Логическое умножение | | |  | Логическое сложение | | |  | Логическое исключающее ИЛИ | | |  | Логическое отрицание | |
| X | Y | Вых. |  | X | Y | Вых. |  | X | Y | Вых. |  | Х | Вых. |
| 0 | 0 | 0 |  | 0 | 0 | 0 |  | 0 | 0 | 0 |  | 0 | 1 |
| 0 | 1 | 0 |  | 0 | 1 | 1 |  | 0 | 1 | 1 |  | 1 | 0 |
| 1 | 0 | 0 |  | 1 | 0 | 1 |  | 1 | 0 | 1 |  |  |  |
| 1 | 1 | 1 |  | 1 | 1 | 1 |  | 1 | 1 | 0 |  |  |  |

Один из распространенных вариантов применения команды AND для выполнения операции маскирования или наложения маски. Как и в случае с обычной, театральной, маской, сквозь нее видно только там, где есть прорези. Если нам необходимо, чтобы в каком-либо операнде одни значения битов под конкретными номерами сохранились, а другие нет (обнулились), то мы выполним команду AND, в которой в качестве второго операнда будет такое значение, в котором на месте битов, значения которых надо сохранить, будут стоять «1», а во всех остальных нули.

Команду OR чаще всего используют для выборочной установки отдельных битов без изменения значений остальных битов. Это достигается за счет того, что команда OR выполняет логическое сложение, а значит, сложение с нулем не меняет значение бита, а сложение с «1» установит бит, вне зависимости от его предыдущего значения.

Команда XOR используется в разных логических операциях. Иногда, чаще используют две ее особенности. Если выполнить команду XOR, в которой приемник и источник равны, то результатом команды будет нуль. Вторая особенность, если дважды выполнить команду XOR с одним и тем же источником, то приемник станет равным своему первоначальному значению.

Флаги, которые устанавливаются в результате этой команды, приведены в таблице 8.

Таблица 8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Синтаксис** | **Биты регистра флагов** | | | | |
| **OF** | **SF** | **ZF** | **PF** | **CF** |
| **AND** П, И | 0 | + | + | + | 0 |
| **TEST** П,И | 0 | + | + | + | 0 |
| **OR** П,И | 0 | + | + | + | 0 |
| **XOR** П,И | 0 | + | + | + | 0 |
| **NOT** П | - | - | - | - | - |

**Команда NOT**

Формат команды: NOTП

Команда NOT выполняет операцию побитовой инверсии приемника П, в качестве которого могут выступать РОН или ячейка памяти.

**Команда TEST**

Формат команды: TESTП,И

Команда TEST эквивалентна команде AND, т.е. побитово осуществляет операцию логического умножения над приемником и источником и устанавливает флаги (см.таблицу 8), но результат операции никуда не записывается. Команда TEST, так же как и CMP, используется в основном в сочетании с командами условного перехода.

**Команды SAL, SAR, SHL, SHR, ROL, ROR, RLC, RRC**

Формат команды:SALП,И

Формат команды:SARП,И

Формат команды:SHLП,И

Формат команды:SHRП,И

Формат команды: ROLП,И

Формат команды: RORП,И

Формат команды: RLCП,И

Формат команды: RRCП,И

Таблица 9.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Мнемоника | Графическое изображение выполняемых действий | Влияние на флаги | | | | | | |
|  | OF | SF | ZF | PF | CF |  |
| SAL |  |  | ? | + | + | + | + |  |
| SAR |  |  | ? | + | + | + | + |  |
| SHL |  |  | ? | + | + | + | + |  |
| SHR |  |  | ? | + | + | + | + |  |
| ROL |  |  | + | - | - | - | + |  |
| ROR |  |  | + | - | - | - | + |  |
| RLC |  |  | + | - | - | - | + |  |
| RRC |  |  | + | - | - | - | + |  |

В данных командах происходит сдвиг битов, в соответствие с таблицей 9, приемника П, в качестве которого может выступать РОН или ячейка памяти, на число двоичных разрядов, указанных в источнике И, в качестве которого может выступать число 1, байтовая константа или регистр CL.

Одно из применений этих команд состоит в том, что они позволяют умножать (при сдвиге влево) или делить (при сдвиге вправо) значение приемника на 2**n**, где n – количество сдвигов. При этом скорость выполнения команд сдвига намного больше, чем скорость выполнения команд умножения или деления. Например, следующий фрагмент программы:

MOV BX,AX

SHL AX,2

ADD AX,BX

SHL AX,1

выполнит умножение содержимого регистра AX на 10 в ОДИННАДЦАТЬ раз быстрее, чем команда MUL.

**Команды SHLD, SHRD**

Формат команды:SHLDП,И1,И2

Формат команды:SHRDП,И1,И2

В данных командах происходит сдвиг битов, в соответствие с таблицей 10, приемника П, в качестве которого могут выступать только 16-ти и 32-ух разрядные РОН или ячейки памяти на число двоичных разрядов, указанных во втором источнике И2, в качестве которого может выступать байтовая константа или регистр CL. Источник И1 содержит биты, которые будут заноситься в приемник по мере смещения, т.е. будет служить источником битов, при этом, в приемник П занесется столько битов из И1, сколько будет указано в И2. В самом И1 биты не сдвигаются и к концу операции в И1 будет храниться первоначальное значение. В качестве И1 может использоваться только РОН, причем такой же размерности, что и П.

Таблица 10.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Мнемоника | Графическое изображение выполняемых действий | Влияние на флаги | | | | | | |
|  | OF | SF | ZF | PF | CF |  |
| SHLD | П И1 |  | ? | + | + | + | + |  |
| SHRD | И1 П |  | ? | + | + | + | + |  |

**Команда BT**

Формат команды:BTИ1,И2

Команда BT считывает во флаг CF значение бита из первого источника И1, в качестве которого может выступать только 16-ти и 32-ух разрядный РОН или ячейка памяти, с номером, указанным во втором источнике И2, в качестве которого может выступать только 16-ти и 32-ух разрядный РОН или байтовая константа. Значение второго источника берется по модулю 32.

После выполнения команды BT флаг CF равен значению считанного бита, а флаги OF, SF, ZF, AF и PF не определены.

**Команды BTS, BTR, BTC**

Формат команды:BTSИ1,И2

Формат команды:BTRИ1,И2

Формат команды: BTCИ1,И2

Эти команды повторяют действие команды BT, т.е. считывают во флаг CF значение бита из первого источника И1 с номером, указанным во втором источнике И2, но после этого, соответственно, считанный бит в приемнике П устанавливается в 1 (BTS), сбрасывается в 0 (BTR) и инвертируется (BTC).

После выполнения команд BTS, BTR и BTC флаг CF равен значению считанного бита до его изменения в результате действия команды, а флаги OF, SF, ZF, AF и PF не определены.

**Команды BSF, BSR**

Формат команды:BSFП,И

Формат команды: BSRП,И

BSF сканирует источник И, в качестве которого может выступать только 16-ти и 32-ух разрядный РОН или ячейка памяти, начиная с самого младшего бита (т.е. слева направо), и записывает в приемник П, в качестве которого может выступать только РОН такой же разрядности, что и источник И, номер первого встретившегося бита, равного 1. Команда BSR эквивалентна команде BSF, но сканирует источник И, начиная с самого старшего бита. Например, если источник И равен 0000'0000'0000'01002, то и команда BSF, и команда BSR возвратит 2.

Если весь источник равен нулю, значение приемника не определено и флаг ZF устанавливается в 1, иначе ZF сбрасывается. Флаги OF, SF, CF, AF и PF не определены.

**Команды CLC, CLD, CMC, STC, STD**

Формат команды: CLC

Формат команды: CLD

Формат команды: CMC

Формат команды: STC

Формат команды: STD

Команда CLC сбрасывает флаг CF (CF=0).

Команда CLD сбрасывает флаг DF (DF=0).

Команда CMC инвертирует флаг CF (CF=).

Команда STC устанавливает флаг CF (CF=1).

Команда STD устанавливает флаг DF (DF=1).

**Примерные контрольные вопросы и задачи**

1. Приведите фрагменты программ на языке ассемблера, выполняющие следующие действия: сдвиг СХ на 2 бита вправо; сдвиг АХ на 2 бита влево; приравнивание значения 426 регистру СХ; сравнение значения переменной BYTE1 типа Byte и числа 2516.

2. Полагая, что DL содержит 1111'00012, а переменная BOOL типа Byte содержит 1110'00112, определите значение регистра DL после выполнения каждой отдельно взятой команды:

а) AND DL,BOOL b) OR DL,BOOL

c) XOR DL,BOOL d) AND DL,0

e) XOR DL,$FF f) NOT DL

После выполнения каких команд будет установлен флаг ZF?

3. Полагая, что DX содержит значение 1011'1001'1011'10012, определите, какое значение будет содержать DX после выполнения следующих отдельно взятых инструкций:

a) SHL DL,1 b) SHL DX,2 c) SHR DX,1

d) SAR DX,2 e) SAL DH,3 f) ROR DX,3

g) ROR DL,3 h) RCL DX,17

4. Чему будет равен регистр АX в результате следующего фрагмента программы:

mov cx,$0C

BSF AX,CX

1. Чему будет равен регистр СX в результате следующего фрагмента программы (М – переменная типа Word):

mov CX,$2C   
mov m,CX

bsf cx,m

Как приравнять одной командой регистр ECX к нулю десятью разными способами (можно использовать команды из всех лабораторных работ)? Начальное значение регистра ECX и флагов случайное.