Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №1**

«Линейное исполнение программ. Арифметические и поразрядные логические операции над байтами»

Выполнил:

студент группы 213 Водолад Д.В.

Проверил:

доцент кафедры ПОиАИС Кривонос А.В.

Курск, 2021

***Цели работы:***

1. изучение принципов функционирования памяти и микропроцессора компьютера при последовательном исполнении команд программы;
2. приобретение навыков использования арифметических команд при написании ассемблерных программ;
3. приобретение навыков использования поразрядных логических команд при написании ассемблерных программ;
4. получение представления об особенностях обработки данных разных размерностей и режимах доступа к данным при выполнении арифметических операций.

***Контрольные вопросы***

1. Понятие сегмента, характеристики сегмента, организация сегмента.

Сегмент — это прямоугольная область памяти, характеризующаяся начальным адресом и длиной. Начальный адрес (адрес начала сегмента) – это номер (адрес) ячейки памяти, с которой начинается сегмент. Длина сегмента – это количество входящих в него ячеек памяти. Сегменты могут иметь различную длину.

Сегменты могут иметь различную длину. Все ячейки, расположенные внутри сегмента, перенумеровываются, начиная с нуля. Адресация ячеек внутри сегмента ведется относительно начала сегмента; адрес ячейки в сегменте называется смещением или эффективным адресом - EA (относительно начального адреса сегмента).

1. На какие сегменты разбита память компьютера? В какие регистры записываются начальные адреса сегментов?

В общем случае программа, размещенная в памяти, может иметь следующие сегменты: сегмент данных для хранения операндов, сегмент кода для хранения операторов программы и сегмент стека – дополнительную память для временного размещения информации.

Команды программы размещаются в сегменте кода, начальный адрес которого хранится в специальном сегментном регистре CS. При выполнении текущей команды микропроцессор должен знать адрес следующей команды, которая поступит на исполнение. Этот адрес хранится в специальном регистре IP. Таким образом, содержимое регистров CS и IP однозначно определяет местонахождение команды.

1. Какие регистры микропроцессора используются при выполнении арифметических операций?

* eax/ax/ah/al (Accumulator register) - *аккумулятор*. Применяется для хранения промежуточных данных. В некоторых командах использование этого регистра обязательно.
* edx/dx/dh/dl (Data register) - регистр *данных*. Так же, как и регистр *eax/ax/ah/al*, он хранит промежуточные данные. В некоторых командах его использование обязательно; для некоторых команд это происходит неявно. Используется как расширение регистра- аккумулятора при работе с 32- разрядными числами.

1. На какие флаги воздействуют арифметические команды?

* Флажок переноса CF. При сложении (вычитании) устанавливается в единицу, если возникает перенос из младшего бита или заем из старшего бита.
* Флажок вспомогательного переноса AF. Устанавливается в единицу, если при сложении (вычитании) возникает перенос (заем) из бита 3. Только для двоично-десятичной арифметики.
* Флажок переполнения OF. Устанавливается в единицу, если возникает переполнение, т.е. получение результата вне допустимого диапазона. При сложении флажок устанавливается, если имеется перенос в старший бит и нет переноса из старшего бита и наоборот.

1. Какие режимы адресации могут применяться для доступа к данным при выполнении арифметических и поразрядных логических операций?

* Непосредственный. Данное длинной 8 или 16 бит является частью команды.
* Прямой. 16-битный эффективный адрес данного является частью команды.
* Регистровый. Данное содержится в определяемом командой регистре (Аx,bx,cx,dx,si,di,sp,bp).

1. Особенности выполнения операции умножения. Особенности выполнения операции деления. Распределение регистров

Особенностью операций умножения и деления является наличие всего одного операнда(сомножителя или делимого). Второй операнд задан неявно; его местоположение фиксировано и зависит от размера операндов. Знаки результатов в операциях со знаком определяются по алгебраическим правилам.

Если результат по размеру совпадает с размером сомножителей, то флаги CF и OF после завершения операции раны нулю, в противном случае - устанавливаются в единицу. Это значит, что результат вышел за пределы младшей части произведения и состоит из двух частей, что необходимо учитывать при дальнейшей работе. Остальные флаги не определены. Расположение операндов при умножении представлено в таблице 1.

Таблица 1 – Расположение операндов при умножении

|  |  |  |
| --- | --- | --- |
| Первый  сомножитель | Второй  сомножитель | Результат |
| Байт | AL | 16 битов в AX; AL – младшая часть результата, AH – старшая часть результата  (AX) ← (SRC) \* (AL) |
| Слово | AX | 32 бита в паре DX:AX; AX – младшая часть результата, DX – старшая часть результата  (DX:AX) ← (SRC) \* (AX) |
| Двойное слово | EAX | 64 бита в паре EDX:EAX; EAX – младшая часть результата, EDX – старшая часть результата  (EDX:EAX) ← (SRC) \* (EAX) |

Делитель может находиться в регистре или в памяти и иметь размер 8, 26 или 32 бита. Местонахождение делимого фиксировано. Результатом команды деления являются частное и остаток от деления. После выполнения операции деления содержимое флагов не определено, но возможно возникновение прерывания с номером ноль (так называемое «деление на ноль») в случаях, когда делитель равен нулю или частное не входит в отведенную для него разрядную сетку.

Что касается режимов адресации, то операнды источники не могут быть непосредственными значениями, а все другие режимы адресации допустимы. Операнды – получатели строго фиксированы. Расположение операндов при умножении представлено в таблице 2.

Таблица 2 – Расположение операндов при делении

|  |  |  |  |
| --- | --- | --- | --- |
| Делимое | Делитель | Частное | Остаток |
| Слово (16 бит) в  регистре AX | Байт в регистре или ячейке памяти | Байт в регистре AL  (AL)← (AX)/ (SRC) | Байт в регистре AH  (AH)← (AX)/ (SRC) |
| Двойное слово (32 бита), в DX – старшая часть в AX – младшая часть | Слово (16 бит) в регистре или ячеек памяти | Слово (16 бит) в регистре AX  (AX) ← (DX:AX)/ (SRC) | Слово (16 бит) в регистре DX  (DX) ← (DX:AX)/ (SRC) |
| Учетверенное слово (64 бита), в EDX – старшая часть, в EAX – младшая часть | Двойное слово (32 бита) в регистре или ячейке памяти | Двойное слово (32 бита) в регистре EAX  (EAX)← (EDX:EAX)/ (SRC) | Двойное слово (32 бита) в регистре EDX  (EDX)← (EDX:EAX)/ (SRC) |

1. Основные логические операции и принципы их выполнения. Логические команды выполняют логические операции над битами операндов.

Размерность операндов должна быть одинакова. Описание логических команд представлено в таблице 3.

Таблица 3 - Формат логических команд

|  |  |  |
| --- | --- | --- |
| Название команды | Мнемоника и формат команды | Описание действия |
| Инвертировать | NOT OPR | (OPR)←not OPR |
| Объединить по «ИЛИ» | OR DST,SRC | (DST)←(DST) or (SRC) |
| Объединить по «И» | AND DST,SRC | (DST)←(DST) and (SRC) |
| Сложить по MOD2 («исключающее ИЛИ») | XOR DST,SRC | (DST)←(DST) xor (SRC) |
| Проверить | TEST OPR1,OPR2 | OPR1 and OPR2 |

Команда NOT не воздействует на флажки. Остальные команды сбрасывают OF и CF, оставляют АF не определенным и устанавливают СF, ZF, PF по обычным правилам. В команде NOT не допускается непосредственный операнд. В остальных командах один из операндов должен быть регистром. Другой операнд может иметь любой режим адресации.

1. Правила формирования масок для установки и сброса битов.

Логические команды наиболее часто используются для селективных установок, инвертирования, сброса или проверки бит в операнде- получателе в соответствии с двоичным набором операнда-источника. Такие действия часто встречаются в операциях над битами регистров и данных ввода-вывода. При этом операнд источник называют маской, а сама операция называется маскированием. Установка или сброс битов происходит при помощи логических операций, представленных в таблице 3.

1. Каким образом выполняются логические команды над словами?

Система команд микропроцессора содержит пять команд, поддерживающих данные операции. Эти команды выполняют логические операции над битами операндов. Размерность операндов, естественно, должна быть одинакова. Например, если размерность операндов равна слову (16 бит), то логическая операция выполняется сначала над нулевыми битами операндов и ее результат записывается на место бита 0 результата. Далее команда последовательно повторяет эти действия над всеми битами с первого до пятнадцатого. Аналогично происходит и с другими размерностями операндов.

***Текст программы***

d\_s segment

data1 db 10, 27

data2 db 0101B

el db 3

sum db 0

raz db 0

pr dw 1

mod1 db 0

divv db 0

d\_s ends

c\_s segment

assume ds:d\_s,cs:c\_s

begin:

mov ax,d\_s

mov ds,ax

mov dl,data1 ;10+27

add dl,data1+1

mov sum,dl

mov al, data1; 10 - 27 signed

sub al, data1+1

mov raz, al

neg data1+1; 10 - (-27)

mov al, data1

sub al, data1+1

mov raz, al

mov al, data1 ; 10 \* (-27)

imul data1+1

mov pr, ax

neg data1+1 ; 10/27

mov al, data1

mov ah, 0h

div data1+1

mov mod1, ah

mov divv, al

mov el, 3

or al, el

not al

mov el, 7

not el

and al, el

mov dl, al

xor dl, al

mov ah,4ch

int 21h

c\_s ends

end begin