МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра информационных систем

Отчет   
по лабораторной работе №3  
на тему: «Команды условного и безусловного переходов. Организация ветвлений и циклов в программе»  
по дисциплине «Архитектура ЭВМ и систем»

Выполнили: Марочкин М.А. Шифр: 170584  
 Яшин М. О. Шифр: 170133  
 Шорин В.Д. Шифр: 171406  
ИПАИТ  
Направление: 09.03.04 «Программная инженерия»  
Группа: 71-ПГ  
Проверил(а):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Отметка о зачете:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Дата «\_\_\_\_» \_\_\_\_\_\_ 2017 г.

Орел, 2017г.

**Цель работы.**

Цели лабораторной работы:

1. Изучение принципов функционирования памяти и микропроцессора компьютера при выполнении ветвлений и циклов.

2. Приобретение навыков использования команд условного и безусловного переходов, циклов при написании ассемблерных программ.

3. Получение представления об особенностях обработки данных, команд и режимах доступа к данным при организации ветвлений и циклов.

**Контрольные вопросы:**

1. Ветвления в алгоритмах. Реализация ветвлений на языке ассемблера.

Ветвление (условный переход) происходит только при соблюдении определенного условия, в противном случае выполняется следующая по порядку команда программы. Условием, на основании которого осуществляется переход, чаще всего выступают признаки результата выполнения предшествующей арифметической или логической команды. Каждый из признаков фиксируется в своем разряде регистра флагов PSW. Возможен и такой подход, когда решение о переходе принимается в зависимости от состояния одного из регистров общего назначения, куда предварительно помещается результат операции сравнения CMP.

2.Команды условного и безусловного переходов.

Команда безусловного перехода обеспечивает переход по заданному адресу без проверки каких-либо условий.

Формат команды:

JMP Модификатор адрес перехода

Этой командой задаются внутрисегментные и межсегментные переходы. Модификатор указывает вид перехода (внутрисегментный прямой, внутрисегментный косвенный, межсегментный прямой, межсегментный косвенный) и принцип изменения содержимого регистров CS и IP. Модификатор не всегда указывается в команде JMP.

Условный переход осуществляется, если выполняется определённое условие, заданное флагами процессора. Состояние флагов изменяется после выполнения арифметических, логических и некоторых других команд. Если условие не выполняется, то управление переходит к следующей команде.

Для организации ветвления используется команда CMP:

CMP OPR1, OPR2

В этом случае выполняется сравнение путем вычитания операндов, при этом сами операнды не изменяются.

Команды условного перехода: JE (перейти, если равно), JL (перейти, если меньше), JG (перейти, если больше) и др.

3. Циклы в алгоритмах. Организация циклов на языке ассемблера. Особенности и ограничения цикла LOOP.

Циклы организуются для многократного повторения одной или нескольких команд программы или процедуры. Цикл можно организовать, используя команды условного и безусловного переходов или с помощью специальных команд.

Команда LOOP и е расширения позволяет организовывать циклы, подобные циклам for в языках высокого уровня с автоматическим уменьшением счетчика цикла. Количество повторений содержится в регистре CX, который в командах управления циклами выполняет функции счетчика цикла.

4. В каком регистре находится во время выполнения программы смещение кода? Каким образом вычисляется адрес команды?

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.

5. Какую принципиальную роль играет оператор безусловного перехода JMP при организации ветвлений?

Команда JMP передает управление в другую точку программы, не сохраняя какой-либо информации для возврата. Оператор JMP бывает необходим при организации ветвлений с циклами для возврата к началу цикла.

6. Что означает корректное завершение программы?

Для окончания работы программы DOS должна выполнить некоторые служебные действия, в том числе освободить занимаемую программой память. Для этого используется функция DOS с номером 4Ch. Эта функция предполагает, что в регистре AL находится код завершения нашей программы, который она передаст DOS. Если программа завершилась успешно, код завершения должен быть равен нулю. После этого вызываем DOS командой int 21h. Поскольку выполняемая часть программы закончилась нужно закрыть сегмент команд с помощью директивы ends.

7. Реальный и защищённый режимы работы процессора. Вычисление физических адресов ячеек памяти.

В реальном режиме процессор функционирует фактически также, как Intel 8086 с повышенным быстродействием и может обращаться только к 1 Мбайту адресного пространства. Оставшаяся память, даже если она установлена на компьютере, использоваться не может. В защищённом режиме также используются сегменты и смещения, но физические начальные адреса сегментов извлекаются из таблиц сегментных дескрипторов, индексируемых с помощью тех же сегментных регистров. Каждый сегментный дескриптор занимает 8 байт, из которых 4 байта (32 бита) отводятся под сегментный адрес. Такой механизм позволяет обеспечить полное использование 32-разрядного адресного пространства.

**Текст программы.**

d\_s segment

a1 db 24h

a2 db 1bh

nod db ?

a db 0

b db ?

d\_s ends

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

c\_s segment

begin:

mov ax, d\_s

mov ds, ax

mov ah, a1

mov al, a2

return: cmp ah, al

je met1 ; переход на метку met1, если значения регистров равны

jg met2 ; переход на метку met2, если значение в ah больше значения в al

jmp met3 ; переход на метку met3

met1: mov bh, ah

jmp stop

met2: sub ah, al

jmp return

met3: sub al, ah

jmp return

stop: mov bl, bh

mov cx, 8 ; в регистр cx заносится количество повторений цикла

mov dl, 0

met4: shr bl, 1 ; начало цикла по метке met4

jc met5 ; переход на метку met5, если флаг переноса содержит единицу

jmp met6

met5: inc dl

met6: loop met4 ; конец цикла

mov ah, 4ch

int 21h

c\_s ends

end begin