**Федеральное государственное образовательное учреждение   
высшего образования**

**«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ И.С. ТУРГЕНЕВА»**

**Институт приборостроения, автоматизации и информационных технологий**

**Кафедра информационных систем и цифровых технологий**

Отчет к лабораторной работе № 2

«Организация межсегментных переходов»

Выполнила:

Студентка группы 21ПГ

Банных Мария Алексеевна

Приняла:

доцент кафедры   
информационных систем   
и цифровых технологий  
Конюхова Оксана Владимировна  
  
доцент кафедры   
информационных систем   
и цифровых технологий  
Амелина Ольга Викторовна

Орёл 2022 год

Ответы на контрольные вопросы

1. *Внутрисегментные и межсегментные переходы. Способы вычисления адресов переходов.*

Внутрисегментный прямой. В адресном поле команды содержится адрес перехода – адрес, по которому размещается следующая выполняемая команда.

Внутрисегментный косвенный. Используется прямой режим адресации данных для указания местоположения адреса перехода, т.е. адрес адреса.

Межсегментный прямой. Начальный адрес нового сегмента кода загружается в сегментный регистр CS, а смещение – в регистр IP.

Межсегментный косвенный. 4-байтовый адрес перехода размещается в смежных ячейках памяти по некому адресу (смещению) в сегменте данных.

1. *Разновидности внутрисегментных переходов и их особенности.*

Внутрисегментный прямой.

Эффективный адрес перехода вычисляется как сумма текущего содержимого указателя команд IP и 8/16 - битного относительного смещения. Данный режим допустим в условных и безусловных переходах.

Внутренний косвенный.

В этом случае двухбайтовый адрес перехода размещается в ячейках памяти по некоторому адресу (смещению) в сегменте данных. В команде перехода это смещение указывается в регистре процессора или ячейке памяти с помощью любого режима адресации данных, кроме непосредственного.

Содержимое указателя команд IP заменяется соответствующим содержимым регистра или ячейки памяти. Данный способ допустим только в командах безусловного перехода.

1. *Флаги процессора и их использование в условиях.*

Условием, на основании

которого осуществляется переход, чаще всего выступают признаки

результата выполнения предшествующей арифметической или логической команды (без флага AF). Каждый из признаков фиксируется в

своѐм 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.

Команда CMP устанавливает статусные флаги в зависимости от

результата сравнения операндов, не меняя самих операндов.

1. *Команды линейного логического и арифметического сдвигов. В чем заключается разница их выполнения? Области применении этих команд.*

Сдвинуть логически влево SHL OPR,CNT

Сдвинуть арифметически влево SAL OPR,CNT

Сдвинуть логически вправо SHR OPR,CNT

Сдвинуть арифметически вправо SAR OPR,CNT

Команды арифметического сдвига позволяют выполнить «быстрое» умножение и деление операнда на степени двойки. Например, сдвиг числа влево на один разряд аналогичен его умножению на 2 , а сдвиг числа вправо на один разряд аналогичен делению его на 2. Преимущество этих команд по сравнению с традиционными командами умножения и деления заключается в скорости исполнения: команды сдвига выполняются быстрее.

1. *Особенности выполнения команд циклического сдвига. Сферы применения этих команд.*

В командах циклического сдвига через перенос сдвигаемый бит сначала помещается во флаг переноса CF. Только при следующем выполнении той же команды находящийся во флаге CF бит вталкивается с другой стороны операнда, а во флаг CF помещается следующий бит сдвигаемого числа

Команды циклического сдвига отличаются от команд сдвига тем, что операнд считается «кольцом», в котором выдвигаемые с одной стороны биты вдвигаются с другой стороны

1. *Что указывает директива ASSUME в программе?*

Она ассоциирует соответствующие сегменты в программе.

Практическая часть:

.model small

stec segment

db 20 dup (' ') ;зарезервировали 20 байт

stec ends

data1 segment

a db 00101100b ;44(d) = 2C(h)

data1 ends

data2 segment

addr dd per2

data2 ends

data3 segment

b db 00110100b ;52(d) = 34(h)

data3 ends

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

kod1 segment

assume DS: data1, CS: kod1

begin:

jmp far ptr start

per2:

mov ax, data1

mov ds, ax

xor ax, ax

mov bl, a ;bl = 2C

ROR bl, 2 ;bl = B

jmp far ptr per3

kod1 ends

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

kod2 segment

assume DS: data3, CS: kod2

per3:

mov ax, data3

mov ds, ax

xor ax, ax

mov bh, b ;bh = 34

ROR bh, 4 ;bh = 43

jmp far ptr exit

kod2 ends

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

kod3 segment

assume CS: kod3

start:

assume DS: data1

mov ax, data1

mov ds, ax

xor ax, ax

mov bl, a ;bl = 2C

SHL bl, 1 ;bl = 58

assume DS: data3

mov ax, data3

mov ds, ax

xor ax, ax

mov cl, b ;cl = 34

SHR cl, 2 ;cl = D

assume DS: data2

mov ax, data2

mov ds, ax

jmp DS: addr

;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

exit:

mov ah, 4ch ;завершение программы (делаем прерывание)

int 21h

kod3 ends

end begin

end