Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(национальный исследовательский университет)

Кафедра «Математического и компьютерного моделирования»

Отчет по лабораторной работе №1

«Способы адресации данных в машинных командах»

*Выполнил(а): Савонин Максим Витальевич*

*Специальность:***02.03.01**

*Группа:* ЕТ-111

*Проверил: Дрозин Д.А.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2022 г.

Челябинск,

2022

Содержание

[Цель работы. 3](#_Toc387303631)

[Ход работы. 3](#_Toc387303632)

[Прямая адресация. 4](#_Toc387303633)

[Косвенная адресация 6](#_Toc387303634)

[Индексная адресация. 8](#_Toc387303635)

[Стековая адресация. 10](#_Toc387303636)

[Выводы 12](#_Toc387303637)

[Список использованной литературы: 13](#_Toc387303638)

# Цель работы

Составить программы для сложения многобайтных двоичных данных, использую следующие типы адресации:

* Прямая.

Это адресация, при которой в составе команды находится не сам операнд, а его адрес.

* Косвенная.

Это вид адресации, при которой адрес операнда размещается в одном из регистров.

* Индексная

Адрес операнда вычисляется как сумма содержимого регистров SI или DI и смещения

* Стековая

Операнд находится в стеке, на вершину которого указывает регистр SP.

Проверить выполнение программ и сравнить полученные результаты с контрольными.

# Ход работы

1. Закодируем исходные слова в ASCII коде.

|  |  |
| --- | --- |
| Слово | Код ASCII |
| Savonin | 53 61 76 6F 6E 69 6E |
| Maxim | 4D 61 78 69 6D |

1. Сложим значения вручную.

Первое слагаемое: 53 61 76 6F 6E 69 6E

Второе слагаемое: 4D 61 78 69 6D

…………………………………………………………………………………………………………………………………………………

Результат сложения: 53 61 C3 D0 E6 D2 DB

1. Распределим память.

Первое слово: DS:100

Второе слово: DS:110

Результат: DS:120

Код программы: DS:200

## Прямая адресация.

Символическая запись программы, Н-коды и комментарии:

1B9C 0200:A10001 MOV AX,[0100] ;Поместим значение 1 и 2 букв первого слова в АХ

1B9C 0203 03061001 ADD AX,[0110] ;Сложим с значением 1 и 2 букв второго операнда

1B9C 0207 A32001 MOV [0120],AX ;Запишем результат

1B9C 020A:A10201 MOV AX,[0102] ;Поместим значение 3 и 4 букв первого слова в АХ

1B9C 020D:13061201 ADC AX,[0112] ;Сложим с значением 3 и 4 букв второго операнда

1B9C 0211:A32201 MOV [0122],AX ;Запишем результат

1B9C 0214:A00401 MOV AL,[0104] ;Поместим значение 5 буквы перового операнда в AL

1B9C 0217:B400 MOV AH,00 ;Поместим значение 0 в АН

1B9C 0219:13061401 ADC AX,[0114] ;Сложим с значением 5 буквы второго операнда

1B9C 021D:A32401 MOV [0124],AX ;Запишем результат

1B9C 0220:B80000 MOV AX,0000 ;Запишем в регистр АХ значение 0

1B9C 0223:13060501 ADC AX,[0105] ;Сложим с значением 5 и 6 букв первого операнда

1B9C 0227:A32501 MOV [0125],AX ;Запишем результат

1B9C 022A:B80000 MOV AX,0000 ;Запишем в регистр АХ значение 0

1B9C 022D:1206070 ADC AX,[0107] ;Сложим с значением 7 и 8 букв первого операнда

1B9C 0231:A32701 MOV [0127],AX ;Запишем результат

1B9C 0234:B80000 MOV AX,0000 ;Запишем в регистр АХ значение 0

1B9C 0237:13060901 ADC AX,[0109] ;Сложим с значением 9 и 10 букв первого операнда

1B9C 023B:A32901 MOV [0129],AX ;Запишем результат

1B9C 023E:B80000 MOV AX,0000 ;Запишем в регистр АХ значение ноль

1B9C 0241:13060B01 ADC AX,[010B] ;Сложим с последними буквами первого операнда

1B9C 0245:A32B01 MOV [0128],AX ;Запишем результат

1B9C 0248:F4 HLT ;Конец программы

Выполнение программы:

|  |  |
| --- | --- |
| Выполняемые операции | Результат |
| MOV AX,[0100]  ADD AX,[0110]  MOV [0120],AX | A5 D1 |
| MOV AX,[0102]  ADC AX,[0112]  MOV [0122],AX | A5 D1 E1 EC |
| MOV AL,[0104]  MOV AH,00  ADC AX,[0114]  MOV [0124],AX | A5 D1 E1 EC CF |
| MOV AX,0000  ADC AX,[0105]  MOV [0125],AX | A5 D1 E1 EC CF 75 74 |
| MOV AX,0000  ADC AX,[0107]  MOV [0127],AX | A5 D1 E1 EC CF 75 74 64 69 |
| MOV AX,0000  ADC AX,[0109]  MOV [0129],AX | A5 D1 E1 EC CF 75 74 64 69 6E 6F |
| MOV AX,0000  ADC AX,[010B]  MOV [0128],AX | A5 D1 E1 EC CF 75 74 64 69 6E 6F 76 61 |

Состояние памяти после выполнения программы:

DS:100 4B 68 75 73 6E 75 74 64 69 6E 6F 76 61

DS:110 5A 69 6C 79 61

DS:120 A5 D1 E1 EC CF 75 74 64 69 6E 6F 76 61

Полученные результаты совпадают с контрольными, программа работает верно

## Косвенная адресация

Символическая запись программы, Н-коды и комментарии:

1B9C:0200 BB0001 MOV BX,0100 ;В ВХ записываем начальный адрес первого операнда

1B9C:0203 BE1001 MOV SI,0110 ;В SI записываем начальный адрес второго операнда

1B9C:0206 BF2001 MOV DI,0120 ;В DI записываем начальный адрес результата

1B9C:0209 8B07 MOV AX,[BX] ;Поместим значение 1 и 2 букв первого слова в АХ

1B9C:020B 0304 ADD AX,[SI] ;Сложим с значением 1 и 2 букв второго операнда

1B9C:020D 8905 MOV [DI],AX ;Запишем результат

1B9C:020F 43 INC BX ;Увеличим значение ВХ на 2

1B9C:0210 43 INC BX

1B9C:0211 46 INC SI ;Увеличим значение SIна 2

1B9C:0212 46 INC SI

1B9C:0213 47 INC DI ;Увеличим значение DI на 2

1B9C:0214 47 INC DI

1B9C:0215 8B07 MOV AX,[BX] ;Поместим значение 3 и 4 букв первого слова в АХ

1B9C:0217 1304 ADC AX,[SI] ;Сложим с значением 3 и 4 букв второго операнда

1B9C:0219 8905 MOV [DI],AX ;Запишем результат

1B9C:021B 43 INC BX ;Увеличим значение ВХ на 2

1B9C:021C 43 INC BX

1B9C:021D 46 INC SI ;Увеличим значение SIна 2

1B9C:021E 46 INC SI

1B9C:021F 47 INC DI ;Увеличим значение DI на 2

1B9C:0220 47 INC DI

1B9C:0221 8A07 MOV AL,[BX] ;Поместим значение 5 буквы первого слова в АL

1B9C:0223 B400 MOV AH,00 ;Поместим значение 0 в АН

1B9C:0225 1304 ADC AX,[SI] ;Сложим с значением 5 буквы второго операнда

1B9C:0227 8905 MOV [DI],AX ;Запишем результат

1B9C:0229 43 INC BX ;Увеличим значение ВХ на 1

1B9C:022A 46 INC SI ;Увеличим значение SIна 1

1B9C:022B 47 INC DI ;Увеличим значение DI на 1

1B9C:022C B90400 MOV CX,0004 ;Поместим значение 4 в регистр СХ, т.е. дадим понять

;что цикл выполняется 4 раза

1B9C:022F B80000 MOV AX,0000 ;Поместим значение 0 в АХ

1B9C:0232 1307 ADC AX,[BX] ;Складываем значение первого операнда с содержимым АХ

1B9C:0234 8905 MOV [DI],AX ;Запишем результат

1B9C:0236 43 INC BX ;Увеличим значение ВХ на 2

1B9C:0237 43 INC BX

1B9C:0238 47 INC DI ;Увеличим значение DI на 2

1B9C:0239 47 INC DI

1B9C:023A E2F3 LOOP 022F ;Выполняем цикл

1B9C:023C F4 HLT ;Конец программы

Выполнение программы:

|  |  |
| --- | --- |
| Выполняемые операции | Результат |
| MOV AX,[BX]  ADD AX,[SI]  MOV [DI],AX  INC BX  INC BX  INC SI  INC SI  INC DI  INC DI | A5 D1 |
| MOV AX,[BX]  ADC AX,[SI]  MOV [DI],AX  INC BX  INC BX  INC SI  INC SI  INC DI  INC DI | A5 D1 E1 EC |
| MOV AL,[BX]  MOV AH,00  ADC AX,[SI]  MOV [DI],AX  INC BX  INC SI  INC DI | A5 D1 E1 EC CF |
| MOV CX,0004  MOV AX,0000  ADC AX,[BX]  MOV [DI],AX  INC BX  INC BX  INC DI  INC DI  LOOP 022F | A5 D1 E1 EC CF 75 74 64 69 6E 6F 76 61 |

Состояние памяти после выполнения программы:

DS:100 4B 68 75 73 6E 75 74 64 69 6E 6F 76 61

DS:110 5A 69 6C 79 61

DS:120 A5 D1 E1 EC CF 75 74 64 69 6E 6F 76 61

Полученные результаты совпадают с контрольными, программа работает верно

## Индексная адресация.

Символическая запись программы, Н-коды и комментарии:

1B9C:0200 BE0000 MOV SI,0000 ;В SI записываем значение 0

1B9C:0203 8B840001 MOV AX,[0100+SI] ;Поместим значение первых букв первого слова в АХ

1B9C:0207 13841001 ADD AX,[0110+SI] ;Сложим с значением первых букв второго операнда

1B9C:020B 89842001 MOV [0120+SI],AX ;Запишем результат

1B9C:020F 46 INC SI ;Увеличим значение SIна 2

1B9C:0210 46 INC SI

1B9C:0211 8B840001 MOV AX,[0100+SI] ;Поместим следующие значения первого слова в АХ

1B9C:0215 13841001 ADC AX,[0110+SI] ;Сложим с значением букв второго операнда

1B9C:0219 89842001 MOV P0120+SI],AX ;Запишем результат

1B9C:021D 46 INC SI ;Увеличим значение SIна 2

1B9C:021E 46 INC SI

1B9C:021F 8A840001 MOV AL,[0100+SI] ;Поместим значение букв первого слова в АL

1B9C:0223 B400 MOV AH,00 ;Поместим значение 0 в АН

1B9C:0225 13841001 ADC AX,[0110+SI] ;Сложим с значением букв второго операнда

1B9C:0229 89842001 MOV [0120+SI],AX ;Запишем результат

1B9C:022D 46 INC SI ;Увеличим значение SIна 1

1B9C:022E B90400 MOV CX,0004 ;Поместим значение 4 в регистр СХ, т.е. дадим понять

;что цикл выполняется 4 раза

1B9C:0231 8B840001 MOV AX,[0100+SI] ;Запишем в АХ значение букв первого операнда

1B9C:0235 150000 ADC AX,0000 ;Сложим АХ с нулем

1B9C:0238 89842001 MOV [0120+SI],AX ;Запишем результат

1B9C:023C 46 INC SI ;Увеличим значение SIна 2

1B9C:023D 46 INC SI

1B9C:023E E2F1 LOOP 0231 ;Выполняем цикл

1B9C:0240 F4 HLT ;Конец программы

Выполнение программы:

|  |  |
| --- | --- |
| Выполняемые операции | Результат |
| MOV AX,[0100+SI]  ADD AX,[0110+SI]  MOV [0120+SI],AX  INC SI  INC SI | A5 D1 |
| MOV AX,[0100+SI]  ADC AX,[0110+SI]  MOV P0120+SI],AX  INC SI  INC SI | A5 D1 E1 EC |
| MOV AL,[0100+SI]  MOV AH,00  ADC AX,[0110+SI]  MOV [0120+SI],AX  INC SI | A5 D1 E1 EC CF |
| MOV CX,0004  MOV AX,[0100+SI]  ADC AX,0000  MOV [0120+SI],AX  INC SI  INC SI  LOOP 0231 | A5 D1 E1 EC CF 75 74 64 69 6E 6F 76 61 |

Состояние памяти после выполнения программы:

DS:100 4B 68 75 73 6E 75 74 64 69 6E 6F 76 61

DS:110 5A 69 6C 79 61

DS:120 A5 D1 E1 EC CF 75 74 64 69 6E 6F 76 61

Полученные результаты совпадают с контрольными, программа работает верно

## Стековая адресация.

Символическая запись программы, Н-коды и комментарии:

1B9C:0200 89E5 MOV BP,SP ;Сохраняем указатель стека

1B9C:0202 FF360001 PUSH [0100] ;Заталкиваем буквы операндов в стек через один

1B9C:0206 FF361001 PUSH [0110]

1B9C:020A FF360201 PUSH [0102]

1B9C:020E FF361201 PUSH [0112]

1B9C:0212 FF360401 PUSH [0104]

1B9C:0216 A01401 MOV AL,[0114]

1B9C:0219 B400 MOV AH,00

1B9C:021B 50 PUSH AX

1B9C:021C FF360601 PUSH [0106]

1B9C:0220 FF360801 PUSH [0108]

1B9C:0224 FF360A01 PUSH [010A]

1B9C:0228 A00C01 MOV AL,[010C]

1B9C:022B B400 MOV AH,00

1B9C:022D 50 PUSH AX

1B9C:022E 8B46FE MOV AX,[BP-02] ;В АХ записываем буквы первого операнда

1B9C:0231 90 NOP ;Ничего не делаем

1B9C:0232 0346FC ADD AX,[BP-04] ;Суммируем значения первого и второго операнда

1B9C:0235 50 PUSH AX ;Результат заталкиваем в стек

1B9C:0236 8B46FA MOV AX,[BP-06] ;В АХ записываем буквы первого операнда

1B9C:0239 1346F8 ADC AX,[BP-08] ;Суммируем значения первого и второго операнда

1B9C:023C 50 PUSH A ;Результат заталкиваем в стек

1B9C:023D 8B46F6 MOV AX,[BP-0A] ;В АХ записываем буквы первого операнда

1B9C:0240 1346F4 ADC AX,[BP-0C] ;Суммируем значения первого и второго операнда

1B9C:0243 50 PUSH AX ;Результат заталкиваем в стек

1B9C:0244 B80000 MOV AX,0000 ;В АХ записываем 0

1B9C:0247 1346F2 ADC AX,[BP-0E] ;Суммируем АХ в буквами перового операнда

1B9C:024A 50 PUSH AX ;Результат заталкиваем в стек

1B9C:024B B80000 MOV AX,0000 ;В АХ записываем 0

1B9C:024E 1246F0 ADC AX,[BP-10] ;Суммируем АХ в буквами перового операнда

1B9C:0251 50 PUSH AX ;Результат заталкиваем в стек

1B9C:0252 B80000 MOV AX,0000 ;В АХ записываем 0

1B9C:0255 1346EE ADC AX,[BP-12] ;Суммируем АХ в буквами перового операнда

1B9C:0258 50 PUSH AX ;Результат заталкиваем в стек

1B9C:0259 B80000 MOV AX,0000 ;В АХ записываем 0

1B9C:025C 1346EC ADC AX,[BP-14] ;Суммируем АХ в буквами перового операнда

1B9C:025F 50 PUSH AX ;Результат заталкиваем в стек

1B9C:0260 8F062C01 POP [012C] ;Все элементы выталкиваем из стека начиная в DS:120

1B9C:0264 8F062A01 POP [012A]

1B9C:028F052801 POP [0128]

1B9C:026C 8F062601 POP [0126]

1B9C:0270 8F062401 POP [0124]

1B9C:0274 8F062201 POP [0122]

1B9C:0278 8F062001 POP [0120]

1B9C:027C HLT ;Конец программы

Выполнение программы:

|  |  |
| --- | --- |
| Выполняемые операции | Результат |
| MOV BP,SP  …  PUSH AX | n/a |
| POP [012C] | 61 |
| POP [012A] | 6F 76 61 |
| POP [0128] | 69 6E 6F 76 61 |
| POP [0126] | 74 64 69 6E 6F 76 61 |
| POP [0124 | CF 75 74 64 69 6E 6F 76 61 |
| POP [0122] | E1 EC CF 75 74 64 69 6E 6F 76 61 |
| POP [0120] | A5 D1 E1 EC CF 75 74 64 69 6E 6F 76 61 |

Состояние памяти после выполнения программы:

DS:100 4B 68 75 73 6E 75 74 64 69 6E 6F 76 61

DS:110 5A 69 6C 79 61

DS:120 A5 D1 E1 EC CF 75 74 64 69 6E 6F 76 61

Полученные результаты совпадают с контрольными, программа работает верно

# Выводы

Первый абзац должен раскрывать суть вашей работы, то есть дать ответ на вопрос – Что сделано? Далее должно следовать заключение о том, в каких случаях эффективно применять каждый из видов адресации, какой способ адресации предпочли бы вы лично и почему.

# Список использованной литературы:

1. Персональные ЭВМ IBM PC ХТ. Программирование на языке ассемблера: Пер. с англ. – 2-е изд., стереотип. – М.: Радио и связь. 1991. – 336 с.: ил.
2. [Микропроцессор Intel 486](http://www.physdep.isu.ru/method/mpi486/index.htm) - http://www.physdep.isu.ru/method/mpi486/22.htm