Лабораторная работа № 4

# Применение косвенной адресации

Цель работы:

1. Изучение косвенной адресации ЭВМ.

2. Разработка циклической программы с переадресацией для модели учебной ЭВМ.

Теоретические сведения:

**Косвенная адресация**

Косвенная адресация получила широкое применение в современных микропроцессорных системах. Без нее сложно реализовать эффективную работу с массивами памяти, обеспечить циклическое выполнение фрагментов программ, а также обеспечить оперировать адресами как данными (адресная арифметика).

Косвенная адресация — предполагает, что в поле адреса команды располагается адрес ячейки памяти, в которой хранится адрес операнда ("адрес адреса") (рисунок 1 а)). Такой способ позволяет оперировать адресами как данными, что облегчает организацию циклов, обработку массивов данных и др. Его основной недостаток— потеря времени на двойное обращение к памяти — сначала за адресом, потом — за операндом.

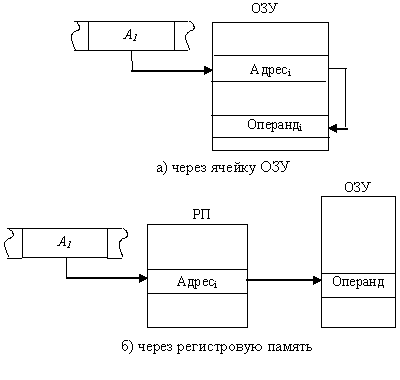


Рис. 1. Косвенная адресация памяти:

а) через ячейку ОП, б) через регистр.

Разновидность — косвенно-регистровая адресация, при которой в поле команды размещается адрес РОН, хранящего адрес операнда (рисунок 1 б)). Этот способ, помимо преимущества обычной косвенной адресации, позволяет обращаться к большой памяти с помощью коротких команд и не требует двойного обращения к памяти (обращение к регистру занимает гораздо меньше времени, чем к памяти).

Реализация косвенной адресации в модельной ЭВМ

Все типы адресации, используемые в учебной ЭВМ приведены в таблице 1.

Таблица 1. Адресация в командах учебной ЭВМ

|  |  |  |
| --- | --- | --- |
| **Код ТА** | **Тип адресации** | **Исполнительный адрес** |
| 0 | Прямая (регистровая) | ADR (R) |
| 1 | Непосредственная | — |
| 2 | Косвенная | O3Y(ADR)[3:5] |
| 3 | Относительная | ADR + RB |
| 4 | Косвенно-регистровая | POH(R)[3:5] |
| 5 | Индексная с постинкрементом | POH(R)[3:5], R:=R + 1 |
| 6 | Индексная с предцекрементом | R:=R-1,POH(R)[3:5] |

Тип адресации, используемый в команде определяется полем команды ТА разряд [2] команды). Косвенной адресации соответствуют коды 2 и 4 (косвенно-регистровая), содержащиеся в данном поле.

Этот тип адресации может быть использован практически всеми одно- и 2-х адресными командами модельной ЭВМ. При этом разряды [3:5] команды могут определять прямой или косвенный адрес памяти, номер регистра (в команде mov номера двух регистров), адрес перехода или короткий непосредственный операнд.. При записи программы в мнемокодах для обозначения этого типа адресации используются обозначения приведенные в таблице 2.

Таблица 2. Обозначение способов адресации при использовании мнемокодов.

|  |  |  |  |
| --- | --- | --- | --- |
| **Обозначение** | **Код** | **Тип адресации** | **Пример команды** |
|  | **0** | Прямая (регистровая) | ADD 23 (ADD R3) |
| **#** | **1** | Непосредственная | ADD #33 |
| **@** | **2** | Косвенная | ADD @33 |
| [ ] | **3** | Относительная | ADD [33] |
| @R | **4** | Косвенно-регистровая | ADD @R3 |
| @R+ | **5** | Индексная с постинкрементом | ADD @R3+ |
| -@R | **6** | Индексная с преддекрементом | ADD -@R3 |

Задания для выполнения:

1. Ознакомиться с теоретическими сведениями о реализации косвенной памяти в модельной ЭВМ.

2. Запустите программу CompModel.exe.

3. Выполнить следующий пример.

Изучить алгоритм работы программы вычисления суммы элементов массива чисел *С1,* С2,..., *Сn.* Исходными данными в этой задаче являются:

- *п* — количество суммируемых чисел

- *С1, С2,* .., *Сп* — массив суммируемых чисел.

Заметим, что должно выполняться условие *п >* 1, т. к. алгоритм предусматривает, по крайней мере, одно суммирование. Кроме того, предполагается, что суммируемые числа записаны в ОЗУ подряд, т. е. в ячейки памяти с последовательными адресами. Результатом является сумма *S.*

Рассмотрим программу для вычисления суммы со следующими конкретными параметрами:

* число элементов массива— 10;
* элементы массива расположены в ячейках ОЗУ по адресам 040, 041, 042,..., 049.

Используемые для решения задачи промежуточные переменные имеют следующий смысл:

- *Аi* — адрес числа *Сi*, *i*  {l, 2,..., 10};

- ОЗУ(*Ai*) — число по адресу *Ai,*

*- S* —текущая сумма;

*к* — счетчик цикла, определяющий число повторений.

Распределение памяти таково. Программу разместим в ячейках ОЗУ, начиная с адреса 000, примерная оценка объема программы — 20 команд; промежуточные переменные:

- *Аi* — в ячейке ОЗУ с адресом 030;

- *к* — по адресу 031;

- *S* — по адресу 032.

Граф-схема алгоритма (ГСА) программы, показана на рис. 2, текст программы с комментариями приведен в табл. 3.

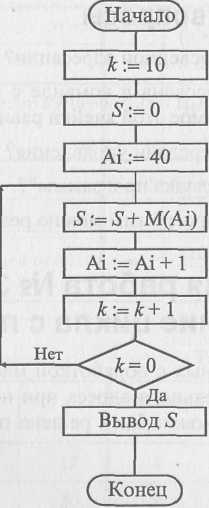


Рис.2. Граф-схема алгоритма

**Таблица 3.** Текст программы примера

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Адрес** | **Команда** | | **Примечание** | |
| 000 | RD #40 | | Загрузка начального адреса массива 040  в ячейку 030 | |
| 001 | WR 30 | |
| 002 | | RD #10 | | Загрузка параметра цикла *к* = 10 в ячейку 031 |
| 003 | | WR 31 | |
| 004 | | RD #0 | | Загрузка начального значения суммы *S* = 0 в ячейку 032 |
| 005 | | WR 32 | |
| 006 | | Ml: RD 32 | | Добавление к текущей сумме очередного элемента массива |
| 007 | | ADD @30 | |
| 008 | | WR 32 | |
| 009 | | RD 30 | | Модификация текущего адреса массива (переход к следующему адресу) |
| 010 | | ADD #1 | |
| 011 | | WR 30 | |
| 012 | | RD 31 | | Уменьшение счетчика (параметра цикла) на 1 |
| 013 | | SUB #1 | |
| 014 | | WR 31 | |
| 015 | | JNZ Ml | | Проверка параметра цикла и переход при *к**0* |
| 016 | | RD 32 | | Вывод результата |
| 017 | | OUT | |
| 018 | | HLT | | Стоп |

4. Разработать программу определения заданной характеристики последовательности чисел *C1*, *С2,*..., *Сп* , n <=10.

4.1. Варианты заданий приведены в табл. 4.

4.2. Записать программу в мнемокодах, введя ее в поле окна **Текст программы.**

4.3. Сохранить набранную программу в виде текстового файла и произвести ассемблирование мнемокодов.

4.4. Загрузить в ОЗУ необходимые константы и исходные данные.

4.5. Отладить программу.

Таблица 4. Варианты задания

|  |  |
| --- | --- |
| **Номер варианта** | **Характеристика последовательности чисел *С1, С2,..* … *Сn*** |
| **1** | Определить количество четных чисел в последовательности |
| **2** | Определить номер (индекс) минимального числа |
| **3** | Вычислить произведение всех чисел последовательности |
| **4** | Определить номер первого отрицательного числа |
| **5** | Определить количество чисел, равных *С1* |
| **6** | Определить количество отрицательных чисел |
| **7** | Определить максимальное отрицательное число |
| **8** | Определить номер первого положительного числа |
| **9** | Определить минимальное положительное число |
| **10** | Определить номер максимального числа |

**Примечание.** Под четными (нечетными) элементами массивов понимаются элементы массивов, имеющие четные (нечетные) индексы. Четные числа — элементы массивов, делящиеся без остатка на 2.

**Содержание отчета**

Включите в отчет:

1. Формулировку варианта задания.

2. Граф-схему алгоритма решения задачи.

3. Распределение памяти (размещение в ОЗУ переменных, программы и необходимых констант).

4. Программу.

5. Значения исходных данных и результат выполнения программы.

**5. Оформление результатов.**

Оформите результаты выполнения данной работы в виде отчета (в электронном виде), в соответствии с требованиями.