

НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ "СЕМИКО"

40 1340

КЛАВИШНАЯ ЭЛЕКТРОННО-ВЫЧИСЛИТЕЛЬНАЯ МАШИНА

ЭЛЕКТРОНИКА МК

ОРГАНИЗАЦИЯ ХРАНЕНИЯ И ОБРАБОТКИ
ИНФОРМАЦИИ В ОПЕРАТИВНОЙ ПАМЯТИ

НПКД.401348.001 Д8 изм. 1

Новосибирск

2014

Содержание

1. Введение.....	3
2. Стек.....	3
3. Регистры памяти.....	4
4. Типы адресации к регистрам памяти.....	5
5. Память программ.....	6
6. Типы адресации к памяти программ.....	7
7. Адресация памяти с использованием шестнадцатеричных значений.....	9

Настоящий документ распространяется на клавишные электронно-вычислительные машины "ЭЛЕКТРОНИКА МК" различных моделей (далее - ЭВМ).

При ознакомлении с документом дополнительно следует использовать руководство по эксплуатации ЭВМ (РЭ).

1. ВВЕДЕНИЕ

1.1. ЭВМ оперирует с числами, представленными в десятичной позиционной системе счисления в естественной форме или в форме с плавающей запятой.

1.2. Для обработки чисел в ЭВМ принята обратная бесскобочная логика вычислений, предложенная польским математиком Яном Лукасевичем. Ее характерными особенностями являются:

- отсутствие скобок в записи выражений;
- отсутствие приоритетов операций;
- выполнение операций в порядке записи;
- возможность выполнения подряд нескольких операций;
- использование стека для хранения исходных значений и результатов операции;
- ввод чисел в стек в порядке, обратном их использованию.

1.3. Информация в процессе обработки (данные) и алгоритмы её обработки (программы) сохраняются в оперативной памяти ЭВМ. Оперативная память имеет отдельные области для хранения программ и данных.

2. СТЕК

2.1. Стекком называется группа регистров памяти магазинного типа. Последнее записанное в стек число считывается из него первым. Передвижение чисел в регистрах стека осуществляется автоматически.

Пример: Выражение $a \uparrow b \uparrow c \uparrow d + \times \div$, где символ \uparrow обозначает ввод числа в стек, эквивалентно алгебраическому выражению: $\frac{a}{b \cdot (c + d)}$.

2.2. Стек ЭВМ состоит из специальных регистров памяти: X, Y, Z, T. Регистры стека X и Y являются операционными. Регистры Z и T используются

для промежуточного хранения результатов вычислений. Регистр предыдущего результата X1 является вспомогательным и предназначен для хранения значения регистра X, существовавшего до выполнения операции. Далее регистр X1 будет рассматриваться как один из регистров стека.

Стек ЭВМ автоматически смещается при вводе чисел и при выполнении операций. В автоматическом режиме калькулятора содержимое всех регистров стека, кроме X1, выводится на индикатор.

При вводе числа или при чтении его из регистра памяти вводимое значение записывается в регистр X.

3. РЕГИСТРЫ ПАМЯТИ

3.1. Регистры памяти позволяют ЭВМ организовывать работу с информацией. Каждый регистр имеет собственный номер (рис. 1), который позволяет идентифицировать размещенные в памяти данные (регистры данных) или обращаться к функциям ЭВМ (регистры функций). В каждом регистре в определенный момент времени может храниться только одно значение.

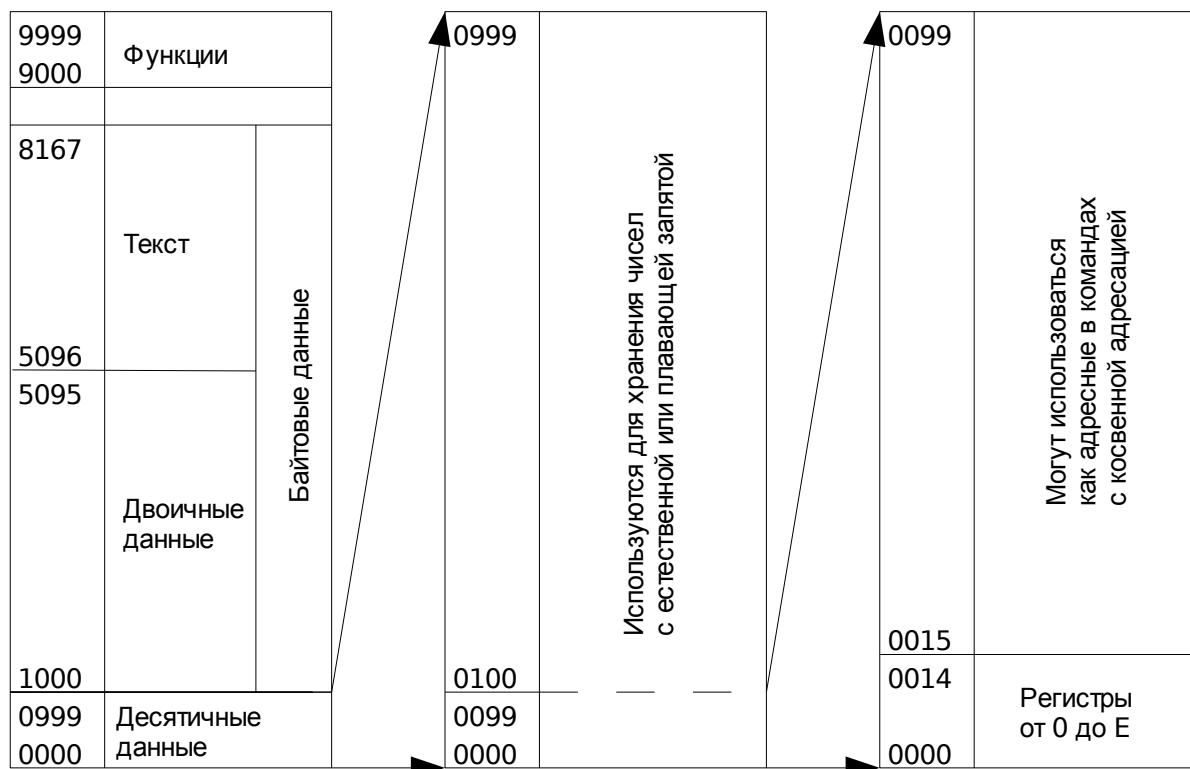


Рис. 1. Регистры памяти ЭВМ.

3.2. Регистры памяти с номерами от 0 до 999 используются для хранения десятичных данных, представленных в форме с естественной или плавающей запятой.

Данные из регистров с адресами от 0 до 999 могут быть сохранены на электронном диске, считаны с него или просмотрены как файл с десятичными данными. Содержимое регистров памяти может быть просмотрено в виде таблицы или очищено.

3.3. Регистры памяти с номерами от 1000 до 8167 используются для хранения байтовых данных. При записи в эти регистры чисел с естественной или плавающей запятой выполняется выделение целой части числа и его приведение к диапазону от 0 до 255. При считывании байтовые значения преобразуются в целые числа в указанном диапазоне.

Данные из регистров с адресами от 1000 до 5095 могут быть сохранены на электронном диске, считаны с него или просмотрены как файл с двоичными данными. Содержимое регистров памяти может быть просмотрено в виде таблицы или очищено.

Данные из регистров с адресами от 5096 до 8167 могут быть просмотрены в виде текста, а также сохранены на диске, считаны с него и просмотрены как текстовый файл.

3.4. Регистры памяти с номерами от 9000 до 9999 применяются для доступа к функциям ЭВМ.

4. ТИПЫ АДРЕСАЦИИ К РЕГИСТРАМ ПАМЯТИ

4.1. Регистровая адресация может быть выполнена к регистрам памяти с номерами от 0 до 14. При этом способе адресации номер используемого регистра является составной частью кода команды. При использовании в таких командах регистры с номерами от 10 до 14 имеют наименования А, В, С, D и Е соответственно.

4.2. Прямая адресация может быть выполнена ко всем регистрам памяти. При этом способе адресации номер используемого регистра записывается непосредственно после кода команды. Для прямого обращения к регистрам могут использоваться команды с диапазоном задания номера от 0 до 99 или от 0 до 9999, которые имеют различную длину.

4.3. Косвенная адресация также может быть выполнена ко всем регистрам памяти. Номер регистра, к которому производится косвенное обращение, записывается в адресные регистры с номерами от 0 до 99.

Содержимое адресных регистров при выполнении команд с косвенной адресацией изменяется. Если записанное в регистр число меньше нуля, в регистр записывается число нуль. Дробная часть числа, записанного в регистр, отбрасывается.

При косвенном обращении с регистровой адресацией через регистры с номерами от 0 до 6 перед обращением происходит дополнительное изменение содержимого этих регистров. При использовании регистров от 0 до 3 происходит косвенная адресация с предварительным автодекрементированием, то есть вычитание числа 1 из содержимого адресного регистра перед выполнением обращения к памяти. При использовании регистров от 4 до 6 - косвенная адресация с предварительным автоинкрементированием, то есть прибавление числа 1 к содержимому адресного регистра перед выполнением обращения к памяти. Обращение происходит после изменения содержимого регистра адресации в соответствии с полученным значением.

При косвенном обращении с прямой адресацией, в том числе при использовании регистров с номерами от 0 до 6, автоинкрементирования и автодекрементирования содержимого адресных регистров не происходит.

В дальнейшем описанное выше изменение содержимого регистра адресации при косвенном обращении называется модификацией.

4.4. Данные, введенные пользователем в регистры памяти с клавиатуры ЭВМ, сохраняются в энергонезависимой памяти и восстанавливаются после выключения и последующего включения питания. Также возможно сохранение содержимого регистров данных в энергонезависимой памяти средствами программы пользователя.

5. ПАМЯТЬ ПРОГРАММ

5.1. В ЭВМ принята десятичная система адресации памяти. Память программ состоит из 10000 ячеек, в каждую из которых записан один байт (целое число в диапазоне от 0 до 255 включительно). Программа состоит из отдельных команд. Каждая команда имеет длину от одного до трех байт.

5.2. Местонахождение команды в памяти программ определяется адресом, который может принимать значения от 0 до 9999. Для управления выполнением программы в ЭВМ имеется счетчик адреса. В нем хранится адрес текущей исполняемой или записываемой команды.

5.3. Часть команд переходов, циклов и подпрограмм использует страничную адресацию памяти программ. При использовании механизма страничной адресации память разбита на 100 страниц по 100 байт на каждой странице. Запись программ на электронный диск и считывание с него также выполняется постранично.

5.4. Программа может занимать любые адреса в памяти и быть запущена с произвольного адреса. Команды программы выполняются последовательно в порядке возрастания адресов. Команды переходов, циклов и подпрограмм изменяют значение счетчика адреса, что приводит к изменению последовательности выполняемых команд.

5.5. Память программ ЭВМ циклически замкнута. После выполнения команды по адресу 9999 происходит автоматический переход к адресу 0000. Переход к адресам памяти вне указанного диапазона вызывает останов программы и выдачу сообщения об ошибке.

6. ТИПЫ АДРЕСАЦИИ К ПАМЯТИ ПРОГРАММ

6.1. Адресация в командах переходов, включая команды циклов и подпрограмм, может быть прямая и косвенная. В командах прямой адресации используются короткие или расширенные адреса, занимающие соответственно один или два байта в программе.

6.2. При использовании прямой короткой адресации происходит переход внутри страницы, на адрес от 0 до 99, указанный в команде. Текущий номер страницы при этом не изменяется.

Примечание: Текущий номер страницы определяется расположением в памяти программ байта с адресом перехода. Если команда расположена на границе двух страниц, переход будет выполнен на адрес, расположенный в пределах страницы с большим номером.

Использование прямой короткой адресации позволяет создавать перемещаемые программы и подпрограммы, которые могут быть исполнены после загрузки в любую страницу памяти.

Примечание. Таким образом обеспечивается совместимость ЭВМ с существующим программным обеспечением, разработанным для программируемых калькуляторов БЗ-34, МК-52, МК-61 и других аналогичных. В памяти ЭВМ может одновременно храниться до 100 программ данного типа и любая из них может быть запущена на выполнение.

6.3. При использовании команд прямой расширенной адресации возможен переход на произвольный адрес в диапазоне от 0 до 9999. При этом происходит изменение текущего номера страницы. Например, при использовании в командах перехода адресов от 0000 до 0099 происходит переход к адресам на нулевой странице.

6.4. Команды, использующие косвенную адресацию, обеспечивают переход на произвольный адрес в диапазоне от 0 до 9999. Для хранения адреса используются регистры с номерами от 0 до 99. При косвенном обращении содержимое этих регистров модифицируется так же, как и при косвенной адресации регистров памяти.

6.5. В памяти программ могут храниться неизменяемые данные пользователя, такие как текстовые строки, графические изображения, команды для устройств, или любые другие, которые могут быть представлены в виде последовательности байтов. Байтовые значения считываются из памяти программ в стек при помощи специальной команды или с использованием регистров функций. Запись или изменение значений в памяти программ средствами программы пользователя возможно только при помощи дисковой операционной системы.

7. АДРЕСАЦИЯ ПАМЯТИ С ИСПОЛЬЗОВАНИЕМ ШЕСТНАДЦАТЕРИЧНЫХ ЗНАЧЕНИЙ

7.1. В памяти программ хранятся байты, соответственно коды команд и адреса представлены в виде шестнадцатеричных чисел. Значения десятичных адресов и номеров регистров в командах с прямой адресацией хранятся в программе в одном или двух байтах в упакованном BCD формате. При этом в записи десятичных значений адресов и номеров регистров используются только те шестнадцатеричные числа, цифры в которых принимают значения от 0 до 9.

Если в команде перехода или при обращении к регистру в поле команды с прямой адресацией тем или иным способом записано шестнадцатеричное значение с цифрами от A до F, то адрес перехода или номер регистра вычисляются особым образом.

Значение шестнадцатеричной цифры умножается на вес десятичного разряда, после чего полученные для каждой цифры значения складываются, образуя десятичный эквивалент записанного в команде значения.

Например, шестнадцатеричное значение 0ABh эквивалентно десятичному 111, поскольку $0Ah \cdot 10 + 0Bh = 10 \cdot 10 + 11 = 100 + 11 = 111$. Здесь и далее символ 'h' после числа указывает на шестнадцатеричное представление.

Если полученное значение эквивалентно допустимому десятичному адресу или номеру регистра, то действия команд с использованием шестнадцатеричного и эквивалентного десятичного значения не отличаются.

При выходе значения адреса за границу 9999 при выполнении команды перехода формируется сообщение об ошибке. При обращении к номеру регистра свыше 9999 никаких действий не выполняется.

7.2. Применение команд с прямой короткой адресацией со значениями адреса от 9Ah до 0FFh позволяет выполнять переход на адреса, расположенные на следующей странице. Применение команд с косвенной адресацией с аналогичными значениями номера регистра позволяет использовать в качестве адресных регистры с номерами от 100 до 165.

Ввод шестнадцатеричных адресов и номеров регистров с клавиатуры при наборе команды в режиме программирования запрещен. Для занесения в память шестнадцатеричных значений следует перейти в режим ввода байтов.

Примечание. Не следует использовать в программах данную особенность без необходимости.