ЛАБОРАТОРНАЯ РАБОТА № 5.

**Тема: *«СОСТАВЛЕНИЕ УПРОЩЕННОЙ СХЕМЫ ВЫПОЛНЕНИЯ ПРОЦЕССОРОМ I8086 МАШИННЫХ КОМАНД ПРОГРАММЫ».***

**Цель:**

* научиться читать состояния функциональных регистров процессора и распределять память под нужды программы;
* научиться анализировать этапы выполнения машинных команд программы;
* понять назначение префикса программного сегмента (PSP).

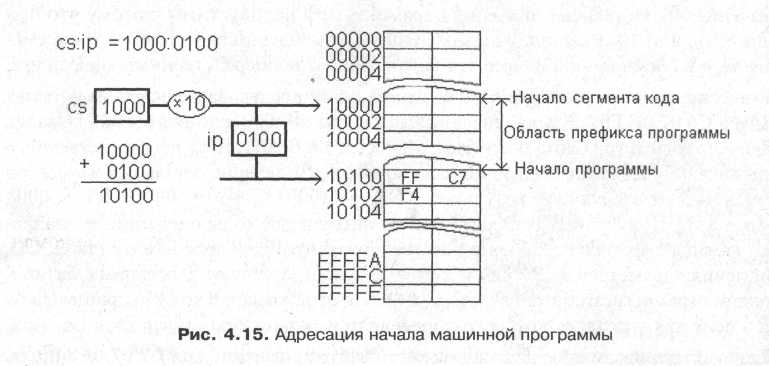
**ХОД РАБОТЫ:**

* 1. Повторить приведенный ниже теоретический материал.
  2. Выделить основные этапы рабо­ты процессора при выполнении машинной команды.
  3. Аналогично приведенному примеру, проанализировать этапы работы процессора по выполнению команд своего варианта.
  4. В качестве отчета по работе подготовить презентацию, содержащую отдельные слайды по каждому этапу выполнения команд. Слайды, помимо текста, должны содержать аналогичные рисунки.
  5. Ответить на контрольные вопросы.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**Схема работы процессора при выполнении машинной команды**

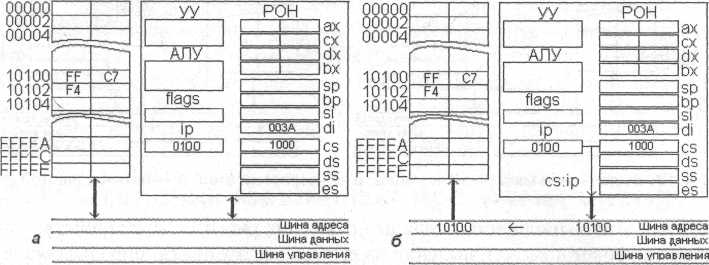
Рассмотрим упрощенную схему выполнения процессором машинных команд программы. Эта схема на логическом уровне адекватна реальной работе процес­сора. Отличие состоит в том, что в реальном процессоре с помощью различных технических приемов организуется параллельное, то есть одновременное выпол­нение нескольких этапов одной и той же команды или нескольких последова­тельных команд программы, за счет чего добиваются значительного ускорения работы процессора.



Перед запуском программы на выполнение операционная система автоматиче­ски записывает в регистр CS адрес сегмента кода, который, как указывалось ранее, содержит код выполняющейся программы. Допустим, что операционная система занесла в регистр CS код 100016. Первые 25610 байтов сегмента кода опе­рационная система всегда занимает вспомогательной информацией. Этот участок памяти принято называть **префиксом программного сегмента,** часто встречается также его обозначение **PSP** (от program segment prefix). Поэтому первый байт программы всегда имеет внутрисегментное смещение 010016 = 25610. Как из­вестно, для задания адресов в сегменте кода используется пара регистров CS:IP. Следовательно, операционная система перед началом выполнения программы должна записать в регистр IP код **010016.** На рис. 4.15 изображена схема задания адреса первого байта программы при сделанном относительно содержимого ре­гистра CS предположении.

На рис. **4.16, *а***слева изображена программная модель оперативной памяти, а спра­ва — программная модель процессора в состоянии, которое будем считать ис­ходным. Пусть в оперативной памяти в поле с адресом 1010016 находятся коды **FFС7F416**, которые будем рассматривать как машинную программу, а ре­гистры CS и IP содержат выбранные ранее коды 100016 и 010016 соответственно. Из всех остальных регистров процессора в данном примере нас интересует толь­ко содержимое регистра DI, которое примем равным, например, **003А16.** Рабо­ту процессора при выполнении машинной команды удобно разбить на несколько этапов.

***Первый этап*** выполнения команды изображен на рис. 4.16, *б.* Указатель из пары регистров CS:IP определяет адрес, который всегда рассматривается процессором как **адрес первого байта текущей команды выполняемой программы**. В момент запуска программы этот указатель, очевидно, представляет адрес ее первой ко­манды. Он выбирается из пары CS:IP, выставляется на адресную шину и переда­ется в оперативную память.



*Рис. 4.16. Выполнение машинной команды: а — исходное состояние; б — первый этап*

На ***втором этапе*** работы процессора содержимое поля, начинающегося с полу­ченного оперативной памятью адреса, передается по шине данных в устройство управления процессора. Поскольку это содержимое рассматривается процессо­ром как код машинной команды, возникает вопрос: сколько байтов поля должно быть передано в устройство управления?

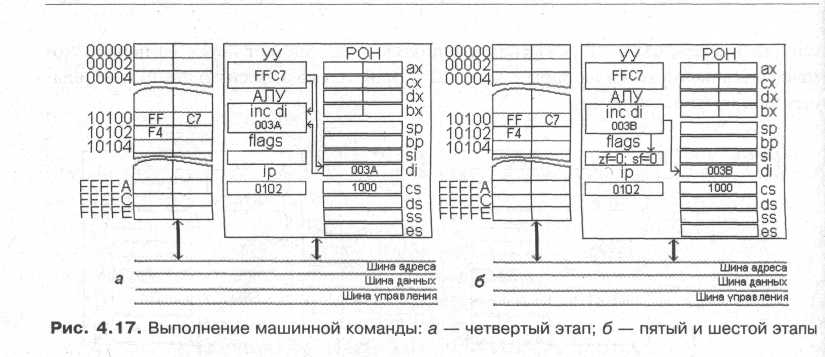
Известно, что в системе команд про­цессора i8086 машинная команда может иметь длину от 1 до 6 байтов. Поэтому в устройство управления передается последовательность не менее чем из 6 байтов. Код из первого байта сравнивается с возможными значениями байта ***префикса замены сегмента***(2616, 2Е16, 3616 и ЗЕ16). Если этот код не является префиксом, он считается байтом команды, в котором находится код операции и по которому определяется ее общая длина. В противном случае информация о длине команды выбирается из следующего байта.

Затем из переданной последовательности для анализа отбирается только необходимое количество байтов, соответствую­щее длине команды.

Следует обратить внимание на то, что байты кода машинной команды передают­ся в устройство управления в том же порядке, в котором они находятся в опера­тивной памяти. *Обратный порядок записи на коды машинных команд не распро­страняется.* Сфера действия обратного порядка записи — это коды данных, над которыми требуется выполнять арифметические операции, коды чисел в форма­тах с фиксированной или плавающей точкой, коды, используемые для формиро­вания адресов, и т. д.

На ***третьем этапе*** устройство управления выполняет описанный ранее анализ, определяет длину команды, выделяет из переданной последовательности байтов код команды, а затем увеличивает содержимое регистра IP на ее длину. В рас­сматриваемом примере содержимое IP увеличивается на 2 и становится равным 010216, после чего в регистре IP оказывается адрес (точнее, внутрисегментное смещение) для следующей команды программы.

***Четвертый этап*** выполнения команды показан на рис. ***4.17, а****.* Устройство управ­ления дешифрует команду, выделяя из ее кода все элементы, определяющие вы­полняемое действие, длину и адреса операндов. Затем в арифметико-логическое устройство передается информация об операции, которую необходимо выполнить, и по заданным в команде адресам запрашиваются операнды, которые также переда­ются в арифметико-логическое устройство.

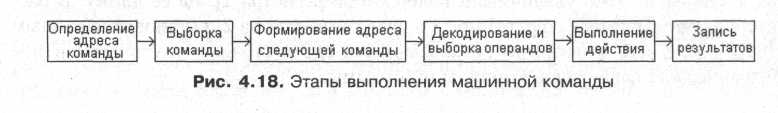


В рассматриваемом примере код **FFС716**(**INC DI**) задает выполнение инкремента содержимого регистра DI. Устройство управления настраивает арифметико-логическое устройство на выполнение ука­занного действия и организует пересылку в него операнда 003А16 из регистра DI.

В рассмотренном примере операнд команды находится в регистре процессора, что значительно упрощает ее выполнение. Если операнд находится в поле опе­ративной памяти, то устройство управления сначала передает на шину адрес операнда, затем оперативная память по шине данных возвращает в арифметико-логическое устройство код запрошенного операнда.

Очевидно, что эти допол­нительные действия значительно снижают скорость выполнения команды по сравнению с вариантом, когда операнд находится в регистре процессора. Поэто­му следует ***всегда, когда это возможно, использовать регистровую или непосред­ственную адресацию* и как можно реже обращаться за операндами в оперативную память.**

Результаты выполнения пятого и шестого этапов изображены на рис. 4.17, б. На пятом этапе арифметико-логическое устройство выполняет заданное в команде действие. Его результатом является код 003В16.



И на последнем, ***шестом этапе*** организуется определенная в команде пересылка результата и формируется соот­ветствующее ему содержимое регистра флагов. В данном примере код 003В16 пе­ресылается в регистр DI. Так как получен ненулевой код, то флаг **zf** получает нулевое значение, zf = 0. Кроме того, код 003В16 не может трактоваться как код *отрицательного числа, поэтому и флаг* ***sf***получает нулевое значение, sf = 0.

На рис. 4.18 представлена рассмотренная последовательность этапов выполне­ния машинной команды процессором. Эта последовательность этапов работы про­цессора в точности повторяется в ходе выполнения следующей машинной коман­ды. По адресу, определяемому указателем из пары регистров CS:IP, из оперативной памяти в устройство управления выбирается машинная команда. Команда дешиф­руется, определятся ее длина и соответствующим образом увеличивается содер­жимое регистра IP. После этого команда выполняется, и результаты ее выполнения заносятся в заданное место хранения. Этот циклический процесс выполнения машинных команд обрывается при поступлении в процессор команды останова с кодом **F416**. В данном примере она попадает в процессор уже на втором шаге.

***КОНТРОЛЬНЫЕ ВОПРОСЫ.***

1. Назовите действия, выполняемые процессором на первом, втором, третьем, четвертом, пятом и шестом этапах выполнения команды.
2. Что называют префиксом программного сегмента? Где он располагается? Каков его размер?
3. На каком этапе выполнения команды происходит установка флагов состояния? Какую ситуацию отражает флаг OF=1?
4. Почему при адресации операндов предпочтительно использовать регистровую или непосредственную адресации?
5. Каков логический адрес команды?

***ЗАДАНИЯ НА ВЫПОЛНЕНИЕ:***

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | **895435h**  **F416** | 8. | 235F08h **F416** |
| 2. | **2B5205h**  **F416** | 9. | **8A4C12h**  **F416** |
| 3. | **8B4F2Fh**  **F416** | 10. | **0B550Eh**  **F416** |
| 4. | **018C2009h**  **F416** | 11. | **2B5212h**  **F416** |
| 5. | **035C1Ah**  **F416** | 12. | **218D3412h**  **F416** |
| 6. | **085F08h**  **F416** | 13. | **335D0Fh**  **F416** |
| 7. | **005D0Eh**  **F416** | 14. | **31906601h**  **F416** |

ПРИЛОЖЕНИЕ ДЛЯ ЛАБОРАТОРНОЙ РАБОТЫ №5

**Взаимодействие основных устройств ЭВМ**

**при автоматическом исполнении команды**

Рассмотрим автоматическое выполнение трехадресной команды с прямой адресацией. Выполнение такой команды состоит из следующих этапов. Допустим, в ОЗУ находится программа решаемой задачи. Фрагмент этой программы в виде команды представлен на структурной схеме ЭВМ (рис.5.1) справа. При каждом обращении к ОЗУ выбираются все поля команды. Все команды программы имеют одинаковую длину.

Рассмотрим этапы выполнения команды, код которой находится в ОЗУ в ячейке с адресом *k+i,* этот адрес зафиксирован на счётчике команд (СК).

Первый этап - выборка кода выполняемой команды из ОЗУ. Для реализации этого этапа необходимо код (СК) = *k+i* передать в ОЗУ, обратиться в ячейку ОЗУ с адресом *k+i*  и код этой ячейки, являющийся кодом этой команды, то есть (*k+i)* = ОП *abc*, передать на регистр команд (РК). Соответствующие передачи отражены на схеме рис.5.1. Они отмечены цифрой 1. Это передача кода СК на РА (регистр адреса ОЗУ), передача (*k+i*) из ОЗУ в РК. На этом этапе после приема кода команды на РК, коммутатор операций (КОП), дешифрируя операционную часть выполняемой команды, определяет тип команды.

Если команда не связана с изменением естественного порядка выполнения программы (не выполняются команды безусловного или условного переходов), то реализуется следующая последовательность этапов как продолжение первого.

Второй этап - выборка первого операнда – (*a*). Необходимо код поля адреса первого операнда - *a* из РК передать в ОЗУ, произвести обращение к ячейке с адресом *a* в оперативную память и код этой ячейки передать в АЛУ. Соответствующие передачи обозначены на схеме рис.5.1 цифрой 2. Это – передача кода *a* из РК на РА и передача (*a*) из ОЗУ в АЛУ.

Третий этап - выборка второго операнда – (*b*). Он аналогичен второму этапу. Соответствующие передачи на схеме рис.5.1 отмечены цифрой 3. Это – передача кода *b* из РК на РА и передача (*b*) из ОЗУ в АЛУ.

Четвёртый этап - выполнение операции в соответствии с кодом операции команды. Еще в конце первого этапа КОП определил тип выполняемой команды. Этот сигнал коммутирует оборудование блока управления операциями (БУОП) таким образом, что на выход БУОП проходят те сигналы с датчика сигналов (ДС), которые необходимы для автоматического выполнения операций в АЛУ. Соответствующие передачи и взаимодействия блоков обозначены на схеме рис.5.1 цифрой 4.

Пятый этап – обращение к ОЗУ и запись по адресу *c* результата операции. Здесь код поля *c* регистра команд передается в ОЗУ на РА. Затем в ячейку ОЗУ с адресом *c* записывается результат операции, находящийся в АЛУ. Соответствующие передачи обозначены на схеме рис.5.1 цифрой 5.

Шестой этап – формирование адреса ячейки ОП, где находится следующая команда программы. При естественном порядке выполнения программы необходимо изменить код в СК, то есть (СК)ст = (СК)нов.

Выполнение этого этапа может совмещаться с выполнением предшествующих этапов, что и реализовано в ряде ЭВМ.

На этом заканчивается выполнение команды: в СК сформирован адрес следующей команды *k+i*+1.Приведенная последовательность этапов повторяется и в дальнейшем для каждой из последующих команд программы, что и обеспечивает её автоматическое выполнение.

Если выполняются команды безусловного или условного переходов, то вышеизложенная последовательность этапов выполнения команды нарушается. Допустим, в конце выполнения первого этапа КОП зафиксировал выполнение команды безусловного перехода. Эту ситуацию можно представить так: (*k +i*) = БП *k + j,* то есть код выполняемой команды выбран из ячейки с адресом *k +i*, это команда безусловного перехода, которая должна передать управление на выполнение команды, находящейся по адресу *k + j*. В данном случае выполнение этапов со второго по четвертый блокируется, и выполнение команды БП заключается в передаче кода *k +i* из адресного поля РК в СК.

При команде условного перехода нарушение естественного порядка выполнения программы (то есть передача кода *k + j* в СК) происходит только при выполнении определенного условия. Это условие характеризует результат команды, полученный перед выполнением команды условного перехода. Таким условие может быть, например, отрицательный результат или результат, равный нулю, и т.д.

Изменим исходные данные (предпосылки). Рассмотрим автоматическое выполнение команд на ЭВМ со структурой, подобной структуре IBM PC. При таких условиях в состав ЭВМ, кроме традиционных устройств (оперативное запоминающее устройство, арифметико-логическое устройство и основных схем устройства управления), будут входить (рис.5.2):

* регистровая память (РП);
* блок формирования адреса операнда (БФАО);
* двадцатиразрядные сумматоры для получения физических адресов данных (ФАД) и физических адресов команд (ФАК).

В рассматриваемой структуре, в определенной степени, изменения коснутся регистра команд.

Состав и основные характеристики РП рассмотрены в разделе 2.2.. На входы РП (рис.5.3) поступают коды адресов выбираемых для записи или считывания регистров. Эти адресные коды поступают из БФАО. На входы выбираемых регистров поступают коды из оперативной памяти; коды, результаты выполнения операции, из АЛУ; коды сегментных регистров, устанавливаемых операционной системой ЭВМ и др. Выходные сигналы – это коды, считываемые из регистров. Они являются:

* составляющими эффективных адресов (ЕА) данных;
* кодами, считываемыми с сегментных регистров;
* кодами, считываемыми с регистров, где хранятся исходные коды, участвующие в выполнении операции.

**Регистровая память**

Коды,

записываемые

в регистр

Коды,

считываемые с регистров

Коды адреса выбираемых регистров

Рис.5.3. Регистровая память

На блок формирования адреса операнда (рис.5.4) поступают:

* сигналы от тактового генератора микропроцессора;
* коды полей второго байта выполняемой команды, находящейся в РК; эти коды определяют режимы адресации первого (поля *md* и *r/m*) и второго (поле *reg*) операндов;
* коды двух младших разрядов первого байта (коды *d* и *w*), которые определяют операнд-источник, операнд-получатель и разрядность (формат) кодов, участвующих в операции.

БФАО вырабатывает следующие выходные сигналы:

* коды адреса выбираемых регистров;
* сигналы считывания кодов с выбранных регистров (данные или составляющие EA данных);
* сигналы считывания смещений (*disp L* и *disp H*);
* сигналы считывания непосредственных данных (*data L* и *data H).*

БФАО – это нерегулярная структура. В своей основе этот блок состоит из логических схем, элементов хранения, дешифраторов. Коды с полей РК так коммутируют оборудование блока, что на выход пропускаются те сигналы ТГ, которые обеспечивают выбор регистров РП, считывание составляющих эффективного адреса данных, считывание непосредственных данных.

Перейдём к рассмотрению взаимодействия основных устройств ЭВМ и блоков (схем) управления при автоматическом выполнении команды. Обратимся к рис.5.2, где изображены все основные устройства, которые будут участвовать в автоматическом выполнении команды.

Основные этапы выполнению команды остаются без изменения, а именно:

* первый этап – выбор кода команды;
* второй и третий этапы – выбор операндов;
* четвертый этап – выполнения операции в АЛУ;
* пятый этап – засылка результата в оперативную или регистровую память;
* шестой этап – формирование основной составляющей адреса ОЗУ, где находится код следующей выполняемой команды.

Но содержание этих этапов изменилось.

Рассмотрим выполнение вышеуказанных этапов при условии, что

* выполняется двухоперандная команда;
* для первого операнда используется относительный регистровый, а для второго прямой регистровый режим адресации;
* длина команды равна трём байтам;
* выполняется команда сложения *ADD AL,[BX+disp*];
* код *dw* = 10 (младшие разряды первого байта).При этом второй операнд является приёмником, и в операции сложения участвуют коды в формате байта.

Допустим, что принята для выполнения команда *ADD AL,[BX+dispI*], ее первый байт находится в ячейке ОЗУ с адресом *i* + 24[*CS*], тогда [*PC*] = *i.*

Рассмотрим выполнение первого этапа. Код *PC*, то есть [*PC*] = *i,* передаётся на сумматор ФАК (∑ФАК). На этот же сумматор поступает код регистра сегмента команд 24[*CS*] из РП. На выходе ∑ФАК сформируется код физического адреса ОЗУ, по которому находится первый байт команды. Код с выхода ∑ФАК поступает на РА ОЗУ. Выбирается из ОЗУ первый байт команды и посылается в РК. И в завершении этого этапа к *PC* добавляется 1.

Все указанные взаимодействия отметим на схеме знаком 11. Указанные взаимодействия будут повторены еще два раза для выбора второго и третьего байтов выполняемой команды. Эти повторные взаимодействия отмечены на схеме знаками 12 и 13.

Второй этап. В начале второго этапа с помощью КОП расшифровывается код операции выполняемой команды. Если выполняемая команда не нарушает естественного порядка выполняемой программы, то осуществляется формирование адреса первого операнда и выборка этого операнда из РП или ОЗУ ЭВМ.

В соответствии с принятыми условиями для первого операнда используется регистровый относительный режим адресации. Соответственно эффективный адрес *EA*= [*BX*] + *disp*. В этом случае коды полей *md* и *r/m* второго байта регистра команд поступают в БФАО и так коммутируют оборудование БФАО, что на его выходе появляются сигналы, обеспечивающие считывание:

* кода регистра *BX*;
* кода *disp L*;
* кода с сегментного регистра *DS*.

Все указанные коды поступают на сумматор физического адреса данных (∑ФАД). Выход ∑ФАД поступает на РА ОЗУ. Происходит выборка байта данных, который направляется в АЛУ. Выполнение второго этапа завершено. Все указанные взаимодействия устройств отметим на схеме цифрой 2.

Третий этап - этап выбора второго операнда. В данном случае БФАО под воздействием сигнала с разряда *w* и поля *reg* регистра команд выдает сигнал обращения к регистру *AL*, код которого подается в АЛУ. Все взаимодействия этого этапа отметим цифрой 3.

Четвертый этап. Выполнение операции сложения в АЛУ. Здесь блок управления операциями выдает те сигналы в АЛУ, которые необходимы для выполнения операции. Линии взаимодействия отметим цифрой 4.

Пятый этап. Код выполненной операции из АЛУ направляется в регистр *AL* (*d*=1) регистровой памяти. Взаимодействие отмечается цифрой 5.

Команда выполнена. В *PC* находится основная составляющая адреса следующей команды программы – (*PC*) = *i*+ 3.Здесь шестой этап как отдельный (автономный) этап исключен. Формирование основной составляющей адреса следующей выполняемой команды было реализовано при выполнении первого этапа.

ЭВМ готова к выполнению следующих команды программы.