1. **Общая характеристика цепочечных команд.**

Система команд процессора имеет группу команд, позволяющих производить действия над блоками элементов (цепочками). Логически цепочки представляют собой последовательности элементов с любыми значениями (символы, числа и прочее), хранящихся в соседних ячейках памяти в виде двоичных кодов. Единственное ограничение состоит в том, что размеры элементов в цепочках фиксированы значениями байт, слово или двойное слово. Команды обработки строк обеспечивают возможность выполнения ряда операций-примитивов, обрабатывающих цепочки поэлементно. Каждая операция-примитив реализуется в процессоре тремя командами, в свою очередь, каждая из этих команд работает с соответствующим размером элемента – байтом, словом или двойным словом. Кроме того, имеется одна команда с явным указанием операндов, служащих только для определения типов операндов. После того как выяснен тип элементов цепочек по их описанию в памяти, генерируется одна из трех машинных команд. Все цепочечные команды обрабатывают только один (текущий) элемент цепочки и автоматически передвигаются к следующему элементу. Адреса текущих обрабатываемых элементов всегда находятся в регистрах ESI для строки, участвующей в команде в качестве источника, и EDI для строки, участвующей в команде в качестве приемника. Переход к следующему элементу производится увеличением или уменьшением содержимого регистров ESI, EDI. Возможны два направления обработки цепочки: – от начала цепочки к ее концу, то есть в направлении возрастания адресов; – от конца цепочки к началу, то есть в направлении убывания адресов. Направление определяется значением флага направления DF в регистре EFLAGS/FLAGS: – если DF = 0, то значения индексных регистров ESI, EDI будут автоматически увеличиваться цепочечными командами, то есть обработка будет осуществляться в направлении возрастания адресов; – если DF = 1, то значения индексных регистров ESI, EDI будут автоматически уменьшаться цепочечными командами, то есть обработка будет идти в направлении убывания адресов. Состоянием флага DF можно управлять с помощью двух команд, не имеющих операндов: – CLD – очистить флаг направления (команда сбрасывает флаг направления DF в 0); – STD – установить флаг направления (команда устанавливает флаг направления DF в 1). Количество байтов, на которые изменяется содержимое регистров ESI, EDI определяется типом элементов строки или кодом цепочечной команды: на 1 при обработке байтов, на 2 при обработке слов и на 4 при обработке двойных слов. 15 Поскольку адреса текущих обрабатываемых элементов извлекаются командами из регистров ESI и EDI, перед выполнением команд необходимо нужные адреса в эти регистры поместить. Обычно перед началом обработки строк в указанные регистры заносятся адреса первых обрабатываемых элементов. Для этого могут использоваться команда LEA или уже использовавшаяся нами ранее операция OFFSET. Команда LEA (загрузить исполнительный адрес) пересылает смещение ячейки памяти в 32-битовый регистр. Она имеет формат LEA регистр, память

1. **Префиксы повторения.**

Логически к командам обработки строк нужно отнести и так называемые префиксы повторения. Префиксы повторения имеют мнемонические обозначения REP, REPE (REPZ), REPNE (REPNZ). Префиксы повторения указываются перед нужной цепочечной командой в поле метки. Цепочечная команда без префикса выполняется один раз. Размещение префикса перед цепочечной командой заставляет ее выполняться в цикле. Число повторений заносится в регистр ЕСХ. После очередного выполнения команды значение в регистре СХ уменьшается на единицу. Решение об очередном выполнении цепочечной команды принимается по состоянию регистра ЕСХ и, возможно, по флагу нуля ZF. Префикс повторения REP (REPeat) заставляет команды выполняться, пока содержимое в ЕСХ не станет равным 0. Префиксы повторения REPE (REPeat while Equal) и REPZ (REPeat while Zero) являются синонимами. Они заставляют цепочечную команду выполняться до тех пор, пока содержимое ЕСХ не равно 0 и флаг ZF равен 1. Как только одно из этих условий нарушается, управление передается следующей команде программы. Префиксы REPNE и REPNZ, также являющиеся синонимами, заставляют цепочечную команду циклически выполняться до тех пор, пока содержимое ЕСХ не равно 0 и флаг ZF равен 0. При нарушении одного из этих условий работа команды прекращается

1. **Пересылка цепочек.**

Синтаксис команды MOVS: MOVS адрес\_приемника, адрес\_источника Команда копирует элемент цепочки-источника (длиной в байт, слово или двойное слово), который располагается по адресу, соответствующему содержимому регистра ESI, в элемент цепочки-приемника, который располагается по адресу, соответствующему содержимому регистра EDI. Команды MOVSB, MOVSW и MOVSD выполняют аналогичные действия. При этом первая команда оперирует элементами длиной в байт, вторая – элементами длиной в слово, третья – элементами длиной в двойное слово. Последовательность действий, которые нужно выполнить в программе для того, чтобы переслать цепочку элементов из одной области памяти в другую с помощью команды MOVS следующая: 1. Установить значение флага DF в зависимости от того, в каком направлении будут обрабатываться элементы цепочки – в направлении возрастания или убывания адресов. 16 2. Загрузить адреса первых обрабатываемых элементов цепочек в регистры ESI и EDI. 3. Загрузить в регистр ЕСХ количество обрабатываемых элементов. 4. Записать команду MOVS с префиксом REP. Пример. Пусть необходимо скопировать 100 элементов из строки SOURCE в строку DEST, обе строки описаны сегменте данных как массивы байтов. CLD LEA ESI, SOURCE LEA EDI, DEST MOV ECX, 100 REP MOVSB

**4. Сравнение цепочек.**Команды, реализующие операциюпримитив сравнения цепочек, производят сравнение элементов цепочки-источника с элементами цепочки-приемника. Синтаксис команды CMPS: CMPS адрес\_приемника, адрес\_источника Алгоритм работы команды CMPS заключается в вычитании элемента цепочки-приемника, который располагается по адресу, соответствующему содержимому регистра EDI, из элемента цепочки-источника, который располагается по адресу, соответствующему содержимому регистра ESI. Команда производит вычитание элементов, не записывая при этом результата, и устанавливает флаги ZF, SF и OF. Так, если флаг нуля ZF равен 1, то элементы оказались равны, а если ZF равен 0, то, наоборот, не равны. Чтобы заставить команду CMPS выполняться несколько раз, то есть произвести последовательное сравнение элементов цепочек, необходимо перед командой CMPS определить префикс повторения. С командой CMPS удобно использовать префиксы повторения REPE/REPZ или REPNE/REPNZ для поиска первой пары несовпадающих или, наоборот, совпадающих элементов. Поскольку при выполнении команды CMPS значения регистров ESI и EDI изменяются, по завершении цикла они указывают не на искомые элементы в цепочке, а на следующие, которые должны были бы сравниваться. Ассемблер преобразует команду CMPS в команду CMPSB (при сравнении байтов), в команду CMPSW (при сравнении слов) или в команду CMPSD (при сравнении двойных слов) в зависимости от описания операндов.

**5. Сканирование цепочек.**Команды, реализующие операцию-примитив сканирования цепочек, производят поиск некоторого значения в цепочке. Искомое значение предварительно должно быть помещено в один из регистров AL, AX или EAX (в зависимости от размера элемента цепочки). Синтаксис команды SCAS: SCAS адрес\_приемника Принцип поиска здесь тот же, что и в команде сравнения CMPS, то есть вычитание элемента цепочки-приемника, который располагается по адресу, соответствующему содержимому регистра EDI, из содержимого регистра AL/AX/EAX. В зависимости от результатов вычитания производится установка флагов, при этом сами операнды не изменяются. 17 Так же как и в случае команды CMPS, с командой SCAS удобно использовать префиксы REPE/REPZ или REPNE/REPNZ. Ассемблер преобразует команду SCAS в команду SCASB (при поиске байта), в команду SCASW (при поиске слова) или в команду SCASD (при поиске двойного слова).

**6. Загрузка элемента цепочки в регистр.**команда LODS. Операцияпримитив загрузки элемента цепочки в регистр позволяет скопировать элемент цепочки в регистр AL, АХ, ЕАХ (в зависимости от размера элемента цепочки). Синтаксис команды LODS: LODS адрес\_источника Работа команды заключается в том, чтобы извлечь элемент из цепочки по адресу, соответствующему содержимому регистра ESI, и поместить его в регистр AL/АХ/ЕАХ. Ассемблер преобразует команду LODS в команду LODSB (при работе с цепочкой байтов), в команду LODSW (при работе с цепочкой слов) или в команду LODSD (при работе с цепочкой двойных слов).

**7. Сохранение содержимого регистра в цепочке.**команда STOS. Операцияпримитив сохранения содержимого регистра в цепочке позволяет произвести действие, обратное действию команды LODS, то есть перенести значение из регистра в элемент цепочки. Синтаксис команды STOS: STOS адрес\_приемника Команда пересылает элемент из регистра AL/AX/EAX в элемент цепочки по адресу, соответствующему содержимому регистра EDI. Ассемблер преобразует команду STOS в команду STOSB (при работе с цепочкой байтов), в команду STOSW (при работе с цепочкой слов) или в команду STOSD (при работе с цепочкой двойных слов).