

Строковые команды

Крайне рекомендуется дополнительно прочитать про строковые команды в методичке Бахтизина стр. 17 – 25.

Фирма Intel, идя по пути разбиения множества команд процессора на обособленные группы, выделила т. н. «строковые» команды в особую группу. Отличие этих команд, называемых также «цепочечными», от остальных состоит в следующем:

- только в них используется автоинкрементная/автодекрементная адресация
- только в них возможно расположение обоих операндов в памяти (M – M)
- только в них по умолчанию используется сегментный регистр ES
- применяется неявная адресация с использованием регистров SI, DI

Каждая строковая операция представлена в процессоре **двумя** видами команд, различающихся по последнему символу мнемоники команды:

- B (byte) – для обработки строк состоящих из символов-байтов (как в данной лабораторной работе);

- W (word) – для обработки строк состоящих из символов-слов.

Если флаг направления DF перед выполнением команды строковой обработки установлен в нуле (выполнена команда **CLD**), то значение в индексном регистре автоматически увеличивается, если в единице (выполнена команда **STD**) – уменьшается. Индексные регистры уменьшаются или увеличиваются на единицу, если команды работают с байтами, или на два – при работе со словами.

Строковые операции обеспечивают выполнение следующих операций:

- сравнение строк (**CMPS**) – команда сравнивает значение элемента одной строки (DS:SI) со значением элемента второй строки (ES:DI) и устанавливает значения индексных регистров на следующие элементы строк. Сравнение происходит так же, как и по команде сравнения CMP. Результатом операции является установка флагов;

- сканирование строки (**SCAS**) – команда производит сравнение содержимого аккумулятора (AL или AX) с байтом или словом памяти, абсолютный адрес которого определяется парой ES:DI, после чего регистр DI устанавливается на следующий символ. Команда SCAS используется обычно для поиска в строке (ES:DI) элемента, заданного в аккумуляторе;

- пересылка строки (**MOVS**) – пересылает поэлементно строку DS:SI в строку ES:DI и устанавливает значения индексных регистров на следующий элемент строки;

- запись в строку (**STOS**) – заполняет строку, содержащуюся по адресу ES:DI, элементом из аккумулятора (AL или AX), не влияет на флаги;

- чтение из строки (**LODS**) – записывает в аккумулятор (AL или AX) содержимое ячейки памяти, адрес которой задается регистрами DS:SI, не влияет на флаги.

Команды строковой обработки чаще всего используются с однобайтными **префиксами** (префиксами повторения), которые обеспечивают многократное автоматическое повторение выполнения следующих команд:

- повторять, пока равно (**REPE**);
- повторять, пока нуль (**REPZ**);
- повторять (**REP**);

- повторять, пока не равно (**REPNE**);
- повторять, пока не ноль (**REPZ**).

Префиксы повторения ставятся перед строковыми командами обязательно в той же строке. Префикс использует регистр **CX** как счетчик циклов. На каждом этапе цикла выполняются следующие действия:

- 1) если $CX=0$, то выход из цикла и переход к следующей команде;
- 2) выполнение заданной строковой операции;
- 3) уменьшение CX на единицу, флаги при этом не изменяются;
- 4) выход из цикла, если:

- условие сравнения не выполняется для **SCAS** или **CMPS**;
- префикс **REPE** и $ZF=0$ (последнее сравнение не совпало);
- префикс **REPNE** и $ZF=1$ (последнее сравнение совпало);

5) изменение значения индексных регистров в соответствии со значением флага направления **DF** и переход на начало цикла.

Префикс **repe/repz** (repeat until equal – повторять, пока равно), применяемый с командами **cmps**, **scas**, завершает цикл по первому несовпадению сравниваемых элементов. Префикс **repne/repnz**, соответственно, завершает цикл при первом совпадении. Т. к. после выхода из цикла необходимо знать: завершился ли цикл по счетчику ($CX=0$) или досрочно (по признаку ZF), для этого предусмотрена команда **jcxz** (jump if CX zero), по которой выполняется анализ содержимого счетчика CX на ноль.