

В следующую пятницу вместо ассемблера будет дискра. Про ассемблер спросить у умных людей.

Существует 6 сегментных регистров, которые содержат в себе старшие 4 цифры адреса начала сегмента.

Адрес начала сегмента кода — cs.

Адрес начала сегмента стека система автоматически загружает в начало ss, регистр sp указывает **на вершину стека** и при добавлении элемента в стек содержимое регистра sp уменьшается. Это значит, что стек растёт вниз головой: значение адреса уменьшается от максимального. Чтобы в стеке хранить и фактические параметры, и локальные, после загрузки фактических параметров содержимое регистра sp сохраняется в регистре bp (base pointer) и тогда к фактическим параметрам можно обращаться с помощью выражения bp+k, а к локальным параметрам bp-n, где k и n вычисляет сам программист, зная количество и размер параметров.

Ещё один регистр **ip (eip)** называется **счётчиком** или **указателем команд**, в нём хранится смещение следующей исполняемой команды.

Регистр флагов определяет состояние программ и процессора в каждый текущий момент времени. Мы будем изучать ривервиальный режим работы, поэтому нас интересуют не все флажки.

1, 3, 5 и с 15 по 31 для 32-разрядных ассемблеров не используются. Следующие флажки используются и в реальном, и в защищённом режиме:

- CF — флажок переноса устанавливается в единицу, если в результате выполнения операций (например, сложения) произошёл перенос из старшего разряда, а при вычитании — заём. $0FFH + 1 = F00$, а CF = 1.
- BF — флажок чётности устанавливается в единицу, если в младшем байте результата окажется чётное число единиц. Используется при проверке правильности работы ОЗУ.
- AF — флажок полупереноса устанавливается в единицу, если при сложении произошёл перенос из четвёртого разряда в третий, а при вычитании требовался заём.
- ZF — флажок нуля (zero flag) устанавливается в единицу, если все разряды результата окажутся равными нулю.
- SF — флажок знака (sign flag) всегда равен содержимому знакового разряда ($0 \Leftrightarrow +$, $1 \Leftrightarrow -$).
- TF — флажок трассировки, установленный программистом в единицу, переводит процессор в режим пошаговой отладки программы.
- IF — флажок прерывания (interrupt flag), установленный программистом в ноль, заставляет процессор перестать обрабатывать прерывания от внешних устройств. Такое делают только для выполнения критических участков программ, это происходит весьма редко.
- DF — флажок направления определяет направление обработки строковых данных. Сброшенный программистом в 0, определяет обработку строк от младших адресов к старшим (слева направо). Установленный в 1, определяет обработку строк от старших адресов к младшим (справа налево). При этом автоматически изменяется содержимое регистра указателей si и di. Содержимое этих регистров или увеличивается, если df равен нулю, или уменьшается, если df равен единице, на размер операнда.
- OF — флажок переполнения (overflow flag) устанавливается в единицу, если результат превышает максимально допустимый для данной разрядной сетки

Следующие флаги используются в защищённом режиме:

- IOPL — флажок привелегий ввода-вывода
- NT — флажок вложенной задачи

- NF — флажок маскирования прерываний
- VM — флажок виртуальных машин
- VC — флажок выравнивания операнда
- флажок вложенных задач
- RF — флажок маскирования прерываний

Флаги за редким исключением устанавливаются автоматически.

Оперативная память

32-разрядный процессор может работать с оперативной памятью размером до 4ГБ с адресами от нуля до $2^{32}-1$, что в шестнадцатеричной системе будет 00000000-FFFFFFFF. Байты памяти могут определяться в поля переменной и фиксированной длины. Адресом начала поля является адрес младшего входящего в поле байта, длина поля — количество входящих в него байтов. Поля фиксированной длины имеют собственные имена, слово состоит из двух байтов, двойное слово — из четырёх байтов. Адресом поля переменной длины может быть любой адрес.

Реальный физический адрес байта состоит из двух частей: адрес начала сегмента и исполняемый адрес (смещение). Смещение формируется в команде и зависит от способа адресации операнда. В защищённом режиме программа может определить до 16 383 сегментов размером до 4ГБ и таким образом использовать до 64ТБ виртуальной памяти. В реальном режиме, как мы уже сказали, адрес сегмента кратен 16 и 4 старшие 16-ричные цифры содержатся в сегментном регистре. А чтобы получить 20-разрядный физический адрес байта, нужно сместить содержимое сегментного регистра на 4 разряда влево и прибавить 16-разрядное смещение.

Про кеш-память можно почитать по адресу: <https://market.marvel.ru/blog/komplektuyushchie-i-optsii/kesh-pamyat-kompjutera>

Форматы данных

Процессор вместе с сопроцессором могут обрабатывать большой набор данных. Числа целые без знака, целые со знаком, действительные с плавающей точкой, двоично-десятичные числа, символы, строки и указатели.

Целое число

Целое число без знака может занимать байт, слово или двойное слово и изменяться в диапазоне от нуля до 255, от нуля до 65535. Двойное слово — до 4294967295 соответственно.

Целое число со знаком также может занимать байт, слово или двойное слово и представляется в дополнительном коде. 7(15,31)

Обратный код числа m равен $10^n - m$, где n — разрядность числа.

Вычитание в машине дополнительный код вычитаемого прибавляется к уменьшаемому.

ДЗ Дома сделать $65 - 42 = 23$

Число с плавающей точкой

Числа с плавающей точкой могут занимать 32 разряда, 64 или 80 разрядов.

32 разряда — короткое вещественное, 64 — длинное вещественное, 80 — рабочее вещественное.

	знак	машинный порядок	мантисса
float	1	8	23
double	1	11	52
long double	1	15	64

Старшая единица, нормализованная мантиссой, в разрядную сетку не записывается для экономии памяти.

1. Нормализуем число: $0.BF4 * 10^3$
2. Получаем машинный порядок: $3_{16} + 7F_{16} = 82_{16}$
3. Запишем число в двоичной системе счисления:
0.100000100111110100000000000000

Процессором могут обрабатываться 8-разрядные в упакованном и неупакованном формате, сопроцессором — 80-разрядные в упакованном формате.

Упакованный формат две цифры в байте

Неупакованный формат одна цифра в цифровой части файла

Символы, строки, указатели

Символы представляются в коде ASCII, каждому символу отводится один байт.

Строки последовательности байтов, слов или двойных слов.

Указатели адреса байтов. Существует длинный указатель (16+32 разрядов, где 16 — селектор (адрес сегмента) и 32 — адрес смещения в сегменте) и короткий (32 разряда смещения).

Форматы команд

Команды как машинные команды — цифровой двоичный код, состоящий из двух последовательностей, определяющий операционную часть (код операции: что нужно сделать) и адресную часть (где взять операнды и куда записать результат). Процессор, который мы рассматриваем, может работать с безадресными командами, с одноадресными, двухадресными и трёхадресными командами.

Данные, участвующие в операции, могут находиться непосредственно в команде, могут содержаться в регистрах или оперативной памяти. В зависимости от кода операции количество операндов и их размещение, команда операции в памяти процессора может занимать от одного до 15 байтов. Наиболее частоиспользуемыми являются двухадресные команды, их формат записывают таким образом:

- R-R (регистр-регистр)
- R-M (регистр-память)
- M-M (память-память)
- M-R (регистр-регистр)
- R-D (регистр-данные)
- M-D (память данные)

Исполняемый адрес может состоять из трёх частей: база, индекс и смещение.

Базовый регистр bs, регистр si...

Существуют различные способы адресации операндов (мы будем использовать 7):

- Регистровая.
- Непосредственная
- Прямая
- Косвенно-регистровая
- По базе с индексированием
- По