Порядковый номер байта в ОЗУ – его физический адрес (далее ФА) (первый байт в адресе 0)

Логически процессор и ОЗУ соединены 3мя шинами:

1. Адреса (указание ФА, для кот. Нужно выполнить операцию)
2. Данных (для передачи данных: для чтения – передача прочитанных данных из памяти, а при записи – записанные данные передаёт процессор). Разрядность шины данных определяет размер памяти, кот. может быть передана за 1 обращение к ОЗУ = разрядность процессора
3. Управления (выбор операции: чтение или запись)

Команды процессора – инструкции:

1. Операция (действие)
2. Исходные данные
3. Результат (место записи)

Для каждого вида процессора свои правила записи инструкции в виде послед. байт – машинном коде.

Этапы выполнения инструкции:

1. Fetch – чтение очередной инструкции из ОЗУ
2. Decode – определение типа инструкции; определение вычислительных устройств, которые нужно задействовать; какие исходные данные и откуда необходимо загрузить для выполнения
3. Read – загрузка исходных данных
4. Execute – выполнение инструкции (преобразование исходных данных в результаты)
5. Write-back – запись результат в отведённое место

Данные, с которыми работают инструкции – операнды. 3 вида операндов:

1. Операнды-регистры (registers - reg). Физически регистр находится внутри процессора – доступ к ним происходит намного быстрее, чем к операндам в памяти.
   1. Регистры общего назначения (GPR)
   2. Сегментные регистры
   3. Управляющие регистры
   4. Отладочные регистры
   5. Регистр программного счётчика
   6. Регистр флагов
   7. И др.
2. Операнды-константы (immediate values - imm). Записываются прямо в коде программы:
3. Операнды в памяти – адреса (memory operands - mem). В квадратных скобках записываются выражения, задающие адрес операнда в памяти. Ограничение: не более 1 в каждой инструкции.

Ограничение на хранение определённых типов данных в регистрах:

1. Некоторые регистры недоступны напрямую (не могут использоваться как операнды инструкций)
2. Значения некоторых регистров используются неявно
3. Некоторые регистры управляют работой процессора

Группы регистров, которые используются при программировании в Real-Mode:

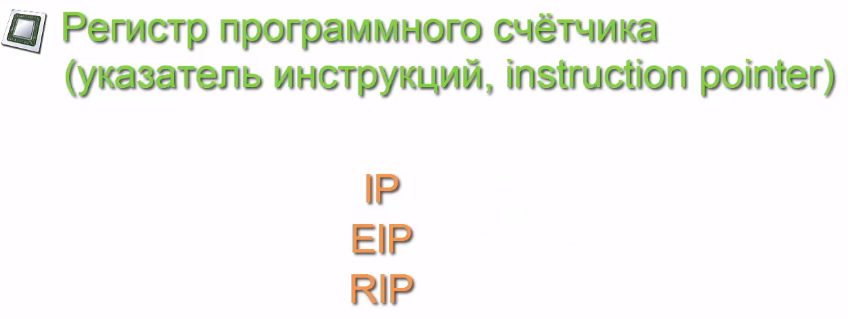


AH – AX-high, AL – AX-low

При программировании в Real-Mode рекомендуется использовать только 8-bit и 16-bit регистры общего назначения



Размер – 16-bit (2 Байта). Используются для хранения номер сегментов в памяти.



На этапе Fetch процессор считывает инструкцию по адресу, хранящемуся в этом регистре. Затем увеличивает его значение на размер прочитанной инструкции. Важно: для использования напрямую регистр недоступен.

IP – 16-bit, EIP – 32-bit. RIP – 64-bit



Хранит состояние какого-либо объекта. Важно: не доступны напрямую

EFLAGS – 32-bit, его младшие 2 байта – FLAGS.

Простейшие инструкции:

mov dest, src

Копирование данных (destination, source) Dest = src; Размеры должны совпадать

add dest, src

Прибавить dest = dest + src; Размеры должны совпадать

sub dest, src

Отнять dest = dest – src; Размеры должны совпадать

And dest, src

Побитовое И dest = dest AND src

Or dest, src

Побитовое ИЛИ dest = dest OR src

Not dest

Побитовое НЕ dest = NOT dest

Xor dest, stc

Побитовое ИСКЛЮЧАЮЩЕЕ ИЛИ dest = dest XOR src

Запуск программы:

1. Загрузка исполняемого файла в ОЗУ
2. Определение формата исполняемого файла
3. Определение адреса точки входа (entry point)
4. Передача управления на точку входа (помещение адреса в IP или EIP)

Задача компилятора языка ассмеблера – формирование исполняемого файла, в котором будут содержаться интсрукции процессора.

Слово “ячейка” имеет много значений, нужно уточнять.

“Пустых” “ячеек” не бывают. Бывают “ячейки”, куда мы ничего не записали (ещё не использованные “ячейки”)

“Код” и “данные” очень условны. В одной “ячейке” могут быть данные, представляющие буквально ЧТО УГОДНО.

Код – это то, адрес чего находится в E(IP)

Записи байтов (директивы):

DB (data bytes) – 1 Byte

DW (data words) – 2 Byte

DD (data doubleWords) – 4 Byte

При записи байтов всё, что введено в ‘А’ или “А” эквивалентно переводу этих записей в числа (65)

“String” = ‘S’, ‘t’, ‘r’, ‘I’, ‘n’, ‘g’

Директивы попарно эквивалентны:

Db 8 dup(5)

Db 5,5,5,5,5,5,5,5

Db 3 dup(2, ‘F’)

Db 2, ‘F’, 2, ‘F’, 2, ‘F’

Если нужно занять память не важно чем, можно инициализировать поле знаком ?

**.com** программа всегда загружается по адресу 0100h

Занесение функции на выполнение:

1. Поместить в ah номер функции
2. Подготовить остальные параметры
3. Вызвать фукнцию через **int 21h**

**Ret** возвращает управление системе