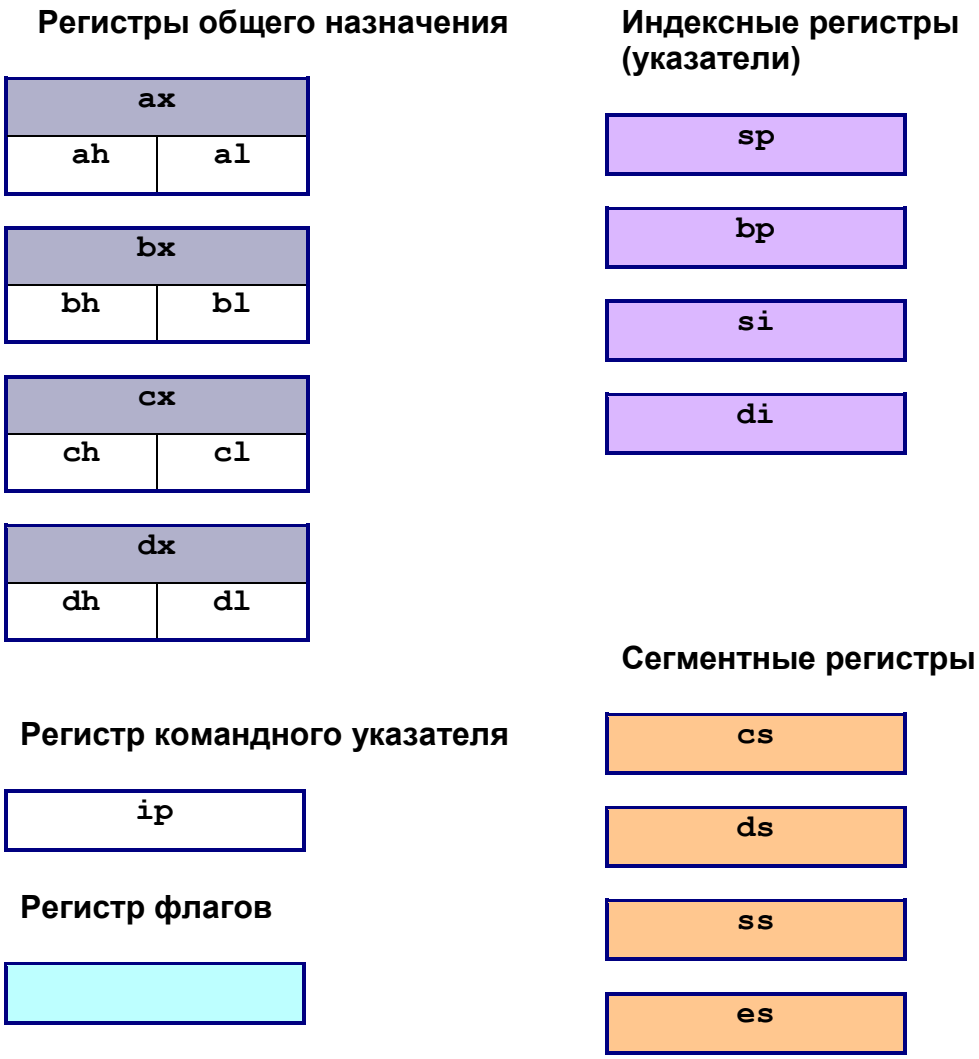
Лекция №2
Тема " Регистры процессора 8086"

При изучении ассемблера мы будем рассматривать процессоры семейства 80X86, к которым относятся 8086, 80286, 80386, 80486, Pentium. Эти процессоры объединены, т.к. соблюдается *приемственность*: программа, написанная для младших моделей, может быть без каких-либо изменений выполнена на старших моделях.
Это обеспечивается тем, что в основе процессоров лежит система команд процессора 8086, а старшие лишь добавляются новыми командами.

В программах на языке ассемблера обращаются к регистрам, используя их символические имена - ax, cl, ds...



14 шестнадцатиразрядных регистров

Регистры общего назначения

ax:	**ah**	**al**	Сумматор	Accumulator
bx:	**bh**	**bl**	Базовый	Base
cx:	**ch**	**cl**	Счетчик	Count
dx:	**dh**	**dl**	Регистр данных	Data

Примечание
В процессоре 8080 были *байтовые* регистры A, B, C, D, но затем их расширили до размера *слова*. Отсюда появились новые названия регистров : AX, BX, CX, DX (eXtended).

- В регистре-*сумматоре* обычно содержат результаты сложения, вычитания и т.п.
- *Базовый* регистр часто указывает на начальный адрес структуры в памяти.
- *Счетчик* определяет количество повторов некоторой операции.
- Регистр *данных* чаще всего содержит данные, передаваемые для обработки в подпрограммы.

У регистров **ax**, **bx**, **cx**, **dx** можно осуществлять доступ к старшим и младшим половинам. Обозначают эти половины буквами **H** (High - выше, старший) и **L** (Low - ниже, младший) и первой буквой из названия регистра

Сегментные регистры

cs	Регистр сегмента кода	Code segment
ds	Регистр сегмента данных	Data segment
ss	Регистр сегмента стека	Stack segment
es	Доп. сегментный регистр	Extra segment

Регистр сегмента кода **cs** указывает в памяти на начало машинного кода программы.

Регистр сегмента данных **ds** содержит начальный адрес переменных программы.

Регистр сегмента стека **ss** определяет начало стекового пространства.

Дополнительный сегментный регистр **es** является вспомогательным.

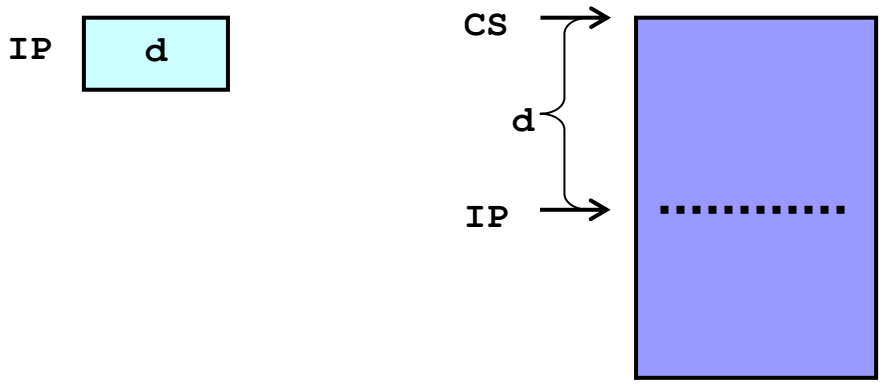
Сегментные регистры служат для обращения к оперативной памяти.
Сегментные регистры являются строго *специализированными* регистрами, над ними нельзя выполнять математические вычисления и хранить в них результаты других операций.

Существуют три основных типа сегментов и один дополнительный.

CS		сегмент кода – содержит машинные команды
SS		сегмент стека – содержит адреса возврата в точку вызова подпрограмм
DS		сегмент данных – содержит данные, то есть константы и рабочие области, необходимые программе
ES		дополнительный сегмент

Регистр командного указателя (регистр счетчика команд)

Специальный *регистр командного указателя* **ip** содержит смещение очередной команды в сегменте кода, адресуемом с помощью регистра CS.



Процессор использует регистр IP совместно с регистром CS для формирования 20-битового физического адреса очередной выполняемой команды, при этом регистр CS задает сегмент выполняемой программы, а IP – смещение от начала сегмента. По мере того, как процессор загружает команду из памяти и выполняет ее, регистр IP увеличивается на число байт в команде. Для непосредственного изменения содержимого регистра IP служат команды перехода.

Индексные регистры (регистры смещения)

sp	Стековый указатель	Stack pointer
bp	Базовый указатель	Base pointer
si	Индекс источника	Source index
di	Индекс назначения	Destination index

Данные регистры содержат смещение в сегментах данных, стека и в дополнительном сегменте.

Регистры тесно связаны с *определенными операциями*.

Указатель стека **sp** и у*казатель базы* **bp** содержат смещение относительно начала сегмента стека, обеспечивая доступ к данным, помещенным в стек.

Индекс источника **si** и *индекс назначения* **di** используются для формирования сложных адресов, состоящих из смещения начала некоторого поля данных в сегменте данных и относительного смещения элемента данных внутри этого поля.

Сегменты, принцип сегментации

Для адресации памяти процессор использует 16-разрядные адресные регистры, что обеспечивает доступ к 65536 (FFFFh) байт или 64К основной памяти. Такой блок непосредственно адресуемой памяти называется *сегментом*.

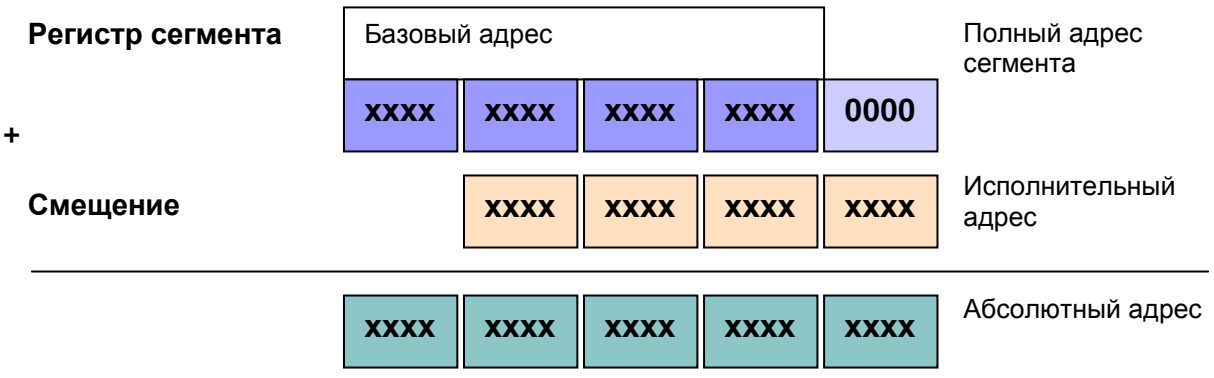
Любой адрес формируется из адреса сегмента (всегда кратен 16) и адреса ячейки внутри сегмента (этот адрес называется смещением).

Для адресации большего объема памяти в процессоре 8086 используется специальная процедура пересчета адресов, называемая вычислением абсолютного (эффективного) адреса.

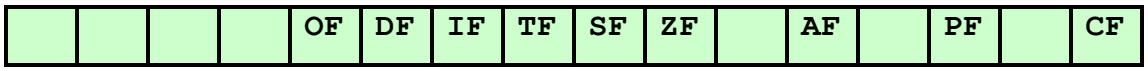
Когда процессор выбирает очередную команду на исполнение, в качестве ее адреса используется содержимое, регистра IP. Этот адрес называется исполнительным. Поскольку регистр IP шестнадцатиразрядный, исполнительный адрес тоже содержит 16 двоичных разрядов. Однако адресная шина, соединяющая процессор и память имеет 20 линий связи.

Чтобы получить 20-битовый адрес, дополнительные 4 бита адресной информации извлекаются из сегментных регистров. Сами сегментные регистры имеют размер в 16 разрядов, а содержащиеся в этих регистрах (CS, DS, SS или ES) 16-битовые значения называются базовым адресом сегмента. Микропроцессор объединяет 16-битовый исполнительный адрес и 16-битовый базовый адрес следующим образом: он расширяет содержимое сегментного регистра (базовый адрес) 4 нулевыми битами (в младших разрядах), делая его 20-битовым (полный адрес сегмента) и прибавляет смещение (исполнительный адрес). При этом 20-битовый результат является физическим или абсолютным адресом ячейки памяти.

Принцип получения абсолютного адреса



Регистр флагов



Регистр флагов состоит из 16 бит, из них используются только 9.

Флаг - это бит, принимающий значение 1 («флаг установлен»), если выполнено некоторое условие, и значение 0 («флаг сброшен») в противном случае.

Флаги условий:

CF	переноса (carry flag)	признак устанавливается в 1, если имеет место переноса из старшего бита при сложении или заем в старший бит при вычитании
OF	переполнения (overflow flag)	равен 1, если операция привела к переносу (заему) в знаковый бит результата, но не привела к переносу (заему) из знакового бита.
ZF	нуля (zero flag)	равен 1, если все биты результата равны 0
SF	знака (sign flag)	равен 1, если старший бит результата равен 0, т.е. SF = 0 - число положительное SF = 1 - число отрицательное
PF	четности (parity flag)	равен 1, если результат имеет четное число единиц
AF	дополнительного переноса (auxiliary carry flag)	равен 1, если во время выполнения команд десятичного сложения и вычитания осуществлено выполнение переноса или заема между полубайтами

Флаги состояний:

DF	направления (direct flag)	устанавливает направление просмотра строк в строковых командах DF = 1, то регистры SI и/или DI уменьшаются на 1 при выполнении операций со строками, если DF = 0, то увеличиваются.
IF	прерывания (interrupt flag)	IF = 0 процессор перестает реагировать на запросы внешних (маскируемых) прерываний IF = 1 блокировка прерываний снимается
TF	трассировки (trap flag)	TF = 1 процессор работает в пошаговом режиме; используется для отладки программы