Архитектура 32-битных Intel-совместимых микропроцессоров

Основные характеристики процессора

Разрядность

количество (двоичных) разрядов в представлении обрабатываемых данных (например, 32)

Адресное пространство

набор допустимых адресов (номеров) ячеек памяти (например, 4 Gb)

Основные характеристики процессора

Система команд процессора

полный набор команд, которые может выполнять процессор

Регистры

набор внутренних ячеек памяти, предназначенных для хранения промежуточных результатов обработки данных или для управления работой процессора

Архитектура микропроцессоров

Большинство выпускаемых в настоящее время процессоров для *персональных* компьютеров соответствуют архитектуре, впервые реализованной в микропроцессоре *Intel 80386* (1985 год).

Обычно её называют *32-bit Intel Architecture* или просто *IA-32.*

Программная модель микропроцессора IA-32

Регистры

высокоскоростные ячейки памяти, расположенные внутри процессора. Доступ к регистрам осуществляется гораздо быстрее, чем к ячейкам оперативной памяти.

Виды регистров в модели IA-32

- 32-разрядные регистры общего назначения (8 шт.);
- 16-разрядные сегментные регистры (6 шт.);
- 32-разрядные регистры состояния и управления (2 шт.);
- 80-разрядные регистры сопроцессора (8 шт.);
- наборы регистров расширений (…);
- системные регистры (…).

Регистры общего назначения
EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP

Предназначены для хранения:

- операндов арифметических и логических операций, а также их результатов;
- адресов (указателей) ячеек памяти;
- компонентов адресов.

Кроме этого, исторически, каждый регистр имеет дополнительные особенности использования.

Программная модель микропроцессора IA-32

Регистр EAX – аккумулятор

- обеспечивает наиболее эффективное выполнение большинства арифметических и логических операций;
- обязательно используется в некоторых арифметических командах (умножение, деление);
- обязательно используется при выполнении операций ввода-вывода;
- неявно используется при последовательной обработке цепочек элементов (ввод, вывод, поиск, заполнение и т. п.)

Регистр EBX – базовый регистр

– обеспечивает наиболее эффективное вычисление адреса операнда при *расширенной (базовой) адресации*, например, при работе с массивами.

Регистр ECX – счетчик

– используется как счетчик итераций при организации повторений и циклов

Регистр EDX – регистр данных

- используется при организации ввода-вывода;
- обязательно используется в некоторых арифметических командах (умножение, деление).

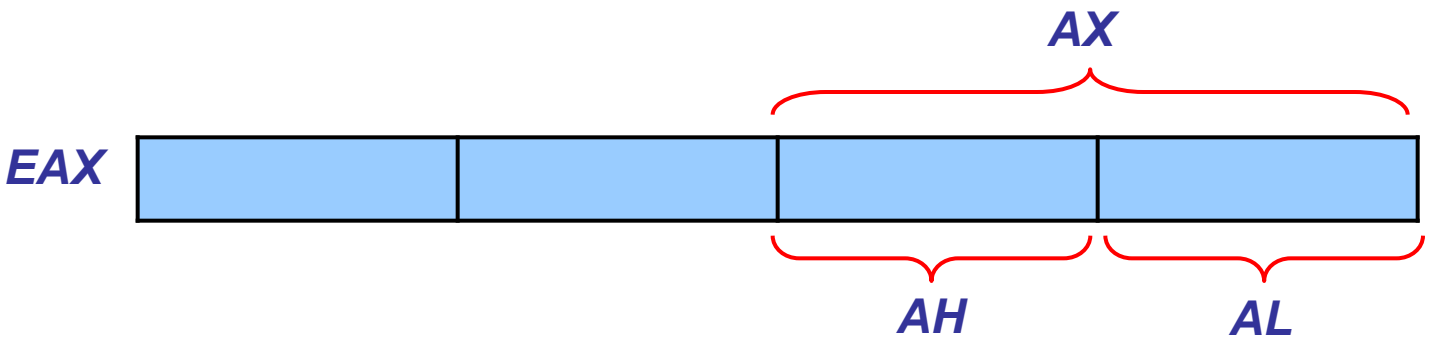
Состав регистров EAX, EBX, ECX, EDX

Части регистров EAX, EBX, ECX, EDX имеют собственные имена, например,

AX – младшее слово регистра EAX;

AL – младший байт слова AX;

AH – старший байт слова AX



Регистры ESI, EDI – индексные регистры

– используются при выполнении цепочечных (последовательных) операций: копирование, заполнение, сравнение и т.п.

ESI – индекс источника

EDI – индекс назначения

При выполнении цепочечных операции регистры ESI, EDI автоматически изменяются на ±1.

Регистры ESP, EBP – указатели стека

– используются при обращении к стеку

ESP – указатель вершины стека (!)

EBP – база окна локальных переменных

Состав регистров ESI, EDI, ESP, EBP

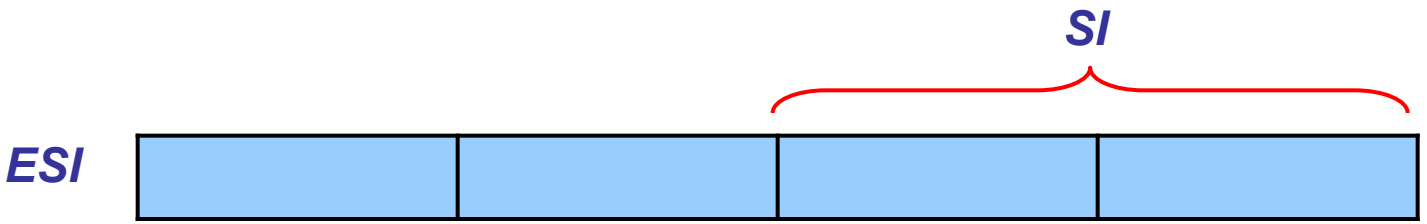
Части регистров ESI, EDI, ESP, EBP имеют собственные имена:

SI – младшее слово регистра ESI;

DI – младшее слово регистра EDI;

SP – младшее слово регистра ESP;

BP – младшее слово регистра EBP;



Сегментные регистры CS, DS, ES, FS, GS, SS

Предназначены для организации хранения данных в различных *сегментах* памяти:

CS – содержит указание на сегмент *кода*;

DS – содержит указание на сегмент *данных*;

SS – содержит указание на сегмент *стека*;

ES, FS, GS – содержат указания на *дополнительные сегменты данных*

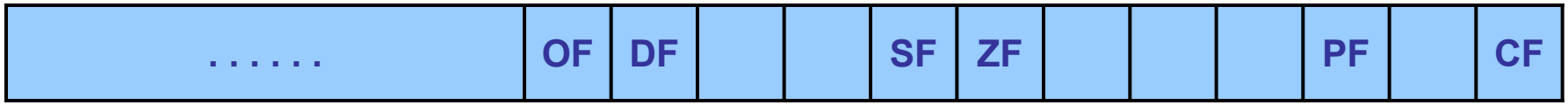
Регистр EIP – указатель инструкций

- содержит адрес (указатель) очередной *инструкции*, подлежащей выполнению;
- после выполнения одной инструкции автоматически заносится адрес следующей инструкции;
- напрямую программе не доступен; для изменения последовательности выполнения инструкций используются специальные *команды перехода*.

Регистр флагов EFLAGS

– содержит информацию о текущем состоянии процессора и результатах выполнения команд; напрямую программе не доступен;

– каждый бит (*флаг*) регистра EFLAGS имеет свой смысл и обозначение



Программная модель микропроцессора IA-32

Флаг переполнения – OF (Overflow Flag)

обозначает выход результата за пределы допустимого диапазона при арифметических операциях со ***знаковыми*** числами:

1 – было арифметическое переполнение;

0 – арифметического переполнения не было.

Пример. Действия с *signed char:*

```
  01111111    127
+ 00000001  +   1
  10000000   -128
```

Флаг направления – DF (Direction Flag)

задает направление изменения индексных регистров ESI, EDI при выполнении цепочечных команд:

0 – приращение (+1, от начала к концу);
1 – убавление (–1, от конца к началу).

Флаг знака – SF (Sign Flag)

показывает знак (старший бит) результата в последней выполненной процессором арифметической операции:

0 – результат неотрицательный;
1 – результат отрицательный.

Флаг нуля – ZF (Zero Flag)

показывает, был ли нулевым результат последней выполненной процессором арифметической операции:

0 – результат ненулевой;
1 – результат нулевой.

Флаг четности – PF (Parity Flag)

показывает, сколько единичных бит было в младшем байте результата последней выполненной процессором арифметической операции:

0 – нечетное количество;
1 – четное количество.

Флаг переноса – CF (Carry Flag)

показывает, был ли перенос за пределы разрядной сетки при арифметических операциях:

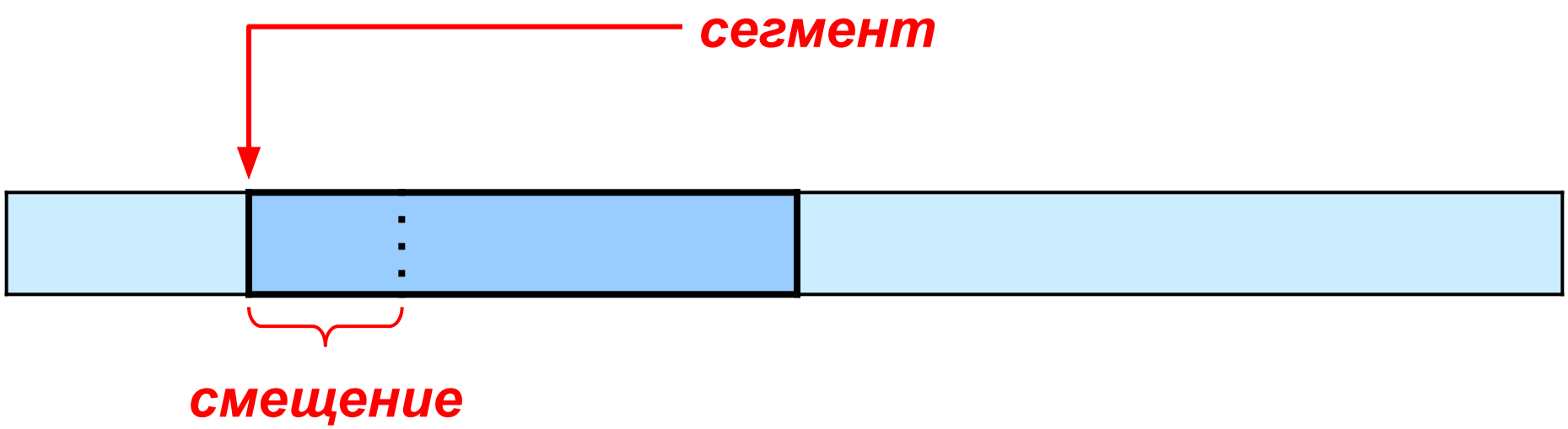
1 – был арифметический перенос;

0 – арифметического переноса не было.

Пример. Действия с *unsigned char:*

```
    11111111     255
+   00000001   +   1
   100000000       0
```

Программная модель микропроцессора IA-32



Программная модель микропроцессора IA-32

Основные режимы работы процессора:

- реальный режим;
- защищенный режим;
- режим виртуального процессора 8086.

Перевод процессора из одного режима в другой осуществляется специальными командами.

Реальный режим (режим реальных адресов)

- соответствует режиму работы процессора Intel 8086 (1978 год);
- при включении компьютера процессор изначально находится в реальном режиме;
- в реальном режиме программа может напрямую обращаться к физической памяти;
- в реальном режиме программе доступно не более 1 мегабайта ОЗУ.

Вычисление адреса в реальном режиме

адрес = сегмент*16 + смещение

Поскольку величины **сегмент** и **смещение** могут принимать значения от 0 до 65535, то максимальный адрес ячейки памяти может быть 1114095.

Подобная схема расчета позволяла 16-разрядному процессору Intel 8086 использовать 20-разрядные адреса.

Программная модель микропроцессора IA-32

Величина **сегмент** берется из соответствующего сегментного регистра (код – CS, стек – SS, данные – DS, ES, FS, GS).

Величина **смещение** указывается непосредственно в команде или берется из какого-нибудь регистра.

Например, при обращении к вершине стека, адрес ячейки памяти будет рассчитываться по формуле:

адрес = SS*16 + ESP

Недостатки механизма сегментации реального режима:

- ограниченный максимальный размер сегмента (64 Кб);
- сегменты могут перекрываться с другими сегментами;
- программа может использовать произвольные адреса начала сегментов и, следовательно, обращаться по любым адресам памяти.

Защищенный режим

- реализован в процессорах, начиная с i80286 (1982 год);
- позволяет использовать большее количество физической памяти – 16 Мб (i80286), 4 Гб (i80386 – Pentium), 64 Гб (Pentium Pro – …) и т.д.;
- поддерживает многозадачность (одновременное выполнение нескольких программ);
- позволяет защитить исполняемые процессором программы от взаимного влияния;

и т.д.

Программная модель микропроцессора IA-32

В защищенном режиме прямой доступ к физической памяти возможен только для программ, имеющих особые привилегии (например, операционная система).

Для остальных программ используется более сложный *аппаратный* механизм, осуществляющий вычисление адреса на основании специальных *таблиц дескрипторов сегментов*, содержащих информацию о том, какие сегменты находятся в распоряжении программы.

Вычисление адреса в защищенном режиме

- сегментный регистр содержит *селектор* – указатель на элемент таблицы дескрипторов;
- из данных этой таблицы извлекается адрес начала сегмента;
- к адресу начала сегмента прибавляется заданное в смещение.

Преимущества сегментации памяти защищенного режима

- большие размеры сегментов (до 4 Гб);
- предотвращается доступ к памяти за границами отведенного сегмента;
- контроль прав доступа к памяти осуществляется на аппаратном уровне;

Режим виртуального процессора 8086 (V86)

В этом режиме процессор эмулирует работу процессора Intel 8086 (механизм адресации, 1 Мб памяти и т.п.), но при этом сохраняет все средства контроля защищенного режима.

Предназначен для организации многозадачной работы программ, разработанных для реального режима процессора Intel 8086.