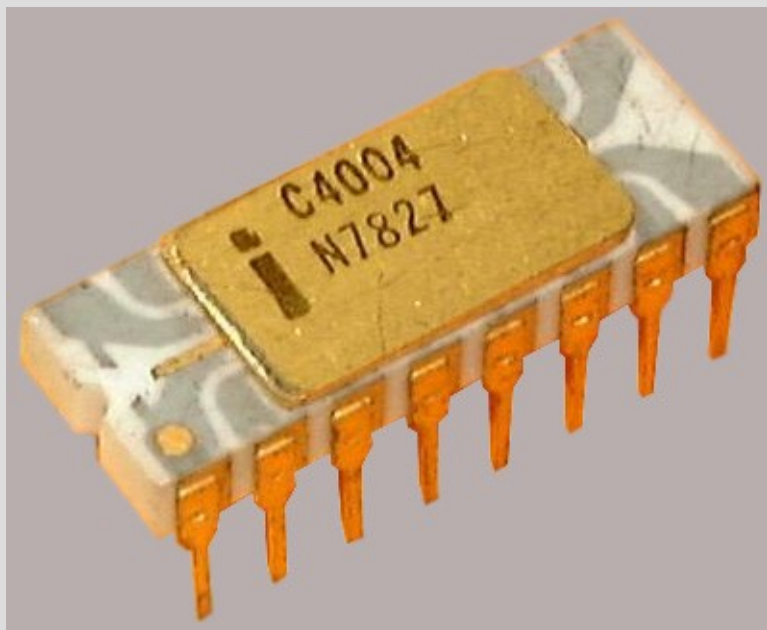


# Введение в архитектуру систем на базе семейства процессоров Intel 80x86

# Линейка процессоров Intel

Подробная информация по линейке процессоров Intel (с 1971 до 2011 г.):

[http://all-ht.ru/inf/pc/proc\\_intel\\_hrono\\_all\\_1.html](http://all-ht.ru/inf/pc/proc_intel_hrono_all_1.html)



# Линейка процессоров Intel

- 1971 Intel 4004 — 4-х разрядный, 4K памяти программ, 640Б памяти данных;
- 1972 Intel 8008 — 8-ми разрядный, 16K памяти;
- 1974 Intel 8080 — 8-ми разрядный, 64K памяти;
- 1976 Intel 8085 — 8-ми разрядный, 64K памяти;
- 1978 Intel 8086 — 16-ти разрядный, 16-ти разрядная шина данных, 20-разрядная шина адреса (1M RAM);  
Основоположник линейки 80x86;
- 1979 Intel 8088 - 16-ти разрядный, 8-ми разрядная мультиплексированная шина данных, 20-разрядная шина адреса (1M RAM);
- 1980 Intel 8087 — математический сопроцессор для Intel 8086/8088

# Линейка процессоров Intel

- 1982 Intel 80286 — 16-ти разрядный, 24-х разрядная шина адреса, адресуемая память - 16 Гб. Два режима работы — реальный и защищённый. В защищённом режиме аппаратная поддержка мультизадачного режима. Сопроцессор — 80287.
- 1985 Intel 80386 DX -32-х разрядный. 32-х разрядная шина данных и 32-х разрядная шина адреса. Адресное пространство 4Гб. Три режима работы — реальный, защищённый и режим виртуального 8086 в рамках защищённого режима. Сегментированная и плоская(Flat) модели памяти. Улучшение мультизадачности, защиты и виртуальной памяти. Сопроцессор — 80387DX — 1987 г.

# Линейка процессоров Intel

- 1988 Intel 80386 SX - 32-х разрядный. 16-х разрядная мультиплексированная шина данных и 24-х разрядная шина адреса. Адресное пространство 16 Мб.  
Сопроцессор — 80387sx;
- 1989 Intel 80486 DX - 32-х разрядный. 32-х разрядная шина данных и 32-х разрядная шина адреса. Адресное пространство 4Гб. Дополнительно к 80386 — гибридное CISC-RISC ядро, интегрированная L1 кэш-память 8К, интегрированный математический сопроцессор, 5-стадийный конвейер команд.
- 1991 Intel 80486 SX — версия 80486 без встроенного сопроцессора. Сопроцессор — 80487.

# Линейка процессоров Intel

- 1992 Intel 80486 DX2, Intel 80486SX2, Intel 80486 OverDrive — дополнительно появилось умножение частоты.
- 1993 Intel Pentium (ядро P5);
- 1994 Intel Pentium (ядро P54C) — 64-х разрядная шина данных, 32-х разрядная шина памяти, отдельный L1-кэш для инструкций и данных(8к+8к), суперскалярная архитектура, позволяющая выполнять 2-е инструкции одновременно.

# Линейка процессоров Intel

- 1995 Intel Pentium Pro (ядро P6) — кэш L2 на кристалле от 256К до 1024М, работающий на частоте ядра процессора. Оптимизация под 32-х разрядные вычисления, расширение параллелизма (доп. декодир.блоки, внеочередное (out-of-order) исполнение операций), возможность совместной работы до 4-х процессоров в SMP-архитектуре;
- 1997 Intel Pentium MMX (ядро P55C) — дополнительные 57 mmx (MultiMedia eXtensions) инструкций, позволяющих ускорить обработку мультимедийных данных (видео и звука) за счет выполнения одной операции над группой данных (целочисленное SIMD-расширение)

# Линейка процессоров Intel

- 1997 Intel Pentium II — развитие линейки PentiumPro (L2-кэш память в 2 раза медленнее ядра и дешевле, набор инструкций MMX) (Фактически Pentium Pro+MMX);
- 1998 Intel Celeron — упрощенная версия P II, с уменьшенным (в первых версиях отсутствующим) кэшем L2 и меньшей FSB;
- 1999 Intel Pentium III — развитие линейки P II, увеличение количества исполнительных блоков, появление SSE, позволяющее выполнять SIMD-операции над данными с плавающей точкой одинарной точности (32bit)



# Линейка процессоров Intel

- 2000 г - Intel Pentium 4 — переработанная микроархитектура, наборы команд MMX, SSE, SSE2, с 2002 г. (Intel Pentium 4 (ядро Northwood)) — поддержка технологии HyperThreading (HT);
- 2004 Intel Pentium 4 (ядро Prescott) — набор инструкций MMX, SSE, SSE2, SSE3, поддержка набора инструкций EM64T (AMD64), XD-bit (Execute Disable), HT;
- 2006 Intel Pentium D (ядро Presler) — двухядерные процессоры, часть линейки с аппаратной поддержкой виртуализации VT-x;
- 2006 Intel Core 2 Duo (ядро Conroe) — более производительная замена линейки Pentium D (~на 40% ) с более низким энергопотреблением

# Линейка процессоров Intel

- 2008 Intel Core 2 Duo (ядро Wolfdale) — 64-х битный, двухядерный, поддержка наборов команд MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, EMT64, EIST, XD-bit, VT-x;
- 2008 Intel Atom — процессоры для MID-устройств, наборы команд и технологий: MMX, SSE, SSE2, SSE3, SSSE3, Intel 64, XD bit, Hyper-Threading;
- 2008 Intel Core i7 — 64-х битный, 4-х ядерный высокопроизводительный процессор семейства Core, новое вычислительное ядро, встроенный контроллер памяти, новая архитектура кэш-памяти (L1, L2, L3), набор команд и технологий: MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, Enhanced Intel SpeedStep Technology, Intel 64, XD bit, Intel VT-x, HT, Turbo Boost, Smart Cache.

# Линейка процессоров Intel

- 2011 Intel Core i3,i5,i7 Dual/Quad-Core (ядро Sandy Bridge) — 64-битный, интегрированное графическое ядро, интегрированный контроллер памяти, поддерживаемые наборы инструкций и технологии: MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, AVX, Enhanced Intel SpeedStep Technology (EIST), Intel 64, XD bit (an NX bit implementation), TXT, Intel VT-x, Intel VT-d, Hyper-threading, Turbo Boost, AES-NI, Smart Cache
- 2013 Intel Core i3,i5,i7 4<sup>th</sup> Generation (ядро Haswell) — 64-битный, улучшенное графическое ядро (Iris Graphics), усовершенствованная архитектура Cache, дополнительные инструкции AVX 2, BMI 1, BMI 2, улучшенные механизмы энергосбережения и сниженное энергопотребление.

# Процессоры Intel для мобильных и встраиваемых систем

Intel Atom — 2008 — 2013 — 64-разрядные процессоры со сниженным энергопотреблением (от 0,65 W (серия Z) до 13 W (серия D)), количеством ядер от 1 до 4 (серия E), область применения — netbooks, nettops, встраиваемые решения;

Intel Bay Trail — 2013-2014 — 64-разрядные мультиядерные SoC со сниженным энергопотреблением, расширенным набором команд (SSE4.1, SSE4.2, AES-NI), поддержкой технологии аппаратной виртуализации Intel VT-x с EPT, модифицированной технологией автотурбо Turbo Burst 2.0, интегрированной графикой Intel HD Graphics

# Программная модель процессора в 16-разрядном режиме

AH	AL											
BH	BL											
CH	CL											
DH	DL											
	O	D	I	T	S	Z		A		P		C

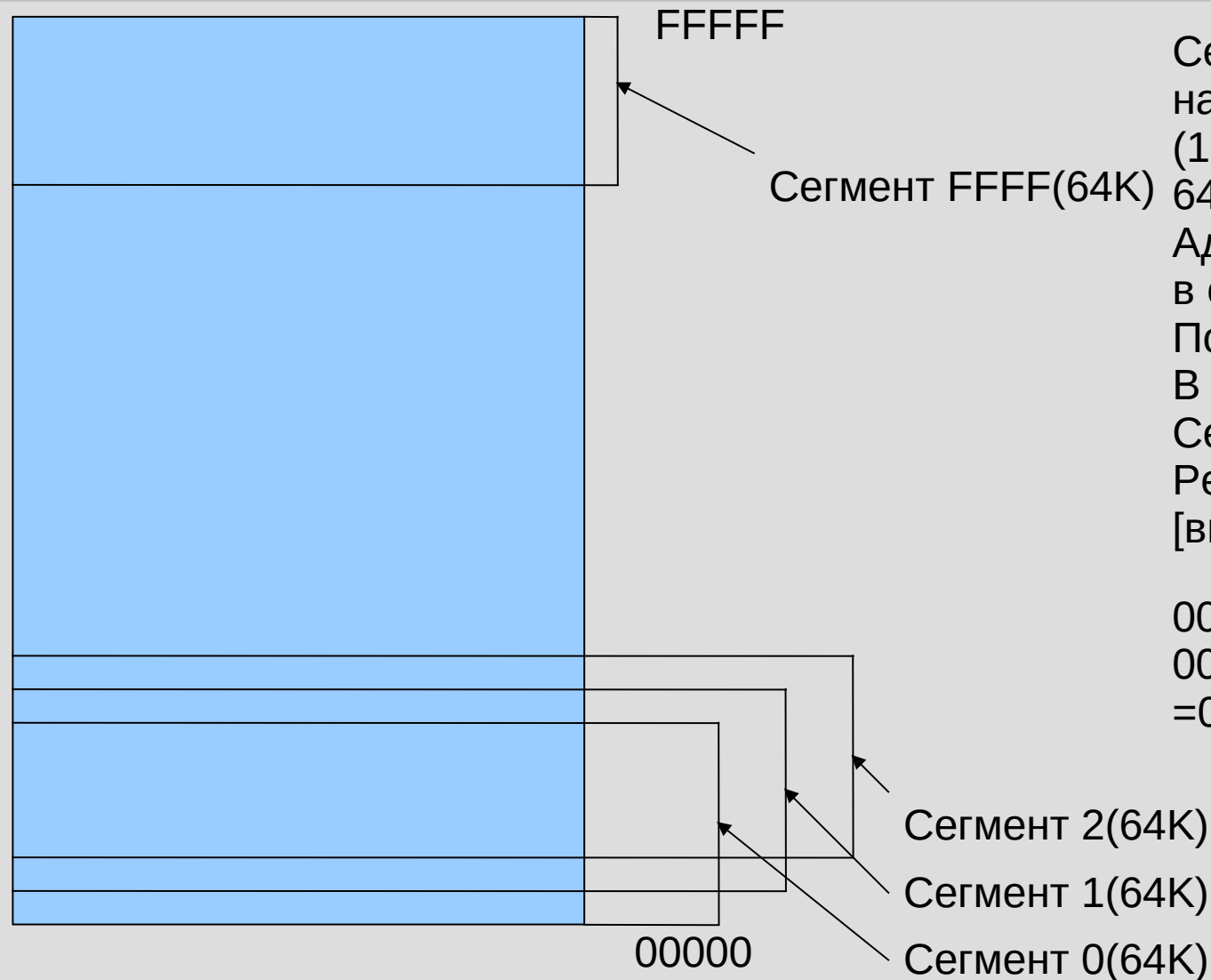
AX  
 BX  
 CX  
 DX  
 SI  
 DI  
 BP  
 SP  
 F  
 IP

Флаги:  
 O - переполнения;  
 D - направления;  
 I - прерываний;  
 T - трассировки;  
 S - знака;  
 Z - нуля;  
 A - вспом.переноса  
 P - чётности;  
 C - переноса

CS
DS
SS
ES

Сегментный регистр кода  
 Сегментный регистр  
 данных  
 Сегментный регистр стека  
 Сегментный регистр расш.

# Сегментная модель памяти



Сегмент выравнивается на границу параграфа (16 байт), размер сегмента 64 Кбайт.

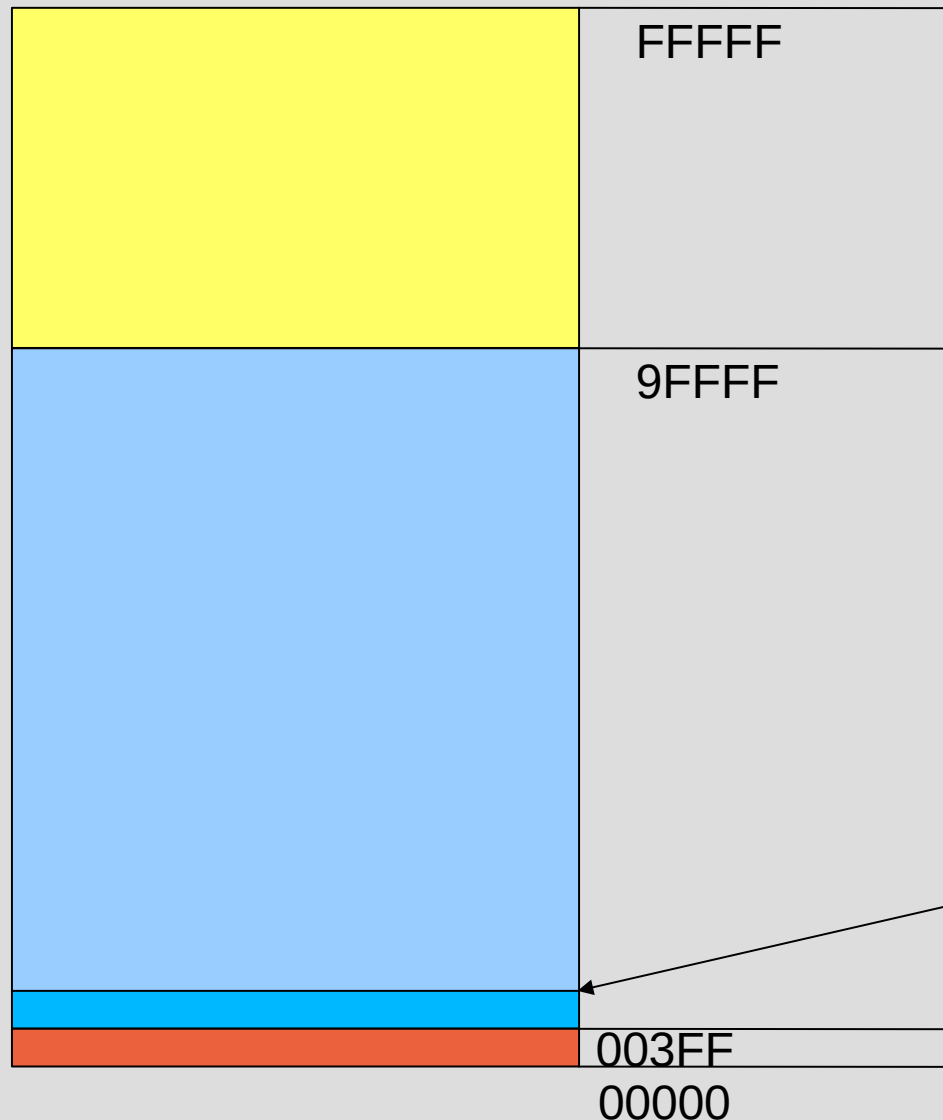
Адрес сегмента хранится в сегментном регистре. Полный адрес представлен в виде:

Сегмент:смещение  
Реальн. адрес = [сегм] << 4 + [внутрисегм.смещение]

0040:0017

$0040 \ll 4 + 0017 = 00400 + 0017 = 00417$

# Модель памяти реального режима



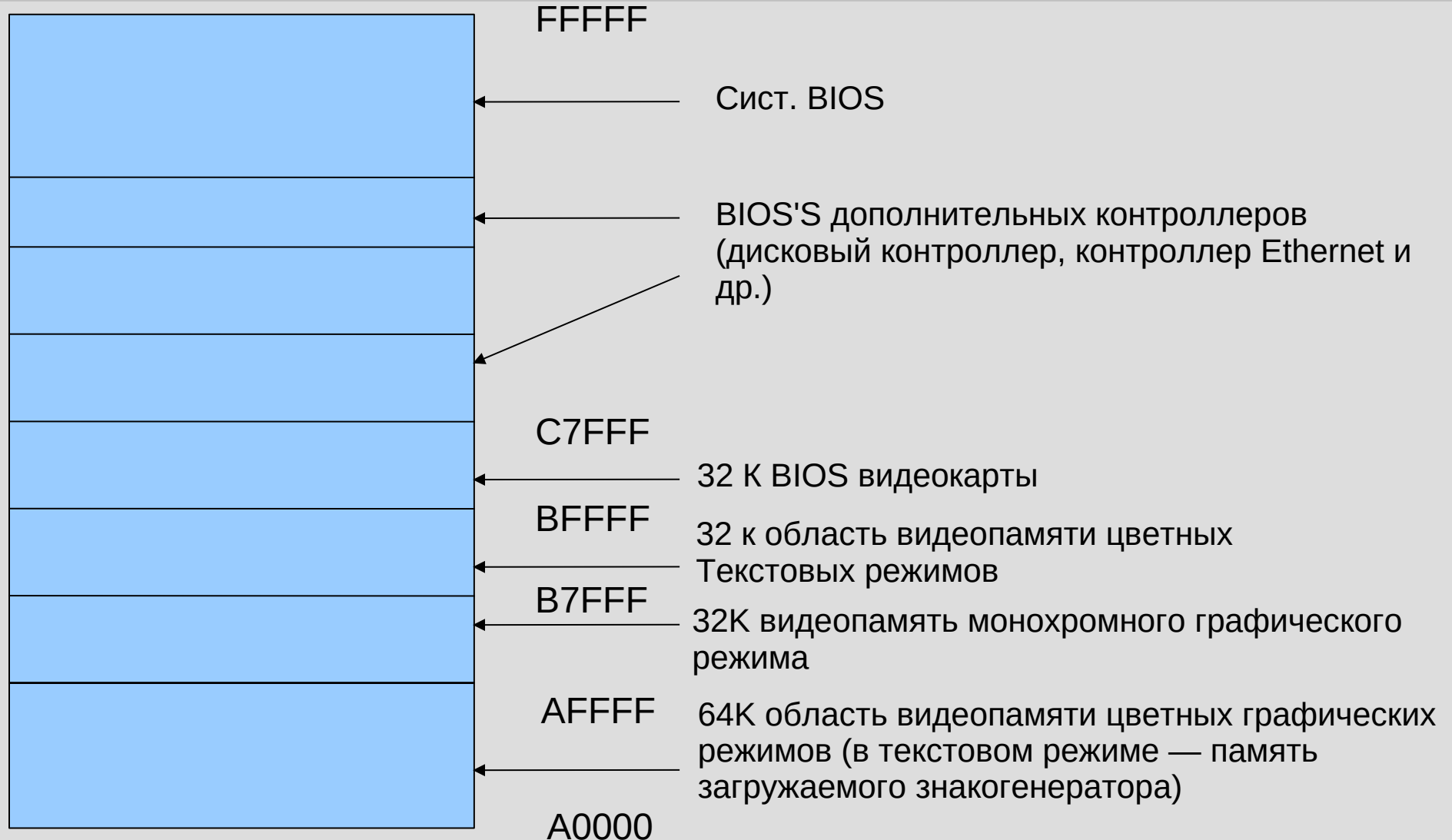
Область контроллеров периферийных устройств

Основная память(conventional memory)

Служебная область BIOS

Область векторов прерываний 1K

# Область памяти контроллеров периферийных устройств





# Пример карты портов I/O реального режима

Устройства	Адреса портов I/O
контроллер прямого доступа к памяти	0 — F, C0 — DF
контроллер прерываний	20 — 21, A0 — A1
спикер	61
контроллер клавиатуры	60,64
таймер	40-43
COM1,COM2	3F8 — 3FF,278 — 27F
контроллер принтера(LPT1)	378-37F
видеоадаптеры	3B0 — 3BB, 3C0 — 3DF
контроллер жесткого диска	1F0 — 1F7,170 — 177
	70-71

# Система прерываний — примеры внутренних аппаратных прерываний

- 0 - деление на ноль; вызываются инструкциями деления DIV и IDIV в случае нулевого делителя;
- 1 - пошаговый режим; прерывание возникает после выполнения каждой команды при установленном флаге трассировки T;
- 2 - немаскируемое прерывание; генерируется процессором при поступлении сигнала на вход NMI;
- 3 - прерывание по точке останова; генерируется в случае, если код очередной команды - CC; используется отладчиками;
- 4 - переполнение; генерируется в случае, если установлен флаг переполнения O и выполняется инструкция INTO;
- 5 - печать экрана (Print Screen); вообще говоря, является внешним, однако традиционно имеет фиксированное назначение и располагается в области внутренних прерываний;
- 6 - (80286 и выше) - неверный код операции;
- 7 - (80286 и выше) - математический сопроцессор недоступен; вызывается при попытке выполнить команду сопроцессора при его отсутствии; может использоваться для эмуляции сопроцессора;
- 8 - (80286 и выше) - обнаружена двойная исключительная ситуация; вызывается в случае возникновения более чем одной исключительной ситуации при выполнении очередной команды; в защищенном режиме вызывается при попытке обратиться за установленный предел таблицы векторов прерываний;
- 9 - (80286 и выше) - попытка обращения к памяти за пределами границы сегмента;
- A - (80286 и выше) - неверный сегмент состояния задачи;
- B - (80286 и выше) - сегмент не найден; возникает при отсутствии требуемого сегмента в памяти в защищенном режиме;
- C - (80286 и выше) - стек переполнен или неверное значение указателя стека;
- D - (80286 и выше) - общее нарушение защиты памяти;
- E - (виртуальный режим 80386 и выше) - используется для реализации виртуальной памяти;
- 10- (80286 и выше) - ошибка сопроцессора.

# Внешние аппаратные прерывания — 2

## каскадированных PIC

- 0 - таймер;
- 1 - клавиатура;
- 2 — каскадирование - управление вторым контроллером прерываний;
- 8 - часы реального времени;
- 9 - программно переводится в 2;
- 10 - Variable ;
- 11 - Variable;
- 12 – PS/2 Mouse;
- 13 - математический сопроцессор;
- 14 - контроллер жесткого диска EIDE/ATA I канал;
- 15 – контроллер жесткого диска EIDE/ATA II канал;
- 3 - прерывания от COM2 ;
- 4 - прерывания от COM1 ;
- 5 - Variable;
- 6 - контроллер гибких дисков;
- 7 - принтер (LPT1).

# Программные прерывания: прерывания BIOS

- 10 - операции ввода/вывода на экран видеомонитора
- 11 - определение конфигурации оборудования
- 12 - определение объема оперативной памяти
- 13 - обмен с дисками
- 14 - организация обмена по последовательным портам
- 15 - ввод/вывод для накопителя на магнитной ленте кассетного типа (устаревшая), работа с XMS памятью
- 16 - ввод с клавиатуры
- 17 - обслуживание принтера
- 18 - инициализация системы кассетного БЕЙСИКа
- 19 - загрузка операционной системы
- 1A - функции даты и времени
- 1B - обработчик прерывания Ctrl-Break
- 1C - "заглушка" для модификации прерывания по таймеру
- 1D - адрес таблицы видеопараметров
- 1E - адрес таблица параметров дискеты
- 1F - адрес второй половины таблицы знакогенератора для графического режима

# Система команд — арифметические инструкции

**ADD** dest,src - сложить два операнда:  $\text{dest} + \text{src} \rightarrow \text{dest}$

**ADC** dest,src - сложить два операнда с учетом переноса  $\text{src} + \text{dest} + \text{cy} \rightarrow \text{dest}$

**INC** dest - увеличить содержимое операнда на 1  $\text{dest} + 1 \rightarrow \text{dest}$

**SUB** dest,src - вычесть из dest src :  $\text{dest} - \text{src} \rightarrow \text{dest}$

**SBB** dest,src - Вычесть с учетом заема:  $\text{dest} - \text{src} - \text{cy} \rightarrow \text{dest}$

**DEC** dest - уменьшить на единицу содержимое dest:  $\text{dest} - 1 \rightarrow \text{dest}$

**NEG** dest - изменить знак операнда:  $0 - \text{dest} \rightarrow \text{dest}$

**MUL** src - беззнаковое умножение:

1. Операнды восьмиразрядные  $(\text{AL}) * \text{src}8 \rightarrow \text{AX}$
2. Операнды шестнадцатиразрядные:  $(\text{AX}) * \text{src}16 \rightarrow \text{DX}:\text{AX}$

**IMUL** src - умножение чисел со знаком

**DIV** src - деление беззнаковое:

1. Восьмиразрядные операнды  $\text{AX} / \text{src}8 \rightarrow \text{AL}; \text{AX} \bmod \text{src}8 \rightarrow \text{AH};$
2. Шестнадцатиразрядный операнд:  $\text{DX}:\text{AX} / \text{src}16 \rightarrow \text{AX}; \text{DX}:\text{AX} \bmod \text{src}16 \rightarrow \text{DX}.$

**IDIV** src - деление чисел со знаком

# Система команд процессора — операции с десятичными числами

- AAA** - Надстройка для сложения чисел в коде ASCII. Корректирует сумму двух байтов в AL. Если правые четыре бита в AL имеют значение больше 9 или флаг AF=1, то команда AAA прибавляет к AH единицу и устанавливает флаги AF и CY. Команда всегда очищает четыре левых бита AL.
- AAS** - Надстройка для вычитания ASCII-чисел. Применяется после вычитания для преобразования AL в две цифры кода ASCII.
- AAM** - Надстройка для умножения ASCII- чисел. Делит содержимое регистра AL на 10, частное помещается в регистр AH, остаток - в регистр AL. Применяется после умножения двух чисел в коде ASCII.
- AAD** - Надстройка для деления ASCII . Применяется перед делением чисел в коде ASCII. Команда корректирует делимое в двоичное значение в регистре AL для последующего двоичного деления. Затем умножает содержимое регистра AH на 10, прибавляет результат к содержимому регистра AL и очищает AH.
- DAA** - Десятичная надстройка для сложения. Корректирует результат операции сложения двух упакованных десятичных чисел с целью получения упакованного десятичного числа.
- DAS** - Десятичная надстройка для вычитания. Корректирует результат вычитания двух упакованных десятичных чисел с целью получения десятичного числа.
- CBW** - Преобразовать БАЙТ в СЛОВО. Исходный операнд в AL, результат в AX
- CWD** - Преобразовать СЛОВО в ДВОЙНОЕ СЛОВО. Исходное слово в AX, рез. в DX:AX

# Система команд — логические инструкции и сдвиги

**AND** dest,src - Логическое И:  $\text{src} \& \text{dest} \rightarrow \text{dest}$

**OR** dest,src - Логическое ИЛИ:  $\text{src} \text{ OR } \text{dest} \rightarrow \text{dest}$

**XOR** dest,src - Исключающее ИЛИ:  $\text{dest} \wedge \text{src} \rightarrow \text{dest}$

**NOT** dest - Логическое НЕ:  $\text{dest} \rightarrow \neg \text{dest}$

**RCL** dest,count - Содержимое dest циклически сдвигается влево через флаг переноса на count двоичных разрядов. В качестве count может использоваться содержимое регистра CL, либо непосредственный операнд 1.

**RCR** dest,count - То же, но сдвиг вправо.

**ROL** dest,count - Циклический сдвиг влево

**ROR** dest,count - Циклический сдвиг вправо

**SAL/SHL** dest,count - Арифметический сдвиг влево

**SAR** dest,count - Арифметический сдвиг вправо

**SHR** dest,count - Логический сдвиг вправо на count разрядов

# Система команд — инструкции пересылки данных

**MOV** dest,src - пересылка данных: src -> dest

**XCHG** dest,src - обмен содержимым операндов: dest <-> src

**LEA** reg16,addr - загрузка адреса : addr -> reg16

**LDS** reg16,mem - загрузка регистра DS и reg16 содержимым памяти, при этом  
reg16 < [mem16]; DS < [mem16+2]

**LES** reg16,mem - загрузка регистра ES и reg16 содержимым памяти, при этом  
reg16 < [mem16]; ES < [mem16+2]

**PUSH** src - запись операнда в стек

**PUSHF** - запись регистра флагов в стек

**POP** src - извлечение операнда из верхушки стека

**POPF** - извлечение регистра флагов из верхушки стека

**LAHF** - Загрузка младшего байта флагов в AH. После выполнения AH содержит:  
значения флагов S Z \* A \* P \* C

**SAHF**- загрузка младшего байта регистра флагов из регистра AH



# Система команд — инструкции пересылки данных (строковые)

**MOVS, MOVSB, MOVSW** - Пересылает данные между областями памяти размером до 64 К. Команда MOVS должна иметь операнды, которые используются ассемблером для определения типа пересылки, команды MOVSB и MOVSW явных операндов не имеют. Пересылка всегда идет из области, адресуемой парой DS:SI в область, адресуемую парой ES:DI. После выполнения пересылки очередного элемента SI и DI увеличиваются (DF=0) или уменьшаются на единицу (для байтов) или на 2 (для слов). Операция обычно применяется с префиксами повторений.

**LODS, LODSB, LODSW** - Помещает в аккумулятор (AL или AX) значение операнда, адресуемого парой DS:SI, за тем, в зависимости от флага DF, адрес операнда инкрементируется или декрементируется

**STOS, STOSB, STOSW** - Операция, обратная LODS

**IN** acc, port ( или DX) - Чтение содержимого порта, заданного адресом port или содержимым DX, в AL или AX

**OUT** acc, port ( или DX) - Запись в порт, заданный адресом port или содержимым DX содержимого AL или AX

**XLAT** src - Транслитерация. Инструкция использует AL, как смещение в 256-байтовой таблице, адресуемой DX:BX. Указанный байт замещает собой значение AL.

# Система команд — переходы и циклы

**JMP** addr - Безусловный переход по адресу addr. Может быть далеким (межсегментным), близким (в пределах сегмента) и коротким (+127/-128 от текущего IP)

**JCXZ** short\_label - Переход, если CX=0 (только короткий).

**LOOP** short\_label - Организация цикла со счетчиком:

1. (CX-1) -> CX
2. jmp short\_label, если CX<>0

**LOOPE** short\_label

**LOOPZ** short\_label

То же, что и LOOP, но переход в том случае, если CX<>0 и ZF=1

**LOOPNE** short\_label

**LOOPNZ** short\_label

То же, что и LOOP, но переход, если CX<>0 и ZF=0

# Система команд процессора — условные переходы

**JA/JNBE** - Более/Не менее и не равно - CF or ZF =0  
**JAЕ/JNB** - Более или равно/Не менее - CF=0  
**JB/JC** - Менее/Перенос - CF=1  
**JE/JZ** — Равно/Нуль - ZF=1  
**JG/JNLE** - Больше/Не меньше и не равно 0 - SF=0 or SF<>OF  
**JGE/JNL** - Больше или равно/Не меньше - SF=0 or ZF=1  
**JL/JNGE** - Меньше/Не больше и не равно - SF=1,ZF<>1  
**JLE/JNG** - Меньше или равно/Не больше - ZF=0 or SF=1  
**JNC** - Нет переноса - CF=0  
**JNE/JNZ** - Не равно/Не нуль - ZF=0  
**JNO** - Нет переполнения - OF=0  
**JNP/JPO** - Нет четности - PF=0  
**JNS** - Нет знака - SF=0  
**JO** — Переполнение - OF=1  
**JP/JPE** – Четность - PF=1  
**JS** — Знак -SF=1

# Система команд процессора — подпрограммы и прерывания

**INT type** - Переход к прерыванию типа type

**INTO** - Прерывание по переполнению. Вызывает прерывание по вектору 4 в случае, если установлен флаг переполнения

**IRET** - Возврат из прерывания

**CALL addr** - Переход к подпрограмме

**CALL Far addr** — Межсегментный переход к п/п

**RET** - Возврат из подпрограммы

**RETF** - Возврат из far процедуры

**RET n** - Возврат с удалением n элементов из верхушки стека

# Управление состоянием процессора

**CMP** dest,src - Сравнить dest и src и установить регистр флагов, операнды не изменяются

**TEST** dest,src - dest&src и установить регистр флагов, операнды не изменяются

**SCAS, SCASB, SCASW** - Сканировать строку. Сравнивается содержимое аккумулятора и элемент, указываемый текущим значением пары ES:DI, соответствующим образом устанавливается регистр флагов. После чего значение DI увеличивается для выборки следующего элемента.

**CLC** - Очистить флаг переноса

**STC** - Установить флаг переноса

**CLI** - Запретить прерывания

**STI** - Разрешить прерывания

**CMC** - Инвертировать флаг переноса (CY)

**CLD** - Очистить флаг направления (установить его в 0 или UP) (для строковых операций)

**STD** - Установить флаг направления (установить его в 1 или DN)(для строковых операций)

**HLT** - Остановить процессор

**WAIT** - Перевести процессор в состояние ожидания внешнего прерывания или сброса

**LOCK** - Блокировка шины на время выполнения следующей команды

**SEG:** - Установить сегмент для последующей адресации (префикс)

# Методы адресации и способы кодирования операндов

Режим адресации	Формат операнда	Регистр	Примеры
регистровая	8 или 16-разрядный регистр	нет	add al,bl mov ax,cs sub bx,dx
непосредственная	8 или 16-разрядный операнд	нет	mov al,10 mov cl,9fh mov ax,235h
Прямая(абсолютная)	Адрес (Внутрисегментное смещение)	DS	mov ax,[100h] mov bx,alfa
косвенно-регистровая	[BX] [BP] [SI] [DI]	DS SS DS DS	sub ax,[bx] add cx,[bp] and ax,[si] mul [di]
базовая адресация	смещение[BX] [смещение+BX] [BX]+смещение смещение[BP] [смещение+BP] [BP]+смещение	DS DS DS SS SS SS	mov ax,tabl[bx] mov ax,[tabl+bx] mov ax,[bx]+tabl mov ax,10[bp] mov ax,[10+bp] mov ax,[bp]+10

# Методы адресации и способы кодирования операндов

Режим адресации	Формат операнда	Регистр	Примеры
индексная адресация	адрес[DI] адрес[SI]	DS DS	or al,100[DI] and dl,mass[SI]
базово-индексная адресация	смещение[BX][SI] смещение[BX][DI] смещение[BP][SI] смещение[BP][DI]	DS DS SS SS	mov ax,[BX][SI]+10 add dx,matr[bx][di] or ax,2[BP][SI] or ax,[BP+SI+4]