

## Архитектура ЭВМ и микроархитектура процессора

Под архитектурой ЭВМ понимают абстрактное представление вычислительной системы на структурном, схемотехническом и логическом уровнях.

1. Архитектура ЭВМ включает:

1.1. Структурную схему вычислительной системы.

1.2. Средства и способы доступа к элементам структурной схемы.

1.3. Организацию и разрядность интерфейсов вычислительной системы

1.4. Архитектуру и микроархитектуру процессора.

1.5. Организацию внутренней (ПЗУ, ОЗУ) и внешней памяти.

## Архитектура ЭВМ и микроархитектура процессора

2. Архитектура процессора включает:

2.1. Набор и доступность регистров (программную модель процессора)

2.2. Режимы работы и логическую организацию памяти в них.

2.3. Способы представления и форматы данных.

2.4. Набор и форматы машинных команд.

2.5. Обработку прерываний.

Микроархитектура процессора – это блочная схема процессора, соответствующая схемотехнической реализации архитектуры процессора.

Поколение	Первый показ модели процессора	Название характерной для поколения процессорной линейки	Микро-архитектура	Процесс (нм)
0		4004, 4040, 8008, 8080, 8085		
1	VI.1978	i8086 (i8088), i80186	86	3000
2	II.1982	i80286	286	1500
3	X.1985	i80386 (DX,SX)	386	1500,1000
4	IV.1989	i80486 (DX,DX2,DX4,SX,SX2)	486	1000, 800, 600
5	III.1993	Pentium (P5,P54C,P54CS)	586	800, 600, 350
6	XI.1995	Pentium Pro, II, III	P6	600, 350, 250, 180, 130
7	XI.2000	Pentium 4	NetBurst	180, 130
8	II.2004	Pentium 4 Prescott, Pentium D	NetBurst	90, 65
9	VII.2006	Core 2 (Duo, Quard)	Conroe, Penryn	65, 45
10	XI.2008	Core i5, i7	Nehalem, Westmere	45, 32
11	2010		Sandy Bridge	32, 22
12	2012			22, 16
13	2014			16, 11
14	2016			11, 8
15	2018			8

# Иерархия памяти [3] Микроархитектура Nehalem

L0: Регистры  
(4 ядра)

L1: 4 кэша L1 (32+32 КБ)

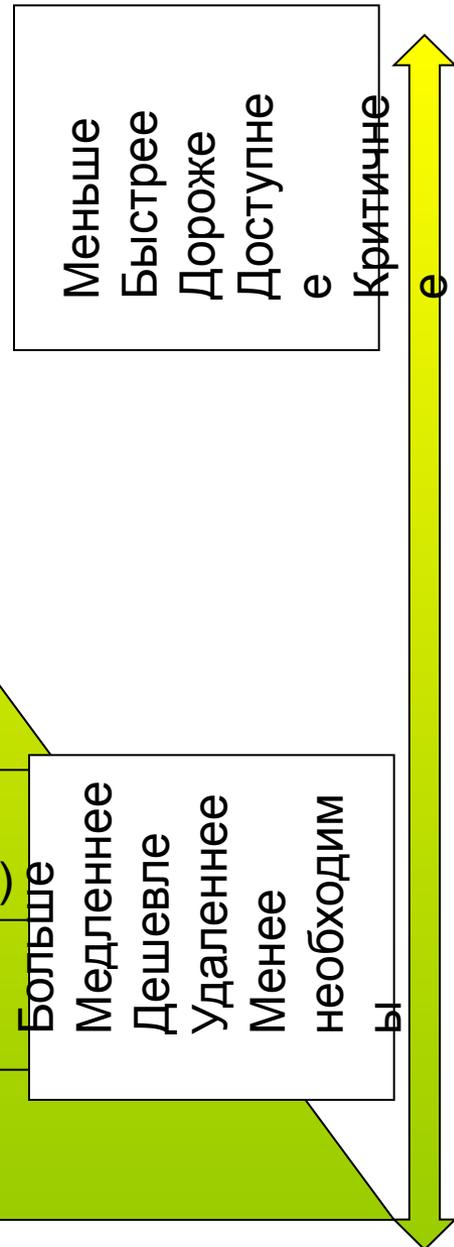
L2: 4 кэша L2 (256 КБ)

L3: Разделяемый кэш L3 (8 МБ)

L4: Память  
(используется встроенный контроллер памяти)

L5: Локальные устройства

L6: Сетевые устройства



## Программная модель процессора [1, 2]

### Архитектура IA-32 до процессора Pentium 4

Программная модель процессора в архитектуре IA-32 включает следующий набор ресурсов:

1. Адресное пространство памяти до  $2^{36}-1$  Б = 64 ГБ ( $2^{32}-1$  Б = 4 ГБ для Pentium II).
2. Набор регистров процессора и сопроцессора.
3. Программный стек.
4. Адресное пространство портов ввода-вывода.
5. Счетчики мониторинга производительности.

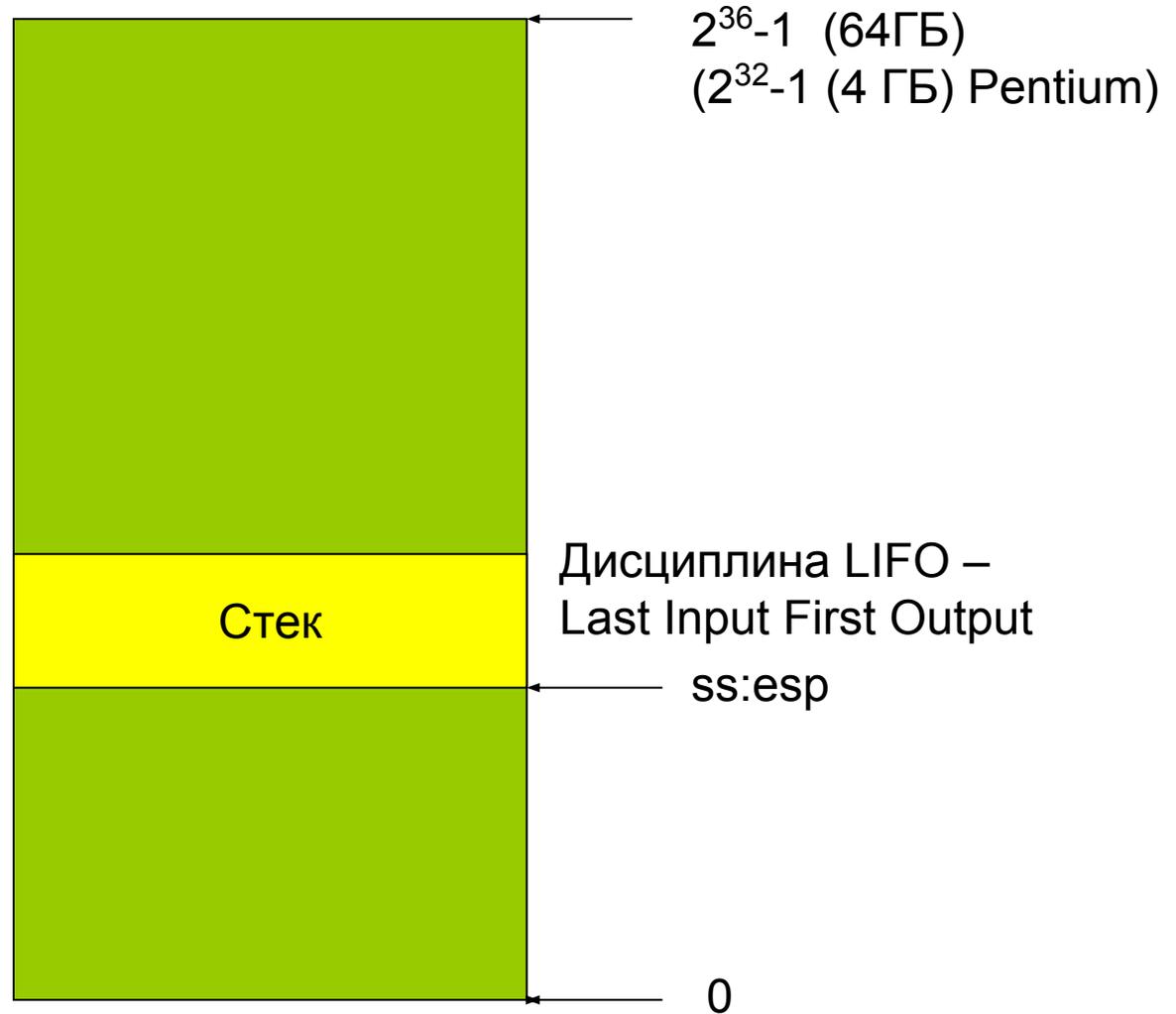
**Аппаратный интерфейс** процессора и памяти использует для передачи данных **физические адреса**.

**Физический адрес** – двоичный код ячейки памяти, устанавливаемый на **шине адреса**.

$n$  проводников (линий) в шине данных позволяют обращаться к  $2^n$  ячейкам памяти.  $n$  – **разрядность** адреса (шины адреса)

**Модель памяти**, в которой адреса ячеек памяти задаются последовательно от 0 до  $2^n - 1$ , называется **линейной** (flat, плоской, сплошной).

# Адресное пространство памяти Pentium 4



**Логическая** адресация осуществляется с использованием блоков смежных адресов, называемые **сегментами**.

В пределах сегмента можно использовать линейную адресацию, начиная с 0 относительно начала сегмента. Такие адреса называются **смещением** (offset).

Адрес начала сегмента называется **базой сегмента**.

Для сегмента можно задавать целевое использование - **тип сегмента**:

различают **сегменты кода, стека и данных**.

# Набор регистров Pentium 4.

## Регистры общего назначения.

Предназначены для хранения данных и адресов.

EAX/AX/AH/AL – accumulator register для хранения промежуточных данных

EBX/BX/BH/BL – base register для хранения базового адреса объекта

ECX/CX/CH/CL – count register для организации циклов

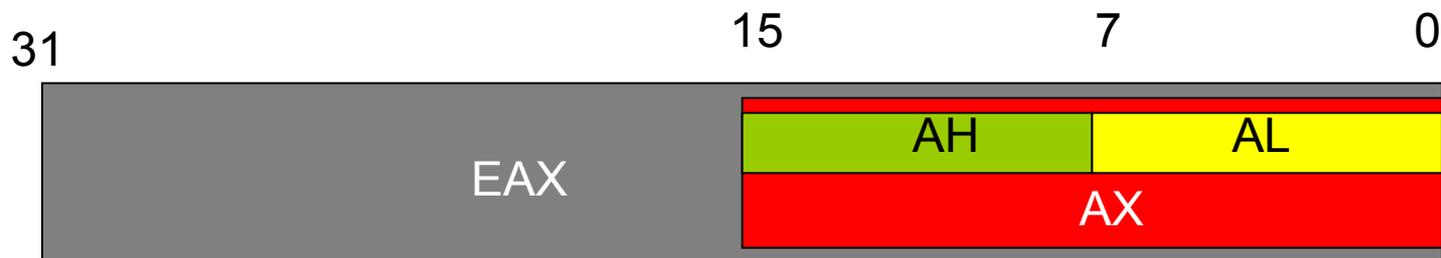
EDX/DX/DH/DL – data register для хранения промежуточных данных

ESI/SI – source index register для текущего адреса элемента в цепочке-источнике

EDI/DI – destination index register для текущего адреса в цепочке-приемнике

ESP/SP – stack pointer register - указатель вершины стека в текущем сегменте стека

EBP/BP – base pointer register для доступа к данным в стеке



Для IA-64 добавляется 64-битные регистры RAX, RBX,...

## Сегментные регистры

Аппаратная поддержка сегментной организации вычислений.  
3 сегмента программы: код (команды), данные, стек.

- CS – code segment register – адрес сегмента с машинными командами
- DS – data segment register – адрес сегмента данных
- ES, FS, GS – extention data segment registers – адреса дополнительных сегментов данных
- SS – stack segment register – адрес сегмента стека.

## Регистры состояния и управления

EIP/IP – instruction pointer register – адрес следующей подлежащей выполнению команде (смещение относительно адреса сегмента кода в CS)

EFLAGS/FLAGS – flag register – регистр флагов состояния, управления и системных флагов.

Для защищенного режима работы процессора используются системные регистры:

5 регистров управления

CR0 – command register 1 – системные флаги

CR1 – зарезервирован

CR2 – страничная организация памяти

CR3 – регистр каталога страниц

CR4 – включение свойств процессора

4 регистра системных адресов:

GDTR (48) - Global Descriptor Table register – предел  
(размер таблицы) (0-15) и базовый адрес GDT

LDTR (16) – Local Descriptor Table register – селектор (указатель)  
дескриптора LDT в GDT

IDTR (48) – Interrupt Descriptor Table register – – предел  
(размер таблицы) (0-15) и базовый адрес IDT

TR (16) – Task register – селектор (указатель)  
дескриптора TSS (task segment status) в GDT  
для переключения задач

8 регистров отладки DR0 – DR7 (32)

DR0 – DR3 – для реализации точек останова

DR4, DR5 - зарезервированы

DR6 состояние отладки (причина останова)

DR7 управления отладкой (условия останова)

## Регистры сопроцессора

8 регистров стека сопроцессора R0-R7 (80)  
0-63 - мантисса, 64-78 – порядок, 79 – знак

3 служебных регистра (16):

SWR – status word register – вершина стека и состояние сопроцессора

CWR – control word register – управление режимами работы  
сoproцессора

TWR – tags word register – состояние регистров стека R0-R7  
(по 2 бита на регистр)

2 регистра указателей (48)

DPR – data point register – адрес операнда команды, вызвавшей  
исключение

IPR – instruction point register - адрес команды, вызвавшей  
исключение

## Регистры целочисленного MMX-расширения

### MultiMedia eXtensions начиная с Pentium MMX

MMX реализует целочисленные SIMD (Single Instruction Multiple Data) вычисления

При выполнении целочисленных MMX-операций сопроцессор переводится в специальный режим, в котором разряды мантииссы (0-63) регистров R0-R7 (80) используются в качестве регистров RMMX0-RMMX7 (64).

Целочисленные регистры MMX отображены на регистры сопроцессора. Биты 64-79 устанавливаются в 1 для предотвращения исключений.

## Регистры XMM-расширения

eXtended MultiMedia начиная с Pentium III

XMM-вычисления реализуются модулем SSE (Streaming SIMD Extentions). В Pentium 4 добавлен модуль SSE2.

8 регистров для данных с плавающей точкой:

XMM0 – XMM7

в каждый регистр помещается 4 числа с плавающей точкой в коротком формате, упакованных в формат 32 бита (XMM-формат)

0-23 – мантисса, 24-30 – порядок, 31 - знак

1 регистр управления/состояния:

MXCSR – Multimedia eXtentions Control and Status register (32)