x86-64指令系统概述

南京大学
计算机科学与技术系
袁春风
email: cfyuan@nju.edu.cn
2015.6

回顾：Intel处理器

x86前产品	4004 • 4040 • 8008 • 8080 • iAPX 432 • 8085	
x87（外置浮点运算器）	8/16位总线: 8087 16位总线: 80187 • 80287 • 80387SX 32位总线: 80387DX • 80487	已停产
x86-16（16位）	8086 • 8088 • 80186 • 80188 • 80286	
x86-32/IA-32（32位）	80386 • 80486 • Pentium（OverDrive、Pro、II、III、4、M） • Celeron（M、D） • Core	
x86-64/Intel 64（64位）	Pentium（4（部份型号）、Pentium D、EE） • Celeron D（部份型号） • Core 2	
EPIC/IA-64（64位）	Itanium	
RISC	i860 • i960 • StrongARM • XScale	
微控制器	8048 • 8051 • MCS-96	
x86-32/IA-32	EP80579 • A100 • Atom（CE、SoC）	现有产品
x86-64/Intel 64	Xeon（E3、E5、E7、Phi） • Atom（部分型号） • Celeron • Pentium • Core（i3、i5、i7）	
EPIC/IA-64	Itanium 2	

x86-64架构

- 背景
  - Intel最早推出的64位架构是基于超长指令字VLIW技术的IA-64体系结构，Intel 称其为显式并行指令计算机EPIC（Explicitly Parallel Instruction Computer）。
  - 安腾和安腾2分别在2000年和2002年问世，它们是IA-64体系结构的最早的具体实现，因为是一种全新的、与IA-32不兼容的架构，所以，没有获得预期的市场份额。
  - AMD公司利用Intel在IA-64架构上的失败，抢先在2003年推出兼容IA-32的64位版本指令集x86-64，AMD获得了以前属于Intel的一些高端市场。AMD后来将x86-64更名为AMD64。
  - Intel在2004年推出IA32-EM64T，它支持x86-64指令集。Intel为了表示EM64T的64位模式特点，又使其与IA-64有所区别，2006年开始把EM64T改名为Intel64。

回顾：IA-32支持的数据类型及格式

C 语言声明	Intel 操作数类型	汇编指令长度后缀	存储长度（位）
(unsigned) char	整数 / 字节	b	8
(unsigned) short	整数 / 字	w	16
(unsigned) int	整数 / 双字	l	32
(unsigned) long int	整数 / 双字	l	32
unsigned) long long int	–	–	2×32
char *	整数 / 双字	l	32
float	单精度浮点数	s	32
double	双精度浮点数	l	64
long double	扩展精度浮点数	t	80 / 96

IA-32架构由16位架构发展而来，因此，虽然字长为32位或更大，但一个字为16位，长度后缀为 w；32位为双字，长度后缀为 l

long double实际长度为80位，但分配96位=12B（按4B对齐）

x86-64中各类数据的长度

C 语言声明	Intel 操作数类型	汇编后缀	x86-64（位）	IA-32（位）
(unsigned) char	整数 / 字节	b	8	8
(unsigned) short	整数 / 字	w	16	16
(unsigned) int	整数 / 双字	l	32	32
(unsigned) long int	整数 / 四字	q	64	32
unsigned) long long int	整数 / 四字	q	64	2×32
char *	整数 / 四字	q	64	32
float	单精度浮点数	s	32	32
double	双精度浮点数	d	64	64
long double	扩展精度浮点数	t	80 / 128	80 / 96

x86-64的通用寄存器

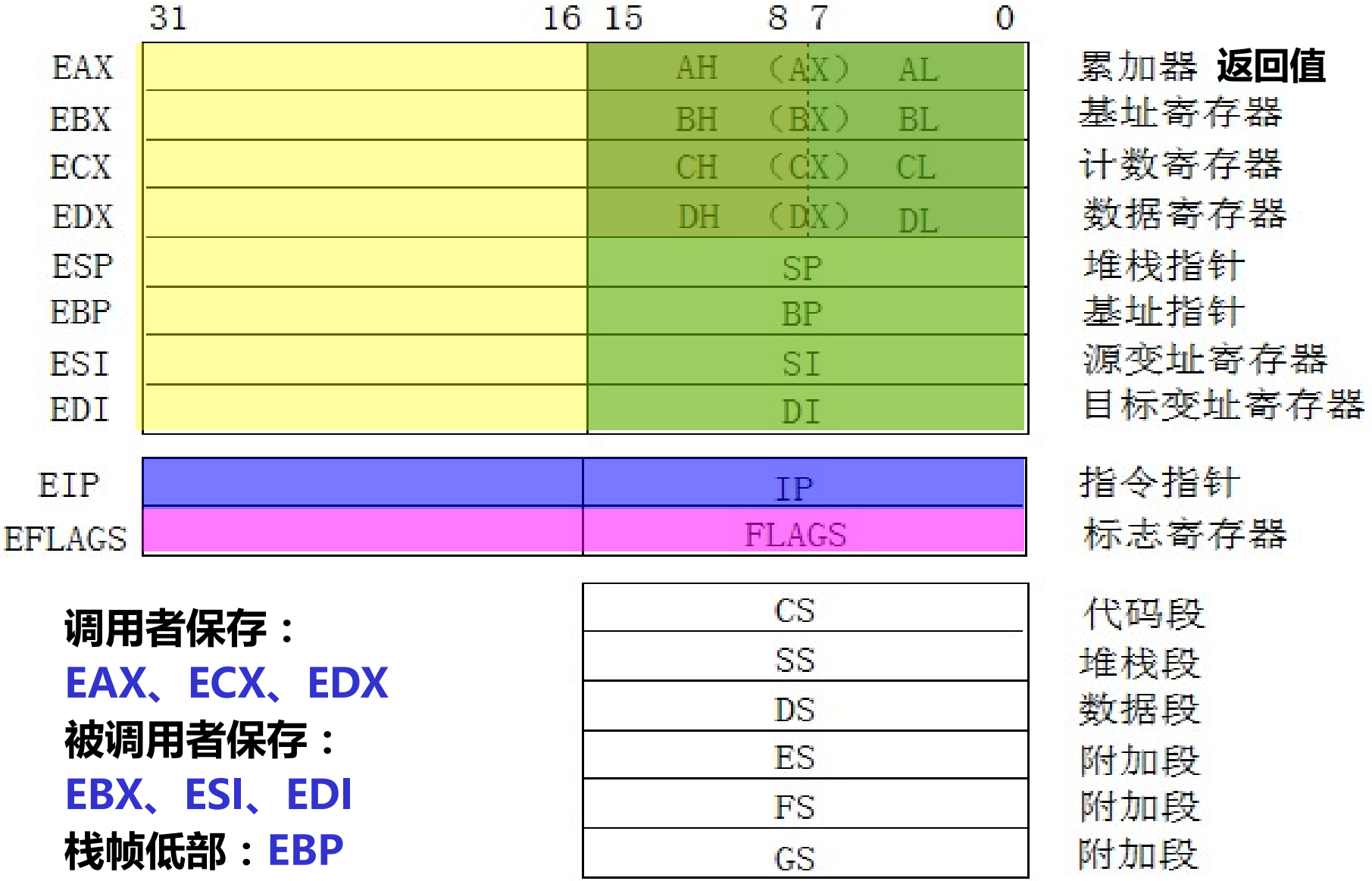
- 新增8个64位通用寄存器（整数寄存器）
  - R8、R9、R10、R11、R12、R13、R14和R15。
  - 可作为8位（R8B~R15B）、16位（R8W~R15W）或32位寄存器（R8D~R15D）使用
- 所有GPRs都从32位扩充到64位
  - 8个32位通用寄存器EAX、EBX、ECX、EDX、EBP、ESP、ESI和 EDI对应扩展寄存器分别为RAX、RBX、RCX、RDX、RBP、RSP、RSI和RDI
  - EBP、ESP、ESI和 EDI的低8位寄存器分别是BPL、SPL、SIL和DIL
  - 可兼容使用原AH、BH、CH和DH寄存器

（使原来IA-32中的每个通用寄存器都可以是8位、16位、32位和64位，如：SIL、SI、ESI、RSI）

x86-64中寄存器的使用

- 指令可直接访问16个64位寄存器：RAX、RBX、RCX、RDX、RBP、RSP、RSI、RDI，以及R8~R15
- 指令可直接访问16个32位寄存器：EAX、EBX、ECX、EDX、EBP、ESP、ESI、EDI，以及R8D~R15D
- 指令可直接访问16个16位寄存器：AX、BX、CX、DX、BP、SP、SI、DI，以及R8W~R15W
- 指令可直接访问16个8位寄存器：AL、BL、CL、DL、BPL、SPL、SIL、DIL，以及R8B~R15B
- 为向后兼容，指令也可直接访问AH、BH、CH、DH
- 通过寄存器传送参数，因而很多过程不用访问栈，因此，与IA-32不同，x86-64不需要帧指针寄存器，即RBP可用作普通寄存器使用
- 程序计数器为64位寄存器RIP

回顾：IA-32的寄存器组织



调用者保存：

EAX、ECX、EDX

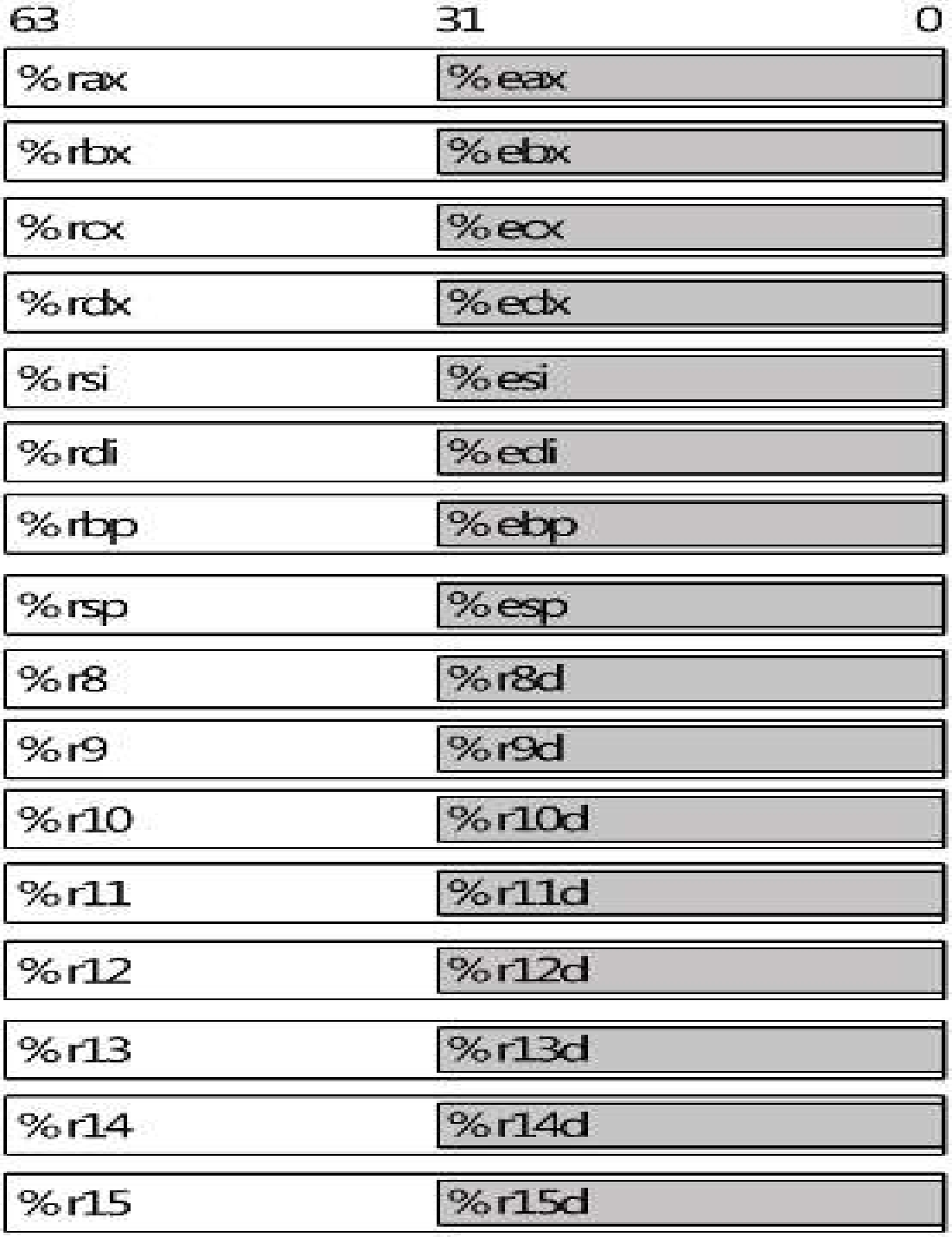
被调用者保存：

EBX、ESI、EDI

栈帧低部：EBP

栈帧顶部：ESP

寄存器



通用寄存器个数从8个增加到16个，宽度从32位增加到64位

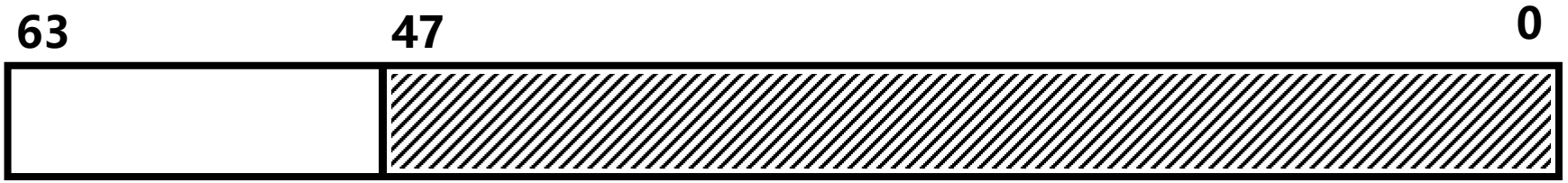
增加了 %sil、%dil、%bpl、%spl 四个8位寄存器

%riw为16位

%rib为8位（i=8~15）

x86-64的地址和寻址空间

- 字长从32位变为64位，64位（8B）数据被称为一个四字（qw: quadword）
- 逻辑地址最长可达为64位，即理论上可访问的存储空间达2^{64}字节或16EB（ExaByte）
- 编译器为指针变量分配64位（8B）
- 基址寄存器和变址寄存器都应使用64位寄存器
- 但实际上，AMD和Intel的x86-64仅支持48位虚拟地址，因此，程序的虚拟地址空间大小为2^{48}=256TB



x86-64的浮点寄存器

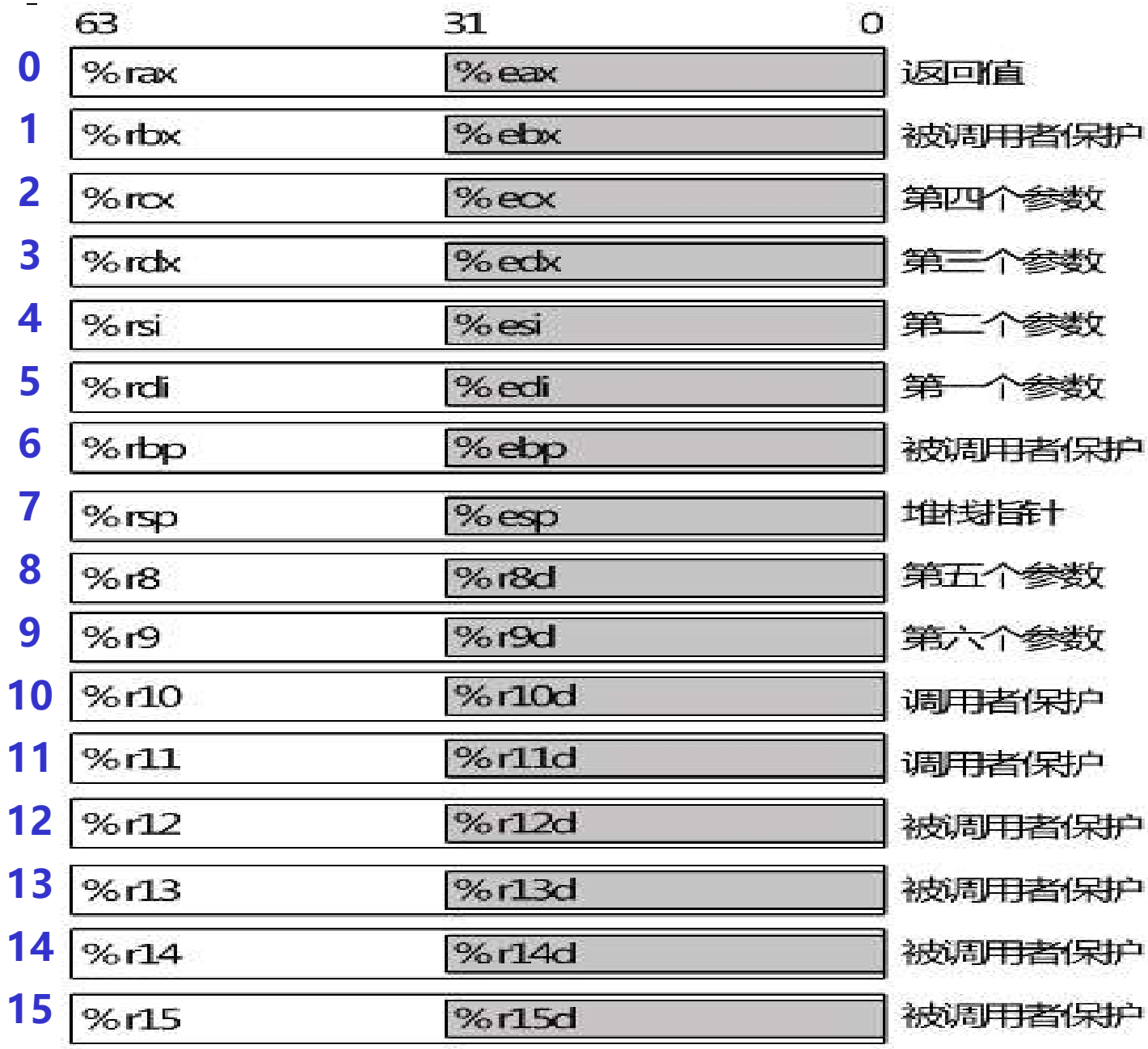
- long double型数据虽然还采用80位（10B）扩展精度格式，但所分配存储空间从12B扩展为16B，即改为16B对齐方式，但不管是分配12B还是16B，都只用到低10B
- 128位的XMM寄存器从原来的8个增加到16个
- 浮点操作指令集采用基于SSE的面向XMM寄存器的指令集，而不采用基于浮点寄存器栈的 x87 FPU 指令集
- 浮点操作数存放在XMM寄存器中

回顾：IA-32的寄存器组织

编号	8 位寄存器	16 位寄存器	32 位寄存器	64 位寄存器	128 位寄存器
000	AL	AX	EAX	MM0 / ST(0)	XMM0
001	CL	CX	ECX	MM1 / ST(1)	XMM1
010	DL	DX	EDX	MM2 / ST(2)	XMM2
011	BL	BX	EBX	MM3 / ST(3)	XMM3
100	AH	SP	ESP	MM4 / ST(4)	XMM4
101	CH	BP	EBP	MM5 / ST(5)	XMM5
110	DH	SI	ESI	MM6 / ST(6)	XMM6
111	BH	DI	EDI	MM7 / ST(7)	XMM7

x86-64继承了IA-32中的8、16、32位通用寄存器和128位XMM寄存器
而取消了IA-32中的80位浮点寄存器栈ST(0)-ST(7)

x86-64的寄存器



增加了 %sil、%dil、%bpl、%spl 四个8位寄存器

%riw为16位
%rib为8位
（i=8~15）

16个128位寄存器%xmmi
（i=0~15）

x86-64中数据的对齐

- 各类型数据遵循一定的对齐规则，而且更严格
- 存储器访问接口被设计成按8字节或16字节为单位进行存取，其对齐规则是，任何K字节宽的基本数据类型和指针类型数据的起始地址一定是K的倍数。
  - short型数据必须按2字节边界对齐
  - int、float等类型数据必须按4字节边界对齐
  - long、double、指针型变量必须按8字节边界对齐
  - long double型数据必须按16字节边界对齐