

15213/18-243 Exam 1 Notes Sheet

Spring 2010

jX	Condition
jmp	1
je	ZF
jne	~ZF
js	SF
jns	~SF
jg	~(SF^OF) & ~ZF
jge	~(SF^OF)
jl	(SF^OF)
jle	(SF^OF) ZF
ja	~CF & ~ZF
jb	CF

Format

addl *Src, Dest*

subl *Src, Dest*

imull *Src, Dest*

sall *Src, Dest*

sarl *Src, Dest*

shrl *Src, Dest*

xorl *Src, Dest*

andl *Src, Dest*

orl *Src, Dest*

Computation

Dest = Dest + Src

Dest = Dest - Src

Dest = Dest * Src

Dest = Dest << Src

Dest = Dest >> Src

Dest = Dest >> Src

Dest = Dest ^ Src

Dest = Dest & Src

Dest = Dest | Src

$$\text{Bias} = 2^{k-1} - 1$$

(Rb, Ri) Mem[Reg[Rb]+Reg[Ri]]

D(Rb, Ri) Mem[Reg[Rb]+Reg[Ri]+D]

(Rb, Ri, S) Mem[Reg[Rb]+S*Reg[Ri]]

Big Endian: Sun, PPC Mac, Internet

- Least significant byte has highest address

Little Endian: x86

- Least significant byte has lowest address

Specific Cases of Alignment (IA32)

- **1 byte: char, ...**
 - no restrictions on address
- **2 bytes: short, ...**
 - lowest 1 bit of address must be 0₂
- **4 bytes: int, float, char *, ...**
 - lowest 2 bits of address must be 00₂
- **8 bytes: double, ...**
 - Windows (and most other OS's & instruction sets):
 - lowest 3 bits of address must be 000₂
 - Linux:
 - lowest 2 bits of address must be 00₂
 - i.e., treated the same as a 4-byte primitive data type
- **12 bytes: long double**
 - Windows, Linux:
 - lowest 2 bits of address must be 00₂
 - i.e., treated the same as a 4-byte primitive data type

Specific Cases of Alignment (x86-64)

- **1 byte: char, ...**
 - no restrictions on address
- **2 bytes: short, ...**
 - lowest 1 bit of address must be 0₂
- **4 bytes: int, float, ...**
 - lowest 2 bits of address must be 00₂
- **8 bytes: double, char *, ...**
 - Windows & Linux:
 - lowest 3 bits of address must be 000₂
- **16 bytes: long double**
 - Linux:
 - lowest 3 bits of address must be 000₂
 - i.e., treated the same as a 8-byte primitive data type

63	31	15	8	7	0	
%rax	%eax %ax	%ah	%al			Return value
%rbx	%ebx %bx	%bh	%bl			Callee saved
%rcx	%ecx %cx	%ch	%cl			4th argument
%rdx	%edx %dx	%dh	%dl			3rd argument
%rsi	%esi %si		%sil			2nd argument
%rdi	%edi %di		%dil			1st argument
%rbp	%ebp %bp		%bpl			Callee saved
%rsp	%esp %sp		%spl			Stack pointer
%r8	%r8d %r8w		%r8b			5th argument
%r9	%r9d %r9w		%r9b			6th argument
%r10	%r10d %r10w		%r10b			Callee saved
%r11	%r11d %r11w		%r11b			Used for linking
%r12	%r12d %r12w		%r12b			Unused for C
%r13	%r13d %r13w		%r13b			Callee saved
%r14	%r14d %r14w		%r14b			Callee saved
%r15	%r15d %r15w		%r15b			Callee saved