| 4 字（64bit，8Byte） | 2 字（32bit，4Byte） | 1 字（16bit，2Byte） | 1 字节（8bit，1Byte） |
|---|---|---|---|
| %rax | %eax | %ax | %al |
| %rbx | %ebx | %bx | %bl |
| %rcx | %ecx | %cx | %cl |
| %rdx | %edx | %dx | %dl |
| %rsi | %esi | %si | %sil |
| %rdi | %edi | %di | %dil |
| %rbp | %ebp | %bp | %bpl |
| %rsp | %esp | %sp | %spl |
| %r8 | %r8d | %r8w | %r8b |
| %r9 | %r9d | %r9w | %r9b |
| %r10 | %r10d | %r10w | %r10b |
| %r11 | %r11d | %r11w | %r11b |
| %r12 | %r12d | %r12w | %r12b |
| %r13 | %r13d | %r13w | %r13b |
| %r14 | %r14d | %r14w | %r14b |
| %r15 | %r15d | %r15w | %r15b |

字节控制标识符：b（1 Byte）,w(2 Byte),l(4Byte),q(8 Byte)，表格中 x,y 为标识符

| | | |
|---|---|---|
| 移动寄存器中值 | mov+x A B | A-->B |
| | movz+x y A B | A(零扩展）-->B |
| | movs+x y A B | A(符号扩展)-->B |
| | movabsq A B | A-->B（绝对四字，仅寄存器） |
| | cltq %eax %rax | %eax(符号扩展）-->%rax |
| 堆栈 | pushq S | 将 4 字压入栈 |
| | popq D | 将四字弹出栈 |
| 计算 | Leaq S D | 加载有效地址，&S-->D |
| | Inc/Dec/Neg/Not D | +1/-1/求负数/取反 |
| | Add/Sub/Imul/Xor/Or/And A B | B+/-/*/^/\|/&A-->B |
| | SAL/SHL/SAR/SHR k A | A<</<</算术 >>/逻辑 >>k -->A |
| 标识码 | CF/ZF/SF/OF（标识码） | 无符号是否溢出/0/负数/补码溢出(leaq 不改变） |
| 比较 | cmp+x A B | 基于 B-A 设置标识码 |
| | test+x A B | 基于 B&A 设置标识码 |
| 设置值 | set(n)e/set(n)z D | 判断（不）为 0 |
| | set(n)s D | 判断（不）为负 |
| | setg(e)/setnl(e) D | 判断大于（大于等于）（有符号） |
| | setl(e)/setng(e) D | 判断小于（小于等于）（有符号） |
| | seta(e)/setnb(e) D | 判断超过（超过或相等）（无符号） |
| | setb(e)/setna(e) D | 判断低于（低于或相等）（无符号） |
| 按条件跳转 | jmp label/*operand | 直接跳转 |
| | j(n)e/j(n)z label | 若（不）为 0 跳转 |
| | j(n)s label | 若（不）为负数跳转 |

| | | |
|---|---|---|
| | jg(e)/jnl[e] label | 按大于（大于等于）跳转（有符号） |
| | jl(e)/jng[e] label | 按小于（小于等于）跳转（有符号） |
| | ja(e)/jnb[e] label | 按超过（超过或相等）跳转（无符号） |
| | jb(e)/jna[e] label | 按低于（低于或相等）跳转（无符号） |
| 按条件传送 | Cmov(n)e/cmov(n)z A B | 当（不）为 0 时进行 A->B |
| | Cmov(n)s A B | 当（不）为负数时进行 A->B |
| | Cmovg(e)/cmovnl[e] A B | 当大于（大于等于）时进行 A->B（有符号） |
| | Cmovl(e)/cmovng[e] A B | 当小于（小于等于）时进行 A->B（有符号） |
| | Cmova(e)/cmovnb[e] A B | 当超过（超过或相等）时进行 A->B（无符号） |
| | Cmovb(e)/cmovna[e] A B | 当低于（低于或相等）时进行 A->B（无符号） |