Assembly

Registers:

8-byte register	Bytes 0-3	Bytes 0-1	Byte 0
%rax	%eax	%ax	%al
%rcx	%ecx	%cx	%cl
%rdx	%edx	%dx	%dl
%rbx	%ebx	%bx	%bl
%rsi	%esi	%si	%sil
%rdi	%edi	%di	%dil
%rsp	%esp	%sp	%spl
%rbp	%ebp	%bp	%bpl
%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

l: 4 bytes, 32 bits

q: 8 bytes, 64 bits

mov S, D Move source to destination

add S, D Add source to destination

sub S, D Subtract source from destination

imul S, D Multiply destination by source

inc D Increment by 1

dec D Decrement by 1

or S, D Bitwise OR destination by source

and S, D Bitwise AND destination by source

cmp S2, S1 Set condition codes according to S1 - S2

jmp Label	Jump to label
jg Label	Jump if greater
je Label	Jump if equal
jge Label	Jump if greater or equal
jl Label	Jump if less
jle Label	Jump if less or equal

push S	Push source onto stack
pop D	Pop top of stack into destination

- “Index Addressing”

  subq % rdx, (%rsi, %rdi, 4) // the address used here is %rsi + (%rdi * 4)

  // address is 1000 + (30 * 4) = 1120.
- “Indexed + offset”

  24(%rcx, %rdi, 8)

  // address is 24 + %rcx + (%rdi * 8)

- “Caller Saved” registers — callee can overwrite
- “Callee Saved” registers — when callee returns, must remain the same