Assembly Arithmetic and Logic

Monday, October 19, 2020 12:40 PM

Data Sizes

Byte		1 byte	b
Word		2 bytes	w
Double word	(dword)	4 bytes	l
Quad word	(qword)	8 bytes	q

Register Sizes

63	31	15	7	0
rax	eax	ax	al	
rbx	ebx	bx	bl	
rcx	ecx	cx	cl	
rdx	edx	dx	dl	
rsi	esi	si	sil	
rdi	edi	di	dil	

Larger registers contain smaller registers
(b/c backwards compatibility)

Some registers have special responsibilities

mov Variants

→ Can take optional suffix for size of data to move

movb, movw, movl, movq

→ Only updates specific bytes, except for movl which 0's out high order 4 bytes

→ Move small to large - movz fills w/ 0's
movs sign-extends MSB

src mem or register,
dst register

The lea instruction

lea - Load Effective Address

Copies value of src to dst
(no mem addr involved)

Ex.

op	mov	lea
6(%rax), %rdx	Copy contents of addr at src to dst	Copy 6 + (%rax val) to %rdx

Unary instructions

Operate on single operand (register or memory)
Can take suffix like mov

inc D	$D \leftarrow D+1$	increment
dec D	$D \leftarrow D-1$	decrement
neg D	$D \leftarrow -D$	negate
not D	$D \leftarrow \sim D$	complement

Binary instructions

Operate on two operands (register or memory),
both cannot be memory

add	S, D	$D \leftarrow D + S$	add
sub	S, D	$D \leftarrow D - S$	subtract
imul	S, D	$D \leftarrow D * S$	multiply
xor	S, D	$D \leftarrow D \wedge S$	xor
or	S, D	$D \leftarrow D \mid S$	or
and	S, D	$D \leftarrow D \,\&\, S$	and

Can take suffix like mov

imul on 64-bit operands will truncate to 64-bits

imul w/ one operand will multiply by %rax and
put high-order 64-bits in %rdx and low in %rax

mul on one operand - unsigned multiply

idivq	Signed	divide w/ modulo
divq	Unsigned	

Can divide 128 bits by 64 bits

		$D \leftarrow D \ll K$	left shift
sal	K, D	$D \leftarrow D \ll K$	left shift
shl	K, D	$D \leftarrow D \gg_A K$	arithmetic rshift
sar	K, D	$D \leftarrow D \gg_L K$	logical rshift
shr	K, D		

K must be immediate, or value in %cl if not present