LESSON 5



ASM

Math & Flags

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

ASSEMBLY LANGUAGE

LESSON 5

Arithmetic, Logic and Flags

add, sub, inc, dec, mul, imul, div, and, or, xor, shifts and the RFLAGS register

add, sub, inc, dec, neg

math.asm

```
mov rax, 10
add rax, 5          ; rax = 15
sub rax, 3          ; rax = 12
inc rax             ; rax = 13
dec rax             ; rax = 12
neg rax             ; rax = -12
add [total], rax    ; memory += rax
```

- add and sub put the result in the first operand. inc and dec add or subtract one, and neg …
- Every arithmetic instruction also updates the flags, which we use for decisions in the next …

01

Multiplication and Division

divide.asm

```
imul rax, rbx, 10    ; rax = rbx * 10

mov rax, 17
xor rdx, rdx         ; clear high half
mov rcx, 5
div rcx              ; rax = 3 (quotient)
                     ; rdx = 2 (remainder)

mov rax, -17
cqo                  ; sign-extend rax into rdx
idiv rcx             ; rax = -3, rdx = -2
```

- imul with two operands multiplies signed numbers simply. The one-operand forms mul and imul …
- div and idiv divide rdx colon rax by the operand: the quotient goes to rax and the …

02

Bitwise Logic

Instruction	al before	al after
`and al, 0x0F`	`0100 0001`	`0000 0001`
`or al, 0x20`	`0100 0001 'A'`	`0110 0001 'a'`
`xor al, 0xFF`	`0100 0001`	`1011 1110`
`not al`	`0100 0001`	`1011 1110`

- and keeps bits that are one in both operands, or sets bits, xor flips them, and not inverts …
- These are used for masks and flags: and with fifteen keeps the low four bits; or with …

03

Shifts and Rotates

ASSEMBLY LANGUAGE · 5

shift.asm

```
mov rax, 5
shl rax, 3        ; 5 * 8 = 40
shr rax, 1        ; 40 / 2 = 20
mov rbx, -16
sar rbx, 2        ; -16 / 4 = -4 (sign kept)
rol al, 4         ; swap the two hex digits of al
```

- shl shifts bits left, which multiplies by two for each position. shr shifts right, dividing …
- Compilers replace multiplying by powers of two with shifts because they are faster.

04

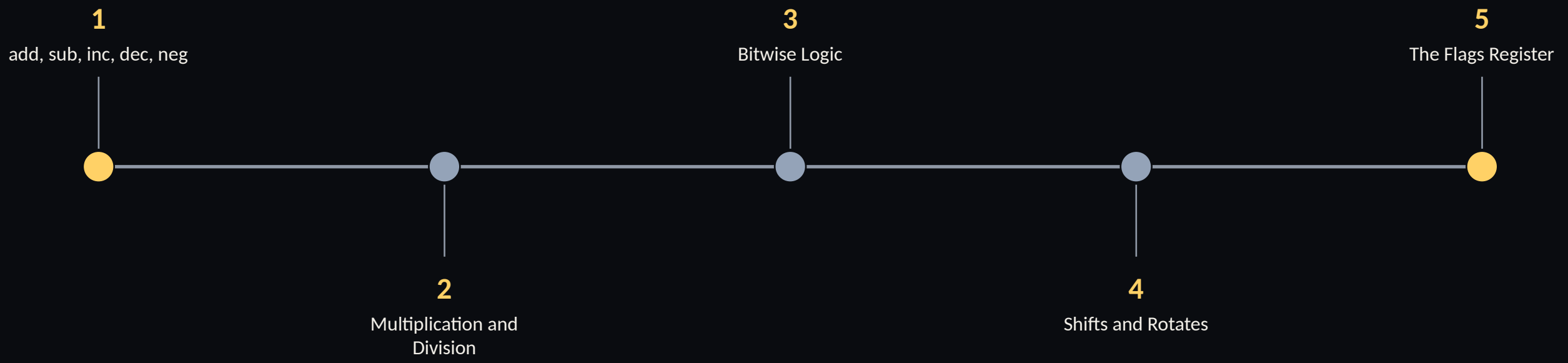
The Flags Register

Flag	Set when	Example
ZF	result is zero	`sub rax, rax`
SF	result is negative	`sub 3, 5`
CF	unsigned carry/borrow	`add al, 200 (al=100)`
OF	signed overflow	`add al, 100 (al=100)`

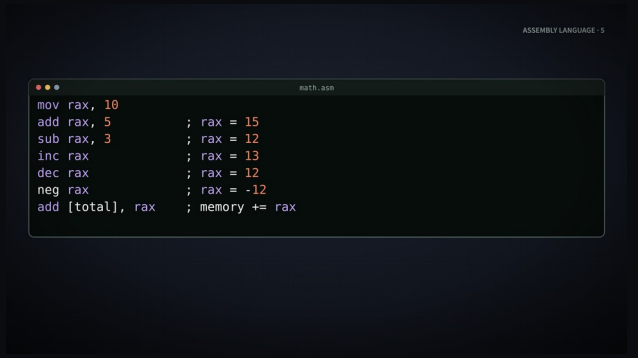
- After most arithmetic, the CPU sets flags in RFLAGS. ZF means the result was zero, SF means ...
- Conditional jumps in the next lesson read these flags.

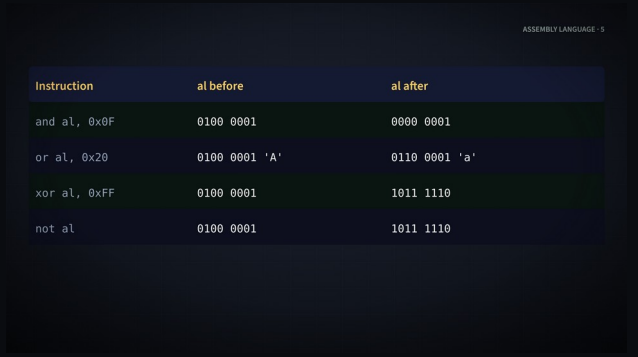
05

Lesson Map


1
add, sub, inc, dec, neg
2
Multiplication and Division
3
Bitwise Logic
4
Shifts and Rotates
5
The Flags Register

Sources and Further Reading





Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari