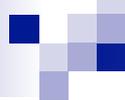


Bitmasks (Practice)

ICS312 Machine-Level and Systems Programming

Henri Casanova (henric@hawaii.edu)



(q1) Division by a Power of 2

- Write one instruction that computes `eax % 64`

(q1) Solutions

- Write one instruction that computes $\text{eax} \% 64$

`and eax, 00000003Fh`

`(and eax, 111111b)`

`(and eax, 63)`

- Mask out the most significant $32-6=26$ bits, and keep the least significant 6 bits (because $64 = 2^6$)
 - $00111111_2 = 3F_{16}$

(q2) Create a Bitmask

- Write code to flip the n^{th} bit of EAX, counting from right to left from the least significant bit, where n is stored in cl

(q2) Solutions

- Write code to flip the n^{th} bit of EAX, counting from right to left from the least significant bit, where n is stored in cl

```
mov    ebx, 1           ; 0—————01
shl    ebx, cl          ; 0————010—0
xor    eax, ebx
```

(q3) Create a Bitmask

- Write code to create in `eax` a 32-bit bitmask that consists of 0's followed by `n` 1's, where `n` is stored in `cl`

(q3) Solutions

- Write code to create in `eax` a 32-bit bitmask that consists of 0's followed by `n` 1's, where `n` is stored in `cl`

```
mov  eax, 0FFFFFFFFh ; 1—————1
shl  eax, cl          ; 1———10—0
not  eax              ; 0———01—1
```

(q4) Create a Bitmask

- Write code to create in `eax` a 32-bit bitmask that contains n 1's, followed by $32-2n$ 0's, followed by n 1's, where n is stored in `cl`

(q4) Solutions

- Write code to create in `eax` a 32-bit bitmask that contains n 1's, followed by $32-2n$ 0's, followed by n 1's, where n is stored in `cl`

```
mov  eax,  0FFFFFFFFh    ; 1—————1
shl  eax,  cl            ; 1————0—0
shl  eax,  cl            ; 1———0——0
ror  eax,  cl            ; 0—01——10—0
not  eax                ; 1—10——01—1
```