

Organisasi Sistem Komputer

Bab 6. Operasi Bit

6.1 Pergeseran Bit

6.2 Operasi Boolean



Instruksi AND, OR, XOR

- Sintaks:

AND *destination, source*

OR *destination, source*

XOR *destination, source*

- Kombinasi operand destination dan source:

- *reg, reg*
- *reg, mem*
- *reg, imm*
- *mem, imm*
- *mem, reg*

Instruksi AND

Tabel Kebenaran AND

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

source

1 1 1 1 1 1 1 1

AND AND AND AND AND AND AND AND

destination

0 0 0 1 1 1 0 0

AND *destination, source*

0 0 0 1 1 1 0 0

- EFLAGS: Instruksi AND meng-nol-kan flag Carry dan Overflow namun memodifikasi flag Sign, Zero, dan Parity sesuai dengan hasilnya

Instruksi OR

Tabel Kebenaran OR

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

source

0 1 1 0 1 1 0 0

OR OR OR OR OR OR OR OR

destination

0 0 0 1 1 1 0 0

OR *destination, source*

0 1 1 1 1 1 0 0

- **EFLAGS:** Instruksi OR meng-nol-kan flag Carry dan Overflow namun memodifikasi flag Sign, Zero, dan Parity sesuai dengan hasilnya

Instruksi XOR

Tabel Kebenaran XOR

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

source

0 1 1 0 1 1 0 0

XOR XOR XOR XOR XOR XOR XOR XOR

destination

0 0 0 1 1 1 0 0

XOR *destination, source*

0 1 1 1 0 0 0 0

- **EFLAGS:** Instruksi XOR meng-nol-kan flag Carry dan Overflow namun memodifikasi flag Sign, Zero, dan Parity sesuai dengan hasilnya

Instruksi NOT

- Instruksi NOT membalik semua nilai bit pada operand
- Sintaks:

NOT *destination*

- Operand destination berupa register atau memori
- Hasil dari instruksi NOT adalah komplement-satu dari operand
- Instruksi NOT tidak memodifikasi FLAG apapun

operand

0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---

NOT NOT NOT NOT NOT NOT NOT NOT

NOT operand

1	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---

Aplikasi Operasi Boolean

- Operasi boolean umumnya digunakan untuk memanipulasi nilai-nilai bit individu
- Membersihkan nilai register dengan XOR
 - Misalkan kita ingin membersihkan (mengnolkan) nilai register EAX
 - Kita dapat menggunakan instruksi: `xor eax, eax` ; EAX = 0
- Untuk menyalakan bit tertentu dengan OR
 - Misal kita ingin menyalakan bit 3 dari nilai dalam register EAX (menyalakan bit berarti membuat nilai bit tersebut menjadi 1)
 - Kita dapat melakukan operasi OR terhadap nilai EAX dengan 0000000000001000
 - Jalankan instruksi:

```
mov    ebx, 1    ; buat angka 0...01
shl    ebx, 3    ; geser bit ke kiri sebanyak 3
or     eax, ebx  ; nyalakan bit 3
```



Aplikasi Operasi Boolean

- Untuk mematikan bit tertentu dengan AND
 - Misal kita ingin mematikan bit 4 dari nilai dalam register `EAX` (mematikan bit berarti membuat nilai bit tersebut menjadi 0)
 - Kita dapat melakukan operasi `AND` terhadap nilai `EAX` dengan `111111111101111`
 - Jalankan instruksi:

```
mov     ebx, 1      ; buat angka 0...01
shl     ebx, 4      ; geser bit ke kiri sebanyak 4
not     ebx         ; ambil nilai komplemen
and     eax, ebx    ; matikan bit 4
```



Instruksi TEST

- Instruksi TEST melakukan operasi AND namun tidak menyimpan hasilnya
- Instruksi TEST memodifikasi nilai dari flag Sign, Zero dan Parity seperti yang dilakukan instruksi AND
- Sintaks:

`TEST destination, source`

- Contoh: Lompat ketika suatu nilai adalah genap

```
test     al, 01h    ; jika genap, maka ZF = 1
jz       target     ; jz (jump if zero): jika genap lompat
                        ; ke target dan jika tidak ke instruksi
                        ; berikutnya
```