

Organisasi Sistem Komputer

Bab 6. Operasi Bit

6.1 Pergeseran Bit

6.2 Operasi Boolean



Pergeseran Bit

- Terdapat tiga macam pergeseran bit:
 - Pergeseran logis: SHL dan SHR
 - Pergeseran aritmatika: SAL dan SAR
 - Pergeseran rotasi: ROL, ROR, RCL, dan RCR

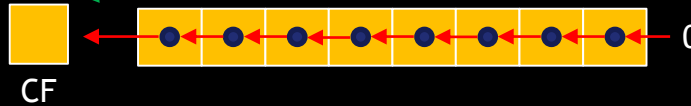


Pergeseran Logis

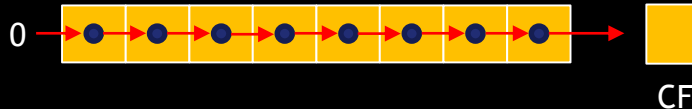
- Dua instruksi:
 - **SHL**: *Shift Left* menggeser satu bit ke **kiri** dan memasukkan nilai 0 pada posisi bit terendah (LSB)
 - **SHR**: *Shift Right* menggeser satu bit ke **kanan** dan memasukkan nilai 0 pada posisi bit terbesar (MSB)

- Ilustrasi Pergeseran Bit

SHL



SHR



Nilai pada bit yang digeser ke luar menjadi nilai bit **flag carry**

Pergeseran Logis

- Sintaks:

SHL *destination, count*

SHR *destination, count*

count = banyak bit yang digeser

- Kombinasi operand:

- *reg, imm8*

imm8 adalah *immediate value* 8-bit (nilai langsung 8-bit)

- *mem, imm8*

- *reg, CL*

CL adalah 8-bit terbawah register ECX

- *mem, CL*

Contoh Pergeseran Logis

- Contoh:

```
mov     al, 0C6h        ; AL = 1100 0110
shl     al, 1            ; AL = 1000 1100 (8Ch), CF = 1
shr     al, 1            ; AL = 0100 0110 (46h), CF = 0
shl     al, 3            ; AL = 0011 0000 (30h), CF = 0

mov     cl, 2
shr     al, cl           ; AL = 0000 1100 (0Ch), CF = 0
```



Aplikasi Pergeseran Logis

- Aplikasi paling umum dari pergeseran logis adalah untuk: **mengalikan dan membagi dengan 2^n secara cepat**
- SHL sebanyak n bit untuk mengalikan dengan 2^n

```
mov    dl, 5  
shl    dl, 2
```

$$5 \times 2^2 = 5 \times 4 = 20$$

Sebelum: 00000101 = 5
Sesudah: 00010100 = 20

- SHR sebanyak n bit untuk membagi dengan 2^n

```
mov    dl, 64  
shr    dl, 3
```

$$64 / 2^3 = 64 / 8 = 8$$

Sebelum: 01000000 = 64
Sesudah: 00001000 = 8

Aplikasi Pergeseran Logis

- Ketika kita membagi bilangan ganjil dengan SHR, hasilnya akan dibulatkan ke bawah

```
mov    al, 10011b      ; 19d
shr    al, 1            ; al = 0000 1001 = 9d
shr    al, 1            ; al = 0000 0100 = 4d
```

- Selama bit terbesar dengan nilai 1 tidak hilang, SHL memberikan hasil perkalian dengan 2^n yang benar
 - Misal, 00101001 (41d), hanya bisa digeser 2 bit ke kiri sebelum bit 5 (bit dengan nilai satu paling kiri) hilang
- Begitu juga dengan SHR, selama bit terbesar dengan nilai 1 tidak hilang, akan memberikan hasil pembagian dengan 2^n yang benar

Pergeseran Logis dan Bil. Bertanda

- Pergerakan logis (SHR dan SHL) untuk perkalian dan pembagian hanya bekerja pada bilangan tidak bertanda
- Contoh:
 - Pada bil. tidak bertanda:
 - $\text{FE} = 254\text{d} = 11111110\text{b}$
 - geser kanan 1 bit: $01111111\text{b} = 7\text{Fh} = 127\text{d}$ (sama dengan $254/2$)
 - Pada bil. bertanda:
 - $\text{FE} = -2\text{d} = 11111110\text{b}$
 - geser kanan 1 bit: $01111111\text{b} = 7\text{Fh} = +127\text{d}$ (tidak sama dengan $-2/2$)
- Untuk bilangan bertanda kita gunakan pergeseran aritmatika

Pergeseran Aritmatika

- Terdapat dua instruksi:

- ❑ SAL (Shift Arithmetic Left)
- ❑ SAR (Shift Arithmetic Right)

- Instruksi SAL:

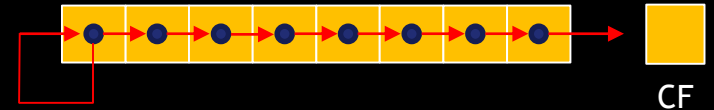
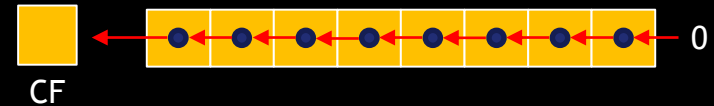
- ❑ Sama persis seperti SHL
- ❑ Hanya nama lain sehingga kita dapat ingat kita sedang bekerja dengan bilangan bertanda

- Instruksi SAR:

- ❑ Hampir sama dengan SHR, namun tidak menggeser bit tanda
- ❑ Bit baru yang masuk dari kiri adalah salinan dari bit tanda

SAL *destination, count*

SAR *destination, count*



Contoh Pergeseran Aritmatika

- Contoh SAR dan SAL:

```
mov    al, 0C3h ; al = 1100 0011 (-61d)
sal    al, 1     ; al = 1000 0110 (86h = -122d)
sar    al, 3     ; al = 1111 0000 (F0h = -16d)
                    ; (catatan: ini bukan pembagian karena kita
                    ; kehilangan bit di kanan)
sal    al, 4     ; al = 0000 0000 (0d, bukan hasil perkalian
                    ; yang benar)
```



Pergeseran Rotasi

- Terdapat empat instruksi pergeseran rotasi:

❑ ROL (Rotate Left)

ROL *destination, count*

❑ ROR (Rotate Right)

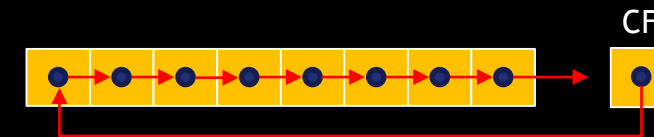
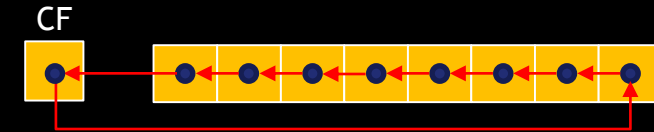
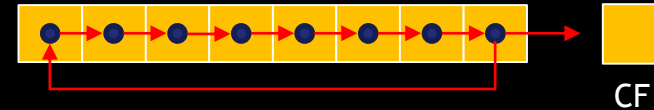
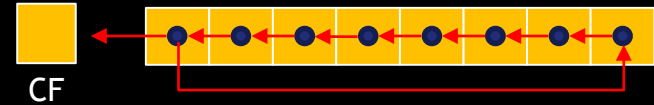
ROR *destination, count*

❑ RCL (Rotate Carry Left)

RCL *destination, count*

❑ RCR (Rotate Carry Right)

RCR *destination, count*



Contoh Pergeseran Rotasi

- Contoh ROL:

```
mov    al, 40h    ; AL = 01000000b
rol    al, 1       ; AL = 10000000b, CF = 0
rol    al, 1       ; AL = 00000001b, CF = 1
rol    al, 1       ; AL = 00000010b, CF = 0
rol    al, 7       ; AL = 00000001b, CF = 1
```



Contoh Pergeseran Rotasi

- Contoh ROR:

```
mov    al, 01h    ; AL = 00000001b
ror     al, 1      ; AL = 10000000b, CF = 1
ror     al, 1      ; AL = 01000000b, CF = 0
ror     al, 4      ; AL = 00000100b, CF = 0
```

- Contoh RCL:

```
clc                ; CF = 0
mov     bl, 88h    ; CF = 0, BL = 10001000b
rcl     bl, 1      ; CF = 1, BL = 00010000b
rcl     bl, 1      ; CF = 0, BL = 00100001b
```

Instruksi `CLC` (clear carry)
meng-clear (mengubah nilai
menjadi nol) nilai flag carry



Contoh Pergeseran Rotasi

- Contoh RCR:

```
stc          ; CF = 1
mov     ah, 10h    ; AH = 00010000b, CF = 1
rcr     ah, 1      ; AH = 10001000b, CF = 0
```

Instruksi STC (set carry)
meng-set (mengubah nilai
menjadi satu) nilai flag carry